THE OPPORTUNITIES AND LIMITATIONS OF USING OLAP CUBES FOR BUSINESS INTELLIGENCE REPORTING IN BANKING

ONUR CAN ULAŞ

BOĞAZİÇİ UNIVERSITY

2011

THE OPPORTUNITIES AND LIMITATIONS OF USING
OLAP CUBES
FOR BUSINESS INTELLIGENCE REPORTING IN BANKING

Thesis submitted to the

Institute for Graduate Studies in the Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

Onur Can Ulaş

Boğaziçi University

2011

The Opportunities and Limitations of Using
OLAP Cubes
for Business Intelligence Reporting in Banking

The thesis of Onur Can Ulaş
has been approved by:

Prof. Dr. Meltem Özturan
(Thesis advisor) ____________________

Prof. Dr. Nuri Başoğlu ____________________

Assoc. Prof. Mehpare Timor ____________________

June 2011

Thesis Abstract

Onur Can Ulaş, "The Opportunities and Limitations of Using OLAP Cubes for Business Intelligence Reporting in Banking"

New rules of competition and rapid changes within the industry force banks to create new business strategies and be more efficient in making decisions. OLAP is a business intelligence technology utilized for rapid access and analysis of information from multiple perspectives. The purpose of this study was to demonstrate opportunities and limitations of using OLAP Cubes for business intelligence reporting in banking from a technological point of view.

Triangulation technique on the basis of methods and data sources was adopted to enhance the validity and reliability of the study and crosscheck the findings from a particular method. First, theoretical background and related literature were reviewed. Secondly, a comparative experiment through design and development of an OLAP Cube was carried out to determine the opportunities and limitations of OLAP cubes and to reevaluate the arguments in theory. Besides, a structured questionnaire was conducted among banking IT experts to confirm the findings of the experiment. Thus, this is an explanatory, deductive and quantitative study, however, descriptive, inductive and qualitative strategies were involved as well at some points of the research.

The results indicated that OLAP was more appropriate than relational model in banking due to greater query performance and simplicity in conducting complex queries with specific purposes such as management information reporting. However, it was less flexible than the relational model, which provides the ability to serve further needs and easily develop structured reports. Thus, it should be well decided which model must be used in accordance with the purpose.

Tez Özeti

Onur Can Ulaş, "Bankacılık İş Zekası Raporlamasında OLAP Küp Kullanımının Yararları ve Sınırlılıkları"

Rekabetin yeni kuralları ve bankacılık endüstrisindeki hızlı değişim, bankaları, yeni iş stratejileri geliştirmeye ve karar verme süreçlerinde daha etkin olmaya zorlamaktadır. OLAP, bilgiye, değişik perspektiflerden ve hızla ulaşmada ve bu bilginin analizinde yararlanılan bir iş zekası teknolojisidir. Bu çalışmanın amacı, OLAP Küplerinin bankacılık iş zekası raporlamasındaki yarar ve sınırlılıklarını teknolojik bakış açısıyla göstermektir.

Çalışmanın geçerliliğini ve güvenilirliğini arttırmak ve belirli bir yöntemle elde edilen bulguları farklı yollardan doğrulamak için, yöntem ve veri kaynakları temelinde triangülasyon tekniği benimsenmiştir. İlk olarak, konunun kuramsal temelleri ve ilgili kaynaklar incelenmiştir. Ardından, OLAP Küplerinin yarar ve sınırlılıklarını belirlemek ve kuramsal savları yeniden değerlendirmek amacıyla, bir OLAP Küpü tasarlanmış, geliştirilmiş ve karşılaştırmalı bir deney yapılmıştır. Ayrıca, bu deneyden elde edilen bulguları sınamak amacıyla, yapılandırılmış bir anket formu kullanılarak banka bilgi teknolojisi uzmanlarının görüşlerine başvurulmuştur. Bu nedenle, bu, açıklayıcı, tümdengelimli ve niceliksel bir çalışmadır, bununla birlikte, bazı noktalarda betimleyici, tümevarımlı ve niteliksel stratejilerden de yararlanılmıştır.

Çalışma sonuçları, daha güçlü sorgulama performansı ve yönetim bilgi raporlaması gibi belirli amaçlara yönelik karmaşık sorgulamaları kolaylıkla gerçekleştirebilmesi nedeniyle, OLAP'ın, bankacılık için, ilişkisel modele oranla daha uygun bir teknoloji olduğunu göstermiştir. Ancak, OLAP, gelecekteki gereksinimleri karşılama ve yapılandırılmış raporları kolaylıkla hazırlama olanağı sağlayan ilişkisel modelden daha az esnektir. Bu nedenle, amaç gözönünde bulundurularak, hangi modelin kullanması gerektiği konusunda doğru karar verilmelidir.

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Prof. Dr. Meltem Özturan, the member of Management Information Systems Department at Boğaziçi University, for her excellent guidance, advices, understanding, trust and patience during the completion of my thesis. The review and comments of Prof. Dr. Nuri Başoğlu, a member of the same department of the University, on my work have also been valuable. I also would like to thank Assoc. Prof. Dr. Mehpare Timor, the member of Istanbul University Business School, for the time she spent over my thesis by attending the thesis jury.

I especially owe many thanks to my mother for her support, encouragement and constant assistance during the study. This thesis would not be completed without her.

I want to express my deepest gratitude to the experts of the Bank, especially to Emine Göktay and Tuğrul Nakiboğlu, who have contributed to my thesis in various ways. They have provided me with their insights, opinions and valuable feedback during every phase of the study, besides answering the questionnaires, which has been so important for this research.

I also would like to thank my girlfriend Gökçe Kurt for her endless patience and moral support.

Last, I would like to thank Aylin Yılmaz, the editor of Institute of Social Sciences at Boğaziçi University, for the time and effort that she put into proofreading my thesis.

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

1NF	First Normal Form
2NF	Second Normal Form
3NF	Third Normal Form
ABC	Activity Based Costing
ABDOP	Analysis-Based Decision-Oriented Information Processing
AI	Artificial Intelligence
AJI	Aggregate Join Indexes
APL	A Programming Language”
BI	Business Intelligence
BO	Business Objects
BUS	DW Bus
CIF	Corporate Information Factory
CMMI	Capability Maturity Model Integration
CPM	Corporate Performance Management
CRM	Customer Relationship Management
DDL	Data Definition Language
DEV	Development
DM	Data Mart
DOLAP	Desktop OLAP
DSS	Decision Support System
DSS	Decision Support Systems
DSV	Data Source View
DW	Data Warehouse

EIS	Executive Information Systems
EOM	End Of Month
ER	Entity Relationship
ETL	Extract, Transform and Load
FASMI	Fast Analysis Of Shared Multidimensional Information
GB	Gigabytes
HOLAP	Hybrid OLAP
JISR	Journal Of Independent Studies And Research
KPI	Key Performance Indicator
MB	Megabytes
MDX	Multidimensional Expressions
MH	Man/Hour
MI Reporting	Management Information Reporting
MIN	Minute
MIS	Management Information Systems
MOLAP	Multidimensional OLAP
MQPSB	Merging Of Query Point Set And B-Tree
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
PROD	Production
QPRQ	Query Processing With Range Queries
RDBMS	Relational Database Management Systems
ROI	Return On Investment
ROLAP	Relational OLAP

SP2	Service Pack 2
SQL	Structured Query Language
SSAS	Server 2005: Analysis Services
SSRS	MS SQL Server 2005: Reporting Services
TWS	Tivoli Work Scheduler
WOLAP	Web OLAP

CHAPTER 1

INTRODUCTION

The widespread use of information technology generates tremendous amounts of data within an organization. This data contains information that is invaluable to the organization's decision makers (Ritacco & Carver, 2007). It is compulsory to transform these data into consumable information for business organizations to be able to compete in the market. Business Intelligence (BI) is a type of Decision Support System (DSS) and refers this transformation. Useful information can be derived from data available inside and outside the organization through BI technologies to promote effective decision-making. Data warehousing, Data Mining and On-Line Analytical Processing (OLAP) are key BI technologies. Data is stored and managed in data warehouse (DW), while OLAP and Data Mining convert the data into consumable information to promote its utilization. OLAP, which has become one of the standard services for business organizations, provides analysts and managers with the opportunity of rapid access and analysis of shared information. Users can also look at the information from multiple perspectives through OLAP Cubes.

There are significant changes in banking industry in recent years. As a result of globalization, technological innovations, domestic or international mergers and acquisition, diversification and deregulation in certain countries and ongoing regulatory changes, banking industry is being forced to create new business strategies for competition, success or even survival in the market. Banks have to be more efficient in making decisions, improve operational performance, maximize channel and product productivity and profitability, increase workforce performance and reduce risks and operational costs. These circumstances create an urgent need for stronger decision making and information management systems and drive banks to invest more in BI technology. The main benefits BI systems bring to any bank are their abilities to provide a deep understanding of past and current operations, and forecast future events (Knapik, 2007). OLAP Cube services are considered to promote fast and multidimensional analysis of information on these business variables in banking as well.

In this thesis, the role of using OLAP Cubes in improving management information reporting (MI reporting) in a banking organisation was examined to gain a more reliable understanding in this scope. The thesis was based on existing BI infrastructure and needs of one of the leading commercial banks of Turkey whose name will remain undisclosed upon their request and be referred as “the Bank” within the text.

Main goal of the study is to demonstrate the opportunities and limitations of using OLAP Cubes for BI reporting in banking. It was limited to a technology perspective and didn’t investigate the consequences of using OLAP Cube services from corporate perspective. An opportunity was assumed in this study as a pattern promoting the improvement of BI reporting and a limitation was considered as a constraint for use of BI technology.

The objectives of the thesis to meet the main goal of study, to demonstrate how an OLAP Cube can be utilized to improve MI reporting in banking are as follows:

Firstly, investigating a Cube structure, which could support efficient reporting, faster processing than relational database queries, provide multidimensional access to such large scale data. For this purpose, a Cube would be designed and developed for reporting. It was aimed to show the quantitative opportunities and constraints of an OLAP Cube by checking its performance and additionally to examine OLAP queries in the context of their functional properties.

Secondly, investigating the opinions of the BI experts in banking sector to confirm the findings of Cube implementation and particularly examine qualitative aspects of using OLAP Cubes through a questionnaire.

This study is guided by a main research question that was specified as: *What are the opportunities and limitations of using OLAP Cubes for BI reporting in banking?*

The concept of OLAP Cube is relatively new to the Turkish banking industry. This study is significant for banking institutions as it extends the knowledge base that currently exists in that field and explores the benefits, advantages and/or constraints of such technology. Findings of this thesis can be utilized by developers and users of BI applications.

Since the academic literature on this subject is rather limited (see Chapter 3), this study may also provide a useful source of information for academicians working in the field of BI.

The thesis is structured as follows:

Chapter 2 gives an overview of fundamental concepts and features of BI; highlights needs for BI in banking and also covers BI applications utilized in banking and adaptation of BI tools in banks including traditional DW applications. More focus is put on OLAP; basic *concepts, operations, types and banking applications of* OLAP *Cubes* are examined.

Chapter 3 consists of literature review; provides an overview of previous studies relevant to purpose of this thesis.

Chapter 4 states research questions; describes the methodology of the thesis, including type, approach and data collection, measurement and analysis methods and also evaluates the reliability and validity of the study.

Chapter 5 analyzes existing BI infrastructure and reporting environment of the Bank and introduces design and development processes of the OLAP Cube and describes general features of business reporting in banking by demonstrating them on this particular implementation.

Chapter 6 presents the performance tests performed to determine quantitative properties of the Cube in comparison with traditional DW and the observations for its functional features.

Chapter 7 interprets the results from the experiment and questionnaires; makes functionality assessments and compares determined properties of the Cube with the features of traditional reporting system.

Chapter 8 summarizes identified opportunities and the limitations of Cube applications in comparison with relational model; concludes the basic findings and main arguments of the study, explains to what extent they differ from theory. This chapter also states contributions of this thesis and provides directions for future research.

A number of appendices follow chapter 8.

CHAPTER 2

THEORETICAL BACKGROUND

Definition and Conceptual Framework of Business Intelligence

Hans Peter Luhn first used the term *Business Intelligence* in 1958. He defined BI as: "the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal" (Luhn, 1958, p. 314). In 1989, Howard Dresner proposed "business intelligence" as an umbrella term to describe concepts and methods to improve business decision making by using fact-based support systems (Power, 2007).

The idea behind BI addresses managerial problems and activities that are not totally new. Decision Support Systems (DSS), Executive Information Systems (EIS) and Management Information Systems (MIS) are the fields where the roots of BI exist. According to Wu (2000), BI is the process of gathering high-quality and meaningful information about the subject matter being researched that will help the individual(s) analyzing the information, draws conclusions or make assumptions. In another study, it is considered that an ideal BI system gives an organization's employees, partners, and suppliers easy access to the information they need to effectively do their jobs, and the ability to analyze and easily share this information with others (Nadeem & Jaffri, 2004). Negash (2004) defines the BI system as a tool that combines data gathering, data storage, and knowledge management with analytical tools to present complex internal and competitive information; provides actionable information delivered at the right time, at the right location, and in the right form to planners and decision makers. Robinson (2008) suggests that the BI infrastructure delivers key information to business users. Providing managers and knowledge workers with new tools allowing them to see data in new ways empowers them to make faster and better decisions. Rather than responding to continuous stream of report requests, BI platforms provide business users self-service decision support.

Pirttimaki (2007) states that the concepts such as competitive intelligence, strategic intelligence, market intelligence and technological intelligence are sometimes used in a context similar to BI. However, almost all of these intelligence approaches are synonym or subgroup of BI and share the same purpose as BI: Transforming data into valuable, useful, meaningful, insightful information. BI produces up-to-date information for both operative and strategic decision making.

On the other hand, there is a big debate on definition of BI in theory, particularly between the two pioneers of the field. Kimball, who is known as “the Father of Business Intelligence” for his definitions about data marts (DM), dimensional hierarchies refers to overall process of providing information to support business decision making as “data warehousing”; defines DW as “the foundation of business intelligence” (Kimball, Ross, Thornthwaite, Mundy & Becker, 2008) while according to Inmon, who is known as “the Father of Data Warehousing concept” DW is one part of overall BI system (Inmon, 2005). Inmon specifies “structured visualization” of data as BI and considers that, with DW, BI became a possibility, without the DW, BI was just a theory. At this point, the process and components of BI should be examined.

Business Intelligence Process

Even if there is a debate regarding the content of BI and there are differences among process models such as the structure of cycles, sources of information and methods of gathering, analyzing and storing information, BI can be approached as “a form of a cycle simply acquires, analyzes, stores and disseminates essential information and contains elements required to produce valuable business information” (Pirttimaki, 2007, p. 72).

Different BI architectures are suggested for *structured, unstructured or semi-structured data* even though all of them are required as parts of BI process. Inmon (2005) defines *Unstructured -semi-structured data- (documents* and *communications)* as data types whose content has no format and cannot be stored in columns and rows and according to this author, structured data is data whose content is organized into predictable format, which has keys, fields, records, databases and created as a byproduct of transactions. Typical BI architecture for structured data centers on a DW. BI architecture for semi-structured data includes business function model, business process model, business data model, application inventory and metadata repository (Negash, 2004).

Although there are different arguments for staging of BI process, BI infrastructure comprises these layers or phases in general:

1. The Data Collection (Sources)
2. Data Integration (ETL: Extract, Transform, and Load)
3. The Data Warehouse (Storage)
4. Data Analysis (OLAP or Data Mining)
5. Presentation

A view of the BI infrastructure is shown in Figure 1.

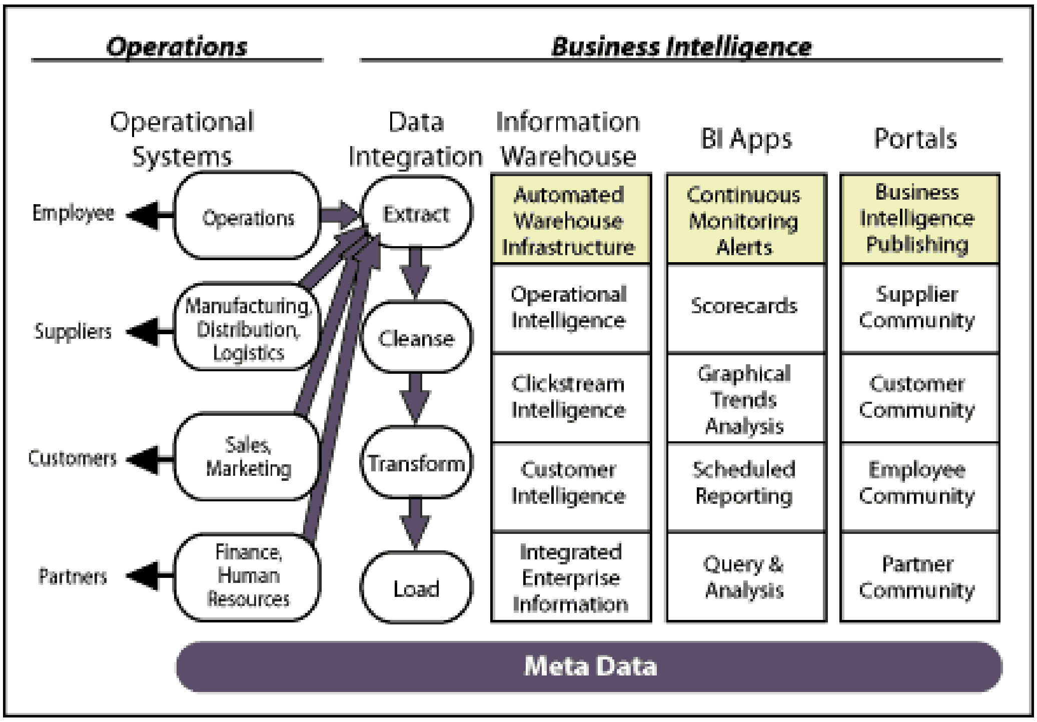


Figure 1. BI infrastructure. Reprinted from "Business Intelligence Infrastructure" by M. Robinson, 2008, *Information Management Online.* Retrieved February 16, 2011 from http://www.information-management.com/specialreports/20020521/5211-1.html.

Raw data that needs to be analyzed is collected from several operational systems or administrative processes or from external sources.

These data is transformed into meaningful information through *data integration*. *Integration layer* involves profiling to evaluate validity and reliability of data along with extracting, transformation, staging, cleansing, merging and loading processes.

Then, data are *stored* in DM or in DW which are databases developed to organize and process integrated information on confirmed variables. "A *data mart* is a data structure that is dedicated to serving the analytical needs of one group of people" (Inmon, 2005, p. 370). DW concept will be explained in detail in the following sections.

The methods such as OLAP and data mining are utilized to *analyze* data. *Data analysis* provides the evaluation and interpretation of current circumstances of business activities.

In the fifth and last stage, the information that have been previously transformed and analyzed are *displayed* to business users through several presentation techniques such as reports, newsletters, Web-based portals, graphs, balanced scorecards, dashboards and enables them to identify, query and analyze business variables of the organization.

These stages and components of BI will be examined in detail in the next parts of the study.

As mentioned above, BI process is considered as a repeating cycle in certain studies and it is believed that the last phase closes the loop between data collecting and information utilization stages (see Figure 2). This cycle occurs repeatedly until the decision-makers find appropriate answers. The outputs of a process serve as input for a new iteration of an ongoing BI process (Pirttimaki, 2007). In the first phase of each new cycle the success of the previous cycle can be evaluated, and taken into consideration when designing the new cycle.

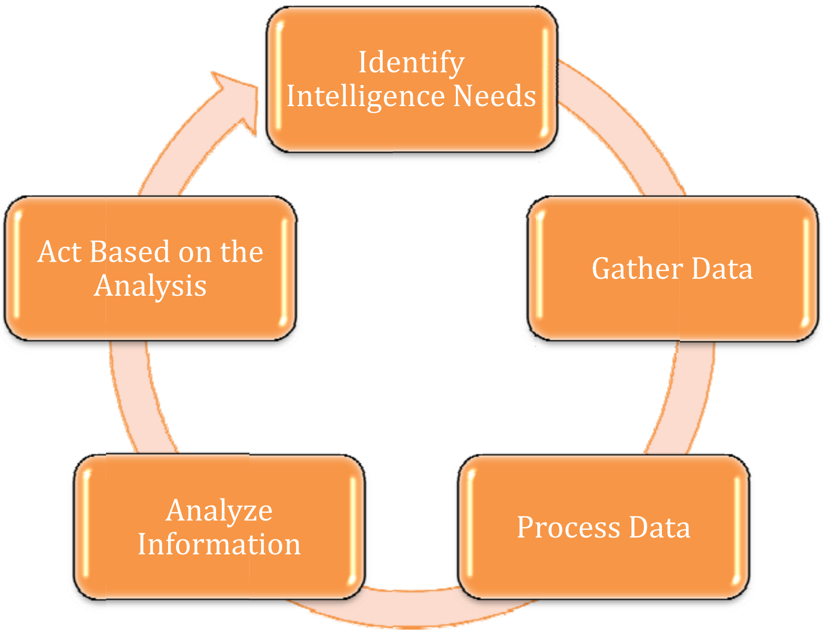


Figure 2. BI cycle: Phases of BI process.

Even though the scope of this thesis is limited to a technology perspective, briefly touching the business benefits of BI may be useful for readers to realize consequences of using OLAP Cube services in BI reporting.

Business Benefits of BI

BI supports all divisions of an organization at operational, tactical and strategic levels.

Ritacco & Carver (2007) group business benefits of BI into three categories:

1. Lowering costs (by improving operational efficiency, eliminating report backlog and delays, negotiating better contracts with suppliers and customers, finding root causes and taking actions, identifying wasted resources and reducing inventory costs, leveraging investments in the enterprise research planning or DW);

2. Increasing revenue (by providing information to customers, partners, and suppliers, improving strategies with better marketing analysis, empowering sales force);
3. Improving customer satisfaction (by giving users the means to make better decisions, providing quick answers to their questions, challenging assumptions with factual information).

An empirical study carried out on fifty Finnish companies demonstrated that the most important three benefits expected from BI were improved quality of information, internal information dissemination and level of awareness (Pirttimaki, 2007).

Robinson (2008) summarizes the quantifiable benefits of providing a BI platform as the decisions which increase revenue by identifying and creating up-sell and cross-sell opportunities, improve "valued customer" profitability, decrease costs or expenses by leveraging infrastructure and automating processes, decrease investment in assets such as inventory, or improve productivity with better decision making and faster response-to-market changes or other business events.

The Need for BI in Banking

Banks encounter global competition, new business rules, mergers and acquisitions, ongoing regulatory changes, diversification of products and technological innovations that force the industry as a whole to create new business strategies for surviving in the market. They must increase profits and market shares, create new revenue sources, reduce operational costs, maximize the value of stocks and return on investment (ROI). Banks have to improve operational performance, maximize channel, workforce and product performance and productivity and reduce operational risks and costs to achieve these goals.

Besides, customers' expectations are changing. They are becoming better informed and more demanding. Therefore, companies are therefore transforming their management strategy to become more customer-centric than product focused (Nadeem & Jaffri, 2004).

"Business users must have access to all information they require to fulfil their tasks and make their informed decisions. Additionally decision makers have to utilize multiple information analysis and knowledge extraction techniques, which subsequently drive decision processes. Knowledge sharing and dissemination is the final achievement towards the effective organizational information integration." (Retail banking, n. d., para. 1).

Effective and innovative use of information technology in banking is highly significant to be able to adapt the changes in market and develop different strategies.

According to Knapik (2007), growth of data volumes in disparate sources and growing technology capabilities are driving financial service organizations to increase investment in BI technologies. Banks must align intelligent technology with data management techniques in order to improve their decision-making processes.

Evolution of BI in Banking and Its Adaptation into Banks

Even when there were no computers, banks had put in place an efficient system of recording various transactions. Most business transactions took place at branches, which were supplying both management and regulatory reports. These reports were manually consolidated at intermediate controlling offices for eventual aggregation at the corporate level. These manual systems worked well till the scale of operations were relatively small. As the banks grew in size and expanded geographically, the volume of transactions became quite large. Manual aggregation became both time consuming and error prone. Thus, banks began to use computers to automate the aggregation process (Misra, 2007, p. 5).

However, in the first years of computer utilization, MIS in the banks had the significant difficulties such as time lag, limited data quality, unavailability of customer specific data and unavailability of data specifications required for developing analytics. Inflexibility and batch processing were soon overcome by powerful desktop systems with rudimentary database systems, which allowed banks to analyse data. These earlier initiatives laid the foundations of BI in banking (Misra, 2007).

Software companies started introducing bank-targeted products within the late 1990's and early 2000's. Until that time, many banks had to create their own bank end solutions, ETL solutions and apply their unique business rules (Banking business intelligence, n.d.).

Today, BI technologies are already widely employed in banking. Banks adopted BI and performance management methods and tools that enable the decision makers within the banking industry to identify the issues, make the best decisions or take the best actions depending most accurate, complete and timely information and evaluate the impacts of decisions.

Utilization of BI in Banking

BI tools can meet basic needs of banking by providing necessary reporting and data analysis infrastructure including:

- Real-time reporting and performance monitoring of key business metrics

- Monitoring of budget performance
- Temporal analysis of basic business measures
- Automatic creation and delivery of important reports
- A user friendly environment demanding minimal effort from the business end-users
- Information analysis and monitoring, over a limitless number of business dimensions
- Effective segmentation of available information (geographical analysis, customer segmentation, product analysis) and implementation of different business scenarios
- Easy access to all information levels, from business performance indicators to actual business activity data (such as portfolio data, transactional data, branches and employees activities)
- Effective monitoring and assessment of business processes
- Investment services and portfolio management
- Information exchange required to ensure compliance to all governmental and international regulations (Banking business intelligence, n.d.; Retail banking, n. d.).

More concretely, the areas that BI information tools promote in banking are:

- Product/service profitability (such as commercial loans, consumer loans, mortgage loans across regions, divisions)
- Customer relationship management (CRM)
- Activity based costing (ABC)
- Customer profiling and segmentation (including customer profitability, credit scores)
- Risk management (credit risk, market risk and operational risk)

- Sales and marketing (financial views by various parameters such as region, personnel, channel, branch)
- Fraud detection
- Tracking and identification of anti-money laundering
- Corporate performance management (CPM)

Through an ideal BI infrastructure, reports and forecasts can be created and consumed easily; information is delivered securely; past and current performance can be evaluated accurately and the impacts of changes can be analyzed quickly in banking.

Relational Database Concept

The relational database model was first introduced and formulated by E. F. Codd in 1970 as a basis for protecting users of formatted data systems from the potentially disruptive changes in data representation caused by growth in the data bank and changes in traffic; as a normal form for the time-varying collection of relationships (Codd, 1970). The conceptual schema of the relational model is largely derived from set theory (and theory of relations) and the predicate calculus. The relational model defines a declaratively oriented relational calculus and a procedurally oriented relational algebra (Thomsen, 2002, p. 31). Codd (1970) considered a database as a collection of *predicates* (such as database queries) that describes constraints on the values. The relational model is a central part of Online Transaction Processing (OLTP), which is the whole concept of managing databases in a relational manner. Software that is used to handle the database is referred to as Database Management Systems (DBMS), and the term Relational Database Management Systems (RDBMS) is used to indicate that it is a relational database system. The adjective *relational* refers to the models with the fundamental principle of representing data consistently by mathematical relations, which are implemented as tables (Westerlund, 2008).

The relational approach to database design begins with the organization of data into a table. Different columns are in each row of the table. The relational table can have different properties. The columns of data have different physical characteristics. Different columns can be indexed and can act as identifiers. Certain columns may be null upon implementation. The columns are all defined in terms of a data definition language (DDL) statement. Figure 3 illustrates a relational database design. There are different tables, and the tables are connected by means of a series of key-foreign key relationships (Inmon, 2005).

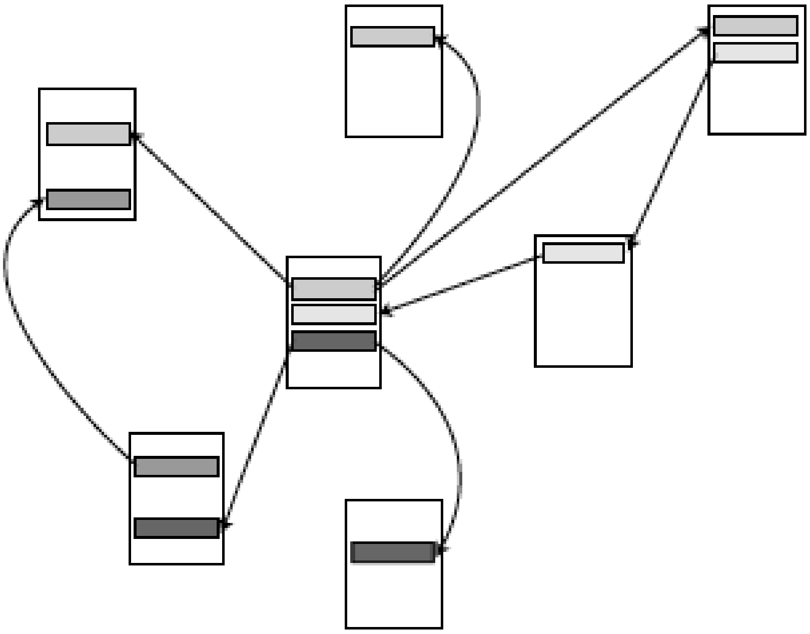


Figure 3. A relational database design. Reprinted from *Building the Data Warehouse* (Forth ed.) (p. 358), by W. H. Inmon, 2005, New York: Wiley Publishing.

Since the tables of records in this model correspond to real-world entities, the model is sometimes referred as "Entity Relationship (ER) Models."

“The most important concept of relational databases is *normalization*. Normalization is done by adding constraints to how data can be stored in the database, which implies restrictions to the way data can be inserted, updated and deleted. The main reasons for normalizing database design is [sic] to minimize information redundancy and to reduce disk space required to store the database” (Westerlund, 2008, p. 7). *Normalization of data* implies that the database design has caused the data to be broken down into a very low level of granularity. When normalized, the data inside a table has a relationship with only other data that resides in the table. Normalization is said to typically exist at three levels—first normal form (1NF), second normal form (2NF), and third normal form (3NF) (Inmon, 2005). 1NF simply means that in every row of a table, there can only be one value per attribute. 2NF means that every attribute that is not contained in the key set must provide a fact about the key, the whole key and nothing but the key. 3NF implies that there can be no transitional dependencies (Westerlund, 2008).

Another important issue of relational databases is how to optimize the database for querying. The most common method is to use *indexing*, that is having an *index structure* containing pointers to the actual data in order to optimize search operations. A good index structure can increase the performance of a search significantly (Westerlund, 2008).

Data Warehousing: The Backbone of BI in Banking

The term “data warehousing” refers to process of building and using DW. However, as previously mentioned, there is a strong dispute on definition of DW between Inmon and Kimball, the two leading authors and constructors of data warehousing. Inmon refers DW as core of the whole system where data is stored and accessed for analysis while

Kimball believes DW is a complete solution for analytical processing and decision making.

Inmon introduces DW as “the heart of architected environment” and the foundation of all DSS processing and defines it as “a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management’s decisions " (Inmon, 2005, p. 29). On the other hand, Kimball and Ross argues against supposed DWs, which are mere copies of the operational system of record stored on a separate hardware platform (Kimball & Ross, 2002). According to Kimball and Ross, DW is “the conglomeration of an organization’s data warehouse staging and presentation areas, where operational data is specifically structured for query and analysis performance and ease-of-use” (Kimball & Ross, 2002, p. 397). Kimball and his friends disagree with others who suggest that the DW is a highly normalized data store whose primary purpose is not query support, but to serve as a source for the transformation and loading of data into summarized dimensional structures (Kimball et al., 2008).

Due to this conceptual debate in literature on the distinction between BI and DW systems, some of the studies refer the processes as “DW/BI” (Kimball & Ross, 2002; Kimball et al., 2008; Boselli, Cesarini & Mezzanzanica, 2010).

The information warehouse is architected to include key business variables and business metrics in a structure that meets all business analysis questions required by the business groups (Robinson, 2008).

Components of Data Warehouse

Kimball and Ross (2002) introduce four separate and distinct components of DW environment which have been listed below and demonstrated in Figure 4:

1. Operational Source Systems

2. Data Staging Area

3. Data Presentation Area

4. Data Access Tools

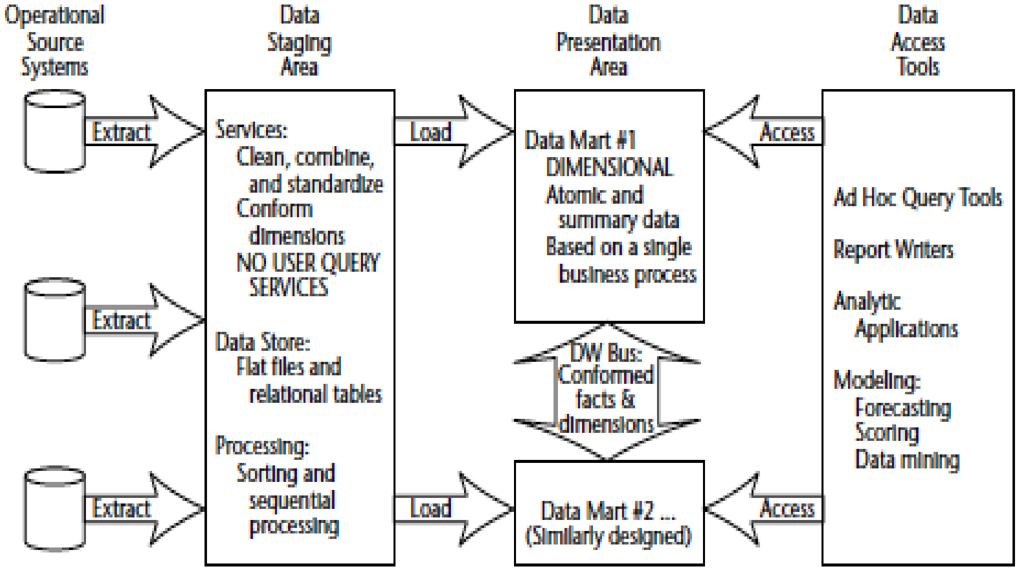


Figure 4. Components of data warehouse. Adapted from *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (Second ed.) (p. 7), by R. Kimball & M. Ross, 2002, New York: John Wiley & Sons.

Operational Source Systems and the Data

Building a DW requires assessing the information sources data contents, in terms of available data, structure in which data is managed, and quality. When multiple data sources are involved, data should also be merged into an integrated archive, which has to be designed and implemented. Next, the source data quality should be evaluated and quality improvement activities should be established (Boselli et al., 2010).

Kimball and Ross (2002) state that the main priorities of the source systems are processing performance and availability and these systems are not queried in the broad and unexpected ways that DWs typically are queried.

Table 1 shows the differences between *operational* (“primitive”) data and *derived* data summarized by Inmon (2005).

Table 1. Differences Between Primitive and Derived Data.

Primitive Data / Operational Data	Derived Data / DSS Data
Application-oriented	Subject-oriented
Detailed	Summarized, otherwise refined
Accurate, as of the moment of access	Represents values over time, snapshots
Serves the clerical community	Serves the managerial community
Can be updated	Is not updated
Run repetitively	Run heuristically
Requirements for processing understood a priori	Requirements for processing not understood a priori
Compatible with the SDLC	Completely different life cycle
Performance-sensitive	Performance relaxed
Accessed a unit at a time	Accessed a set at a time
Transaction-driven	Analysis-driven
Control of update a major concern in terms of ownership	Control of update no issue
High availability	Relaxed availability
Managed in its entirety	Managed by subsets
Non-redundancy	Redundancy is a fact of life
Static structure; variable contents	Flexible structure
Small amount of data used in a process	Large amount of data used in a process
Supports day-to-day operations	Supports managerial needs
High probability of access	Low, modest probability of access

Note. Differences between primitive and derived data. Adapted from *Building the Data Warehouse* (Fourth ed.) p.15, by W. H. Inmon, 2005, New York: Wiley Publishing.

Inmon (2005) describes the characteristics of data in process of data warehousing as:

- *Subject-oriented*: Each type of company has its own unique set of subjects. The data model of DW differs from classical operational systems and is build around specific subject areas of the organization.
- *Integrated*: As the data is fed from multiple, disparate sources into the DW, it is converted, reformatted, resequenced, summarized, and so forth. The result is that data—once it resides in the DW—has a single physical corporate image.

- *Nonvolatile:* Data is updated in the operational environment as a regular matter of course, but DW data is loaded and accessed, but it is not updated (in the general sense). Instead, when data in the DW is loaded, it is loaded in a snapshot, static format. When subsequent changes occur, a new snapshot record is written. In doing so, a historical record of data is kept in the DW.
- *Time-variant:* Time variance implies that every unit of data in the DW is accurate as of some moment in time. In some cases, a record is time stamped. In other cases, a record has a date of transaction. But in every case, there is some form of time marking to show the moment in time during which the record is accurate.

Data Storage and ETL Processes

After the data sources have been studied and an integrated repository has been designed, the "Extraction, Transformation, and Loading" (ETL) process should be designed and implemented. The ETL process is in charge of extracting the data from the sources, correcting the errors, merging data, aligning them to a single codification when the original ones are different, and loading the results to the destination archive. Once the DW layer is available and populated with data, there is a single source of integrated and trusted information about the business carried out by the organization (Boselli et al., 2010).

Kimball and Ross (2002) refer both storage area and ETL as "data staging". They describe the *data staging* as an area dominated by the simple activities of sorting and sequential processing. According to these authors, data staging takes the raw data from operational systems and prepares it for the dimensional model in the data presentation area. "It is a backroom service, not a query service that requires a robust system application" (Kimball & Ross, 2002, p. 358). In this area, data is stored in flat files and/or relational tables.

There is a strong dispute between the authors about the form in which the data should be stored in the DW. One school of thought recommends storing the data directly in the *dimensional form* within the DW. The second school of thought (e.g. Inmon) believes that this approach does not provide a long-term, universal and efficient solution for an enterprise. They claim the data is mutilated and thus not versatile. They advise to use *traditional relational approach* and then add the third purpose-oriented analytical level to which the data is supplied from the DW (Grekov, 2009).

<u>Presentation</u>

Data are organized, stored, and made available for direct querying by users, report writers, and other analytical applications in data presentation area. (Kimball & Ross, 2002) The data presentation area consists of *DMs* containing specialized data based on a single business process that are designed for a specific user group. In the DW community, there are two different views on how the data presentation area should be organized. The Inmon approach consists of an integrated database containing all data, which then serves as a source for *independent* DM. On the other hand, the Kimball approach consists of a series of *integrated* DMs, which are directly fed from the data staging area. Both approaches offer a viable alternative for modeling a DW (Linders, 2008). Figure 5 demonstrates the structure of the data presentation area proposed by Inmon versus the structure proposed by Kimball.

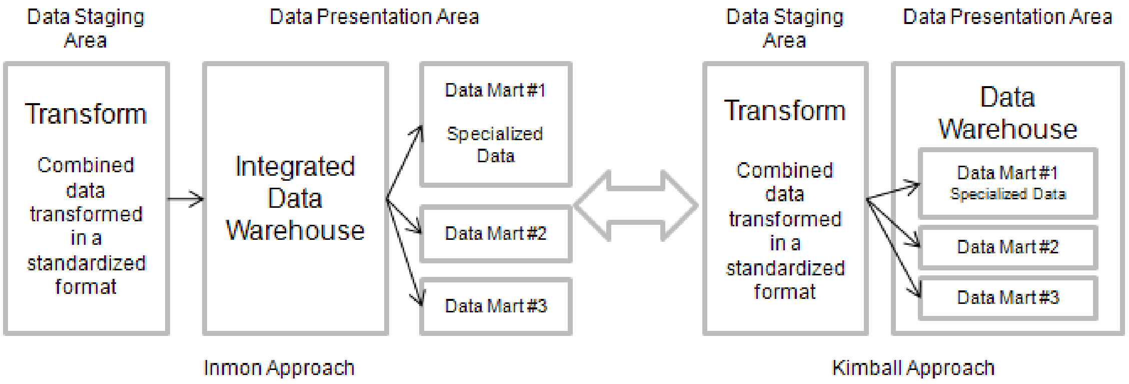


Figure 5. Data presentation area: Inmon vs. Kimball. Reprinted from *Opportunities and Limitations of Using SOA Concepts and Technologies for Building BI Applications: A Delphi Study* (Unpublished master's thesis, University of Twente), (p .9), by S. Linders, 2008. Retrieved February 25, 2011 from http://essay.utwente.nl/58525/.

Data Access Tools

The most visible components of DW infrastructure are analysis and reporting applications, which deliver the information to business users.

Data access tools are variety of capabilities that can be provided to business users to leverage the presentation area for analytic decision making. All data access tools query the data in the DW's presentation area. Querying is the whole point of using the DW (Kimball & Ross, 2002).

The analysis and reporting tools that can be utilized in banking are ad hoc reports, standard reports, scheduled reports, forecasting (predictive analysis) tools, balanced scorecards, dashboards, data mining and OLAP Cubes. A brief description of each concept and/or tool is provided below:

Ad hoc reporting consists of queries that are formulated by the user on the spur of the moment (Kimball & Ross, 2002); it is “one-time-only, casual access and manipulation of data on parameters never before used, usually done in a heuristic, iterative manner” (Inmon, 2005, p. 489).

Standard reports usually have a fixed format, are parameter-driven and, in their simplest form, are prerun. Standard reports provide a core set of information about what's going on in a particular business area (Thornthwaite & Mundy, 2006).

Scheduled reports promote the management of low or noninteractive usage time periods and preparation of required information in possible off-peak times to be retrieved later by business users.

Forecasting (or *predictive analysis*) *tools* such as *graphical trend analysis* enables business users to accurately identify and forecast by using past data to plot a comprehensive picture of future business trends. By graphically representing particular summarized data on a time line the executives can be alerted about current or past trends and they can use this information to determine future development or can start the quest for the underlying reasons for this particular trend (Grekov, 2009).

Dashboards are visual reporting tools to depict business performance usually defined by metrics and time series information, which gives users a snapshot of performance and enables them drill down one or more levels to view more detailed information about a metric. In essence, a dashboard is a visual exception report, highlighting performance anomalies (Eckerson & Hammond, 2011) Thus, BI dashboards provide executives and managers with the ability of making decisions more efficiently by visualizing key business information on a single screen like an automobile dashboard.

Balanced scorecards display key performance indicators (KPIs) current values and the targets for financial, customer, internal systems and human capital categories. The balance scorecard is a summary of key business analytics rolled up to the appropriate level for the user with capabilities to drill down into more detail (Robinson, M., 2008). It integrates financial measures with other key performance indicators around customer perspectives, internal business processes and organizational growth, learning and innovation. To measure overall corporate performance, goals are set for each of these perspectives and then specific measures for achieving such goals are determined (Ghosh & Mukherjee, 2006).

Data mining is one of the most important analytical techniques used in banking to access and analyze the information. It is a decision support process, which is based on artificial intelligence (AI), machine learning, and other technologies; highly automated analysis of the original enterprise data. It helps decision-makers to adjust marketing strategies to reduce risks and make the right decisions (Qihai, Tao & Tao, 2008). Data mining is a class of undirected queries that seek to find unexpected patterns in the data. The most valuable results from data mining are clustering, classifying, estimating, predicting and finding things that occur together. The principal tools of data mining process include decision trees, neural networks, visualization tools, genetic algorithms, fuzzy logic, and classical statistics. Generally, data mining is a client of the DW (Kimball & Ross, 2002).

Jensen (2006) classifies the tasks of data mining in six groups: Classification, estimation, segmentation, forecasting, association and text analysis. Data mining provides banks with opportunities to look for hidden pattern in a group and discover unknown relationship in the data. In banking, *CRM*, *Risk Management* and *Fraud Detection* are the main areas that data mining technology is broadly utilized.

One of the most important data access and analysis tools of BI/DW, *OLAP*, which is the basis for the thesis, will be examined in detail in Chapter 3.

Two Types of Data Warehousing Architecture

The Corporate Information Factory (CIF) by Inmon, which is considered as enterprise-wide DW using a normalized relational model and the Kimball's dimensional DW Bus (BUS) representing the organization's key business processes with a dimensional model (which also constitutes the basis for multidimensional model), are the two main types of data warehousing architecture.

Inmon (2005) introduces four levels of data in the architected environment—the *operational level*, the *atomic* (or the *DW*) *level*, the *departmental* (or the *DM*) level, and the *individual level*. These different levels of data are the basis of a larger architecture called *CIF*. An illustration of these levels with their details is given in Figure 6.

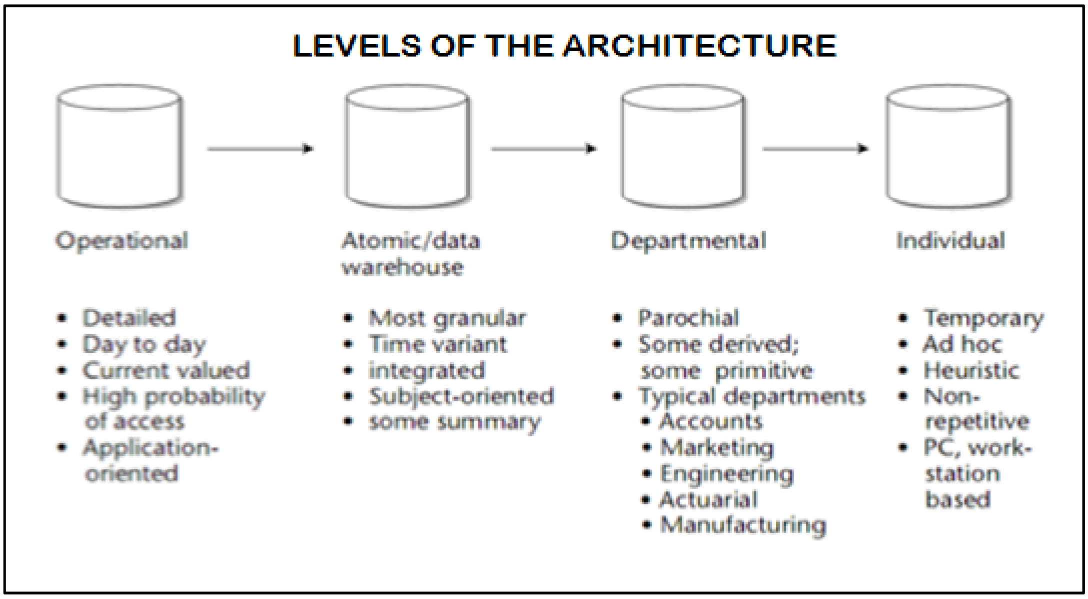


Figure 6. Levels of DW architecture. Reprinted from *Building the Data Warehouse* (Fourth ed.) (p. 16), by W. H. Inmon, 2005, New York: Wiley Publishing.

Inmon (2005) considers that the operational level of data holds application-oriented primitive data only and primarily serves the high-performance transaction-processing community. DW level of data holds integrated, historical, primitive data that cannot be updated. In addition, some derived data are found there. The departmental or DM level of data contains derived data almost exclusively and is shaped by end-user requirements into a form specifically suited to the needs of the department. There is a different data structure for each DM and all of these structures are fed from the granular data found in DW. Heuristic analysis, in which the next step is determined by the results of the current step of analysis, is done in individual level.

Inmon (2005) refers the term *atomic data* as the data with the lowest level of granularity that is stored in DW. He defines *granularity* as “level of detail or summarization of the units of data in the data warehouse” (Inmon 2005, p. 41). The more detail there is, the lower the level of granularity. He considers that granularity is the single most critical design issue in DW environment because it profoundly affects the volume of data that resides in the DW and the types of queries that can be answered. The granular data found in DW is the key to reusability, because it can be used by many people in different ways. With a DW, the different organizations are able to look at the data as they wish to see it. Another benefit of granular data is that it contains a history of activities and events across the corporation. Flexibility is a further advantage of a low level of granularity. However, a very low level of granularity creates too much data and the system is overwhelmed by the volumes of data. A very high level of granularity is efficient to process, but precludes many kinds of analyses that need detail. To handle this issue, DW with dual levels of granularity that serve different types of queries can be built (Inmon, 2005).

Partitioning a DW is the second important issue. When data is partitioned it can be managed in separate, small, discrete units. This means that loading the data into the DW will be simplified; building indexes will be streamlined and archiving data will be easy (Inmon, 2005).

Inmon (2005) believes that a *normalized* or *relational* approach is proper for optimal DW design and the *relational model* is a superior choice while *dimensional (star join) model* has many disadvantages. Dimensional design is not flexible, is not useful as a foundation for reconciliation and is not standing ready for a new set of unknown requirements. But the normalized granular data found in a DW is indeed all of those things.

Kimball and Ross consider that the *bus architecture* is essential for creating an integrated DW from a distributed set of related business processes. The bus architecture is independent of technology and database platform. All flavors of relational and OLAP-based DMs can be full participants in the DW Bus if they are designed around conformed dimensions and facts (Kimball & Ross, 2002). “Data in the queryable presentation area of the data warehouse must be dimensional, must be atomic, and must adhere to the data warehouse bus architecture (…) All the data marts must be built using common dimensions and facts, which we refer to as *conformed.* This is the basis of the data warehouse bus architecture” (Kimball & Ross, 2002, p.12).

Kimball and Ross (2002) describe dimensional modeling concepts as follows:

A *fact table* is the primary table in a dimensional model where the numerical performance measurements of the business are stored. The most useful facts in a fact table are numeric and additive. Fact tables express the many-to-many relationships between dimensions.

Dimension tables are integral companions to the fact tables, which are the textual descriptors of the business. A *dimension table* is a table with a single-part primary key and descriptive attribute columns. They have a highly denormalized structure and they often represent hierarchical relationships. Dimensions are *conformed* when they are either exactly the same (including the keys) or one is a perfect subset of the other (Kimball & Ross, 2002).

The fact table consisting of numeric measurements is joined to a set of dimension tables filled with descriptive attributes. *A star schema* (or *star-join schema*) is “the generic representation of a dimensional model in a relational database in which a fact table with a composite key is joined to a number of dimension tables, each with a single primary key” (Kimball & Ross, 2002, p. 414). Figure 7 illustrates the components of the star join.

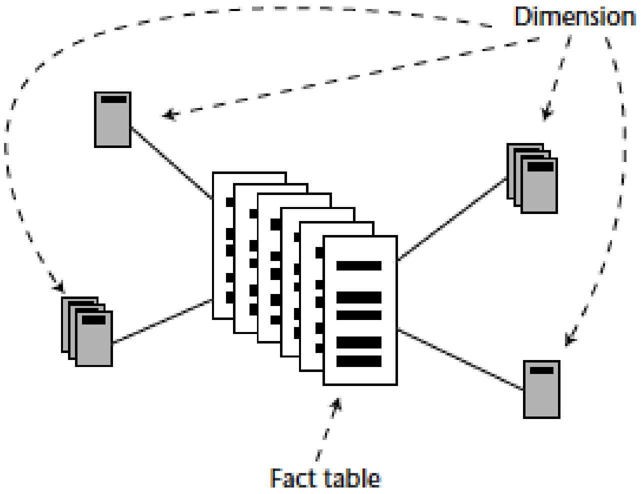


Figure 7. Components of the star join. Reprinted from *Building the Data Warehouse* (Fourth ed.) (p. 360), by W. H. Inmon, 2005, New York: Wiley Publishing.

"In a star schema, the dimension tables do not have references to other dimension tables. If they do, the structure is called a snowflake schema instead" (Westerlund, 2008, p.14). Figure 8 shows a snowflake structure. The fact tables in both a snowflake and star schema would be identical, but the dimensions in a snowflake are normalized, usually under the guise of space savings and maintainability (Kimball & Ross, 2002).

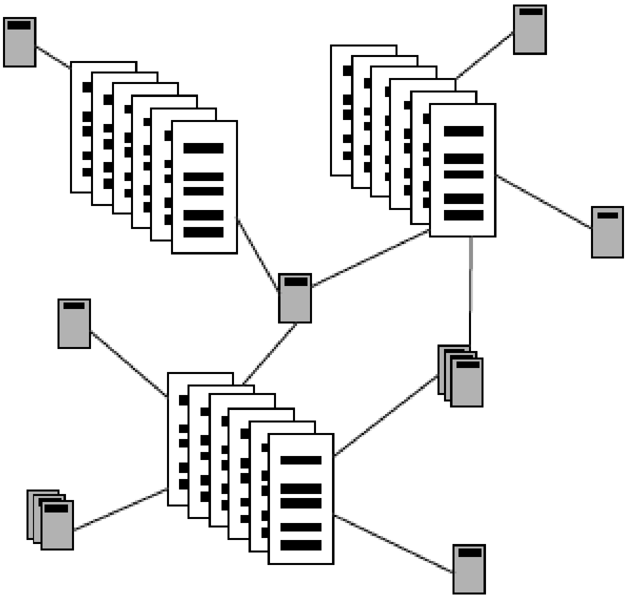
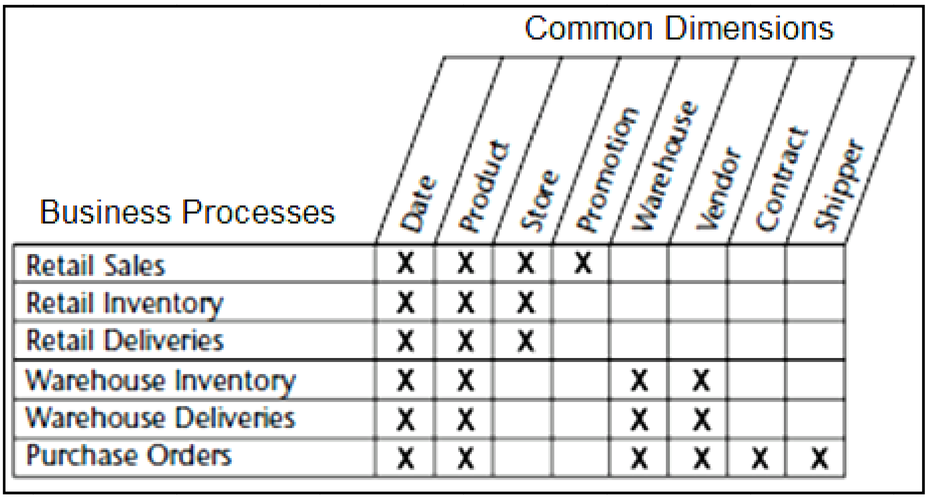


Figure 8. A snowflake structure. Reprinted *Building the Data Warehouse* (Fourth ed.) (p. 361), by W. H. Inmon, 2005, New York: Wiley Publishing.

The tool Kimball uses to create, document, and communicate the bus architecture is DW Bus *matrix,* a sample of which was illustrated in Figure 9.



	Common Dimensions							
Business Processes	Date	Product	Store	Promotion	Warehouse	Vendor	Contract	Shipper
Retail Sales	X	X	X	X				
Retail Inventory	X	X	X					
Retail Deliveries	X	X	X					
Warehouse Inventory	X	X			X	X		
Warehouse Deliveries	X	X			X	X		
Purchase Orders	X	X			X	X	X	X

Figure 9. Sample data warehouse bus matrix. Reprinted from *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (Second ed.) (p. 79), by R. Kimball & M. Ross, 2002, New York: John Wiley & Sons.

The rows of the bus matrix correspond to DMs and the columns of the matrix represent the common dimensions used across the enterprise.

According to Kimball and Ross (2002) a dimensional model contains the same information as a normalized model but packages the data in a format whose design goals are user understandability, query performance, and resilience to change. The ability to visualize something as abstract as a set of data in a concrete and tangible way is the secret of understandability promoted by dimensional structure.

The authors add “It is acceptable to create a normalized database to support the staging processes; however, this is not the end goal. (…) Normalized modeling is immensely helpful to operational processing performance because an update or insert transaction only needs to touch the database in one place. Normalized models, however, are too complicated for data warehouse queries” (Kimball & Ross, 2002, p. 9-11). These authors believe that dimensional models have proved to be understandable, predictable, extendable and highly resistant to the ad hoc attack from groups of business users (Kimball & Ross, 2002).

Inmon (2005) describes most important difference between relational and dimensional structures in terms of *flexibility* and *performance*. According to the author, the relational model is highly flexible, but is not optimized for performance for any user. The dimensional model is highly efficient at servicing the needs of one user community, but it is not good at flexibility (Inmon, 2005, p. 362). On the other hand, Kimball and Ross (2002) believe that the price paid for greater flexibility is often greater complexity.

Inmon (2005) suggests that, in the relational model, data elements can be shaped and reshaped in many different ways. The detailed data are collected and can be combined, many different views of the data can be supported when the design for the DW is based on the relational model. He states that the granular data in the relational model is also used to service unknown future needs for information, not only to meet existing needs of existing user groups.

Storing the atomic data in dimensional structures provides business users with the ability of getting answers to immediate questions and sometimes figures out some unpredictable problems. According to the Kimball approach, this puts usable data in the hands of the business user making the query without requiring a DW expert to drill into the different normalized structures for the data (Dwerek, 2005b).

Kimball and Ross also refute the perceptions and assumed characterization in the industry, which claim that the dimensional models and DMs are: (Kimball & Ross, 2002, p. 24-26) (for full text, see Appendix A):

1. *for summary data only,*
2. *departmental, not enterprise solutions,*
3. *not scalable,*
4. *only appropriate when there is a predictable usage pattern*
5. *those can't be integrated and therefore lead to stovepipe solutions*.

As shown above, while there are some similarities between these two techniques such as the emphasis on level of granularity and need for atomic data, there are some notable differences as well. The primary difference between these two techniques is the normalized data foundation. Another thing that separates these two approaches is the management of atomic data. With Inmon's architecture, atomic data will be stored within a normalized DW. In contrast, the Kimball method states that the atomic data should be placed within a dimensional structure.

Considering the similarities and differences between the two models, Dwerek (2005b, para. 14) answers the question "Which is Better?" as: "(…) it depends—on how you cleanse your data; the level of granularity you choose to access it; the variety of analytical techniques you use to analyze the data, the time and resources you have to build it and your prevailing corporate culture". According to Dwerek (2005a) all enterprises require a means to store, analyze and interpret the data they generate and accumulate in order to implement critical decisions. Corporations must develop operating and feedback systems to use the underlying data means (DW) to achieve their goals. Both the CIF and BUS architectures satisfy these criteria.

Online Analytical Processing (OLAP)

OLAP is considered as a critical category of information technology and a significant tool for high performance and multidimensional analysis of large scale business data.

Multidimensional analysis, the basis for OLAP, is not a new concept. In fact, it goes back to 1962, with the publication of Ken Iverson's book, *A Programming Language (APL)*. APL is a mathematically defined language with multidimensional variables and elegant processing operators (Pendse, 2002). First multidimensional marketing applications were introduced to the market in 1970 and first multidimensional financial application was developed in 1982 (Pendse, 2002). In 1993, the term OLAP was coined in a white paper authored by famous database researcher Ted Codd and his associates, who also established the twelve rules for an OLAP product (Codd, Codd & Salley, 1993).

Definition, Basic Concepts and Features of OLAP

Codd and his associates have submitted, regarding the limitations of relational model that "most notably lacking has been the ability to consolidate, view and analyze data according to multiple dimensions, in ways that make sense to one or more specific enterprise analysts at any given point in time" (Codd et al., 1993, p. 4). This requirement is called *multidimensional data analysis.* In *multidimensional database model* data is presented in multidimensional structure, opposing to tables in a relational database platform. Codd and his associates have predicted in 1993 that OLAP would permeate organizations at all levels, empowering managers to provide more timely strategic and tactical direction in accordance with the increasing number of internal and external factors impacting contemporary business enterprises. The quality of strategic business decisions made as a result of OLAP would be significantly higher and more timely than those made traditionally (Codd et. al, 1993).

Codd and his associates (1993, p.12) have determined twelve rules for evaluating OLAP systems (for full text, see Appendix B):

1. Multidimensional Conceptual View
2. Transparency
3. Accessibility
4. Consistent Reporting Performance
5. Client-Server Architecture
6. Generic Dimensionality
7. Dynamic Sparse Matrix Handling
8. Multi-User Support
9. Unrestricted Cross-dimensional Operations
10. Intuitive Data Manipulation
11. Flexible Reporting
12. Unlimited Dimensions and Aggregation Levels

Nigel Pendse (2005), the famous OLAP analyst, suggests an alternative term and definition for OLAP: *Fast Analysis of Shared Multidimensional Information* (*FASMI*). He summarizes the OLAP definition with the following key words (Pendse, 2005):

- *Fast* means that the system is targeted to deliver most responses to users within about five seconds, with the simplest analysis taking no more than one second and very few taking more than 20 seconds.
- *Analysis* means that the system can cope with any business logic and statistical analysis that is relevant for the application and the user, and keep it easy enough for the target user.
- *Shared* means that the system implements all the security requirements for confidentiality (possibly down to cell level) and, if multiple write access is needed, concurrent update locking at an appropriate level.
- *Multidimensional*, which is the key feature, means that the system must provide a multidimensional conceptual view of the data, including full support for hierarchies and multiple hierarchies, as this is certainly the most logical way to analyze businesses and organizations.
- *Information* refers to all of the data and derived information needed, wherever it is and however much is relevant for the application.

According to Thomsen (2002, p. 24) "the term OLAP (…) denotes descriptive modelling for analysis-based decision-oriented information processing (ABDOP)". The functional requirements for OLAP are:

- Rich dimensional structuring with hierarchical referencing
- Efficient specification of dimensions and dimensional calculations

- Separation of structure and representation
- Sufficient speed to support ad hoc analysis
- Multi-user support (Thomsen, 2002, p. 5)

OLAP Council defines OLAP as “a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user” (OLAP: On-Line Analytical, n.d., para .1). The Council states (OLAP: On-Line Analytical, n.d.) that OLAP functionality is characterized by dynamic multidimensional analysis of consolidated enterprise data supporting end user analytical and navigational activities including:

- Calculations and modeling applied across dimensions, through hierarchies and/or across members
- Trend analysis over sequential time periods
- Slicing subsets for on-screen viewing
- Drill-down to deeper levels of consolidation
- Reach-through to underlying detail data
- Rotation to new dimensional comparisons in the viewing area

Databases, which use relational model, can perform large amounts of small transactions, keeping the database available and the data consistent at all time. The *normalization* helps keeping the data consistent, but it also introduces a higher degree of complexity to the database, which causes huge databases to perform poorly when it comes to composite aggregation operations. In the context of business it is desirable to have historical data covering years of transactions, which results in a vast amount of database records to be analyzed. The performance issues will arise when processing analytical queries that require complex joining on such databases. Another issue with doing analysis with the relational model is that it requires complex queries, particularly composed for each request. OLAP has been proposed to handle these issues (Westerlund, 2008).

Thomsen (2002) considers that it was particularly difficult in relational model to define decision support applications that depended on *complex data aggregation*. Many efforts made in building higher-level abstractions into the relational model, adding abstraction capabilities and notions of hierarchy were added to relational concepts. Products were built to extend the capabilities of RDBMS. However, these works such as graphical queries, natural language queries and arranged end-user queries were all attempts to hide the *complexity* from the end-user. According to Thomsen (2002), conducting most basic operations of analysis such as *comparisons* is also difficult in relational model. Codd and his associates stated in 1993 that the RDBMS were never intended to provide the very powerful functions for data synthesis, analysis, and consolidation that were being defined as multidimensional data analysis. These types of functions were always intended to be provided by separate, end-user tools that were outside and complementary to the RDBMS products (Codd et al.1993).

In Table 2 OLAP is compared with relational reporting.

Table 2. A Comparison Between OLAP Analysis and Relational Reporting.

TOPIC	OLAP	RELATIONAL
Data Analysis	OLAP provides for online interaction with the data. Users can investigate relationships within the data with simple navigation tools. OLAP also provides context, relevance and visualization of the data.	Some relational reporting can provide limited interactivity. Overall not nearly as robust for analysis.
Numeric Calculations	OLAP technology is based upon calculations and aggregations. Cubes are very good at providing very complex calculations.	Relational reporting can also provide numeric calculations, but complex calculations are oftentimes difficult to implement.
Formatted Reports	OLAP client tools usually provide for very limited formatting capabilities. For the most part, OLAP should be used for interactive analysis, not formatted hard copy reports.	Relational reporting tools are very good at providing report formatting. Relational reporting is well suited to nicely formatted hard copy reporting.
Query Performance	OLAP usually provides for very fast query performance. The usual OLAP query is returned in under 4 seconds.	Relational reporting can be fast but oftentimes it is slow. (This is dependent on the underlying schema)
Textual Data	Although some OLAP tools will provide a means at viewing textual meta-data, overall OLAP is not the tool for storing or providing textual data. Example: OLAP cubes are not designed to store names, addresses, contact information, etc.	Relational Reporting is well suited to handle textual information.
Database Maintenance	OLAP may provide less maintenance since all aggregations are automatically provided within the cube.	Relational Reporting can be maintenance intensive if numerous "aggregation tables" are necessary.
Ad hoc Query Creation	Most OLAP client tools make it very easy to create new "views" of the data. There are no complex joins to create. Most users can easily create their own analytic views.	Depends on how the underlying database is designed. Oftentimes user can become frustrated with table joins.

Note. A comparison between OLAP analysis and relational reporting. Adapted from OLAP Analysis vs. Relational Reporting. (n.d.). *OLAP Business Solutions.* Retrieved March 9, 2011, from http://www.obs3.com/olap_vs_relational.shtml.

OLAP Cube

If the dimensional models are presented within a RDBMS, then these dimensionally modeled tables are referred to as *star schemas.* If dimensional models stored in an OLAP database, they're commonly referred to as *Cubes* (Kimball et al., 2008). *Cubes*, as the core of OLAP, are the structures specially designed to retrieve, analyze and report data efficiently on the OLAP database platform.

An OLAP Cube consists of numeric facts called *measures*, which are categorized by *dimensions*. The cube *metadata* is typically created from a *star schema* or *snowflake schema* of tables in a relational database. *Measures* are derived from the records in the *fact tables* and dimensions are derived from the *dimension tables* (OLAP Functionality, n.d.). *Measures* are the items to be counted, summarized and aggregated and they reference the dimensions in the cube so the facts can be grouped by the dimensional data (Westerlund, 2008). A sample illustration of an OLAP Cube can be seen in Figure 10.

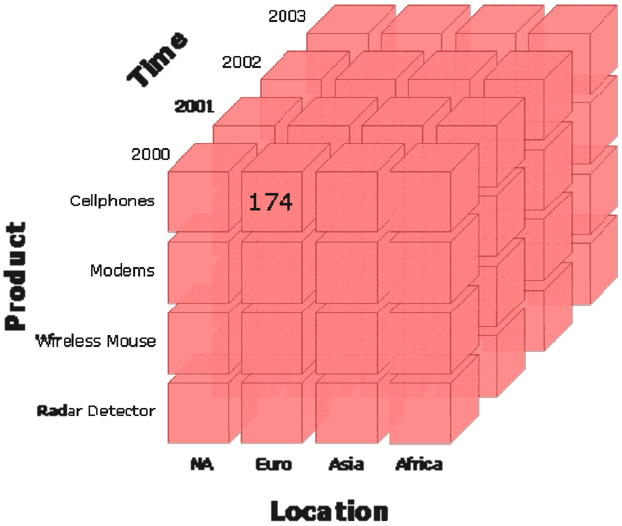


Figure 10. Sample OLAP Cube. Reprinted from Cubes (n. d.). *On-Line Analytical Processing (OLAP) Tutorial*. Retrieved March 5, 2011 from http://training.inet.com/OLAP/Cubes.htm.

Hierarchy is a significant concept related to OLAP Cube, which refers a method to organize elements in a dimension. Thomsen (2002, p. 584) defines hierarchy as "an organization of members into a logical tree structure, with each member having at most one 'parent' member and an arbitrary number of 'children' members".

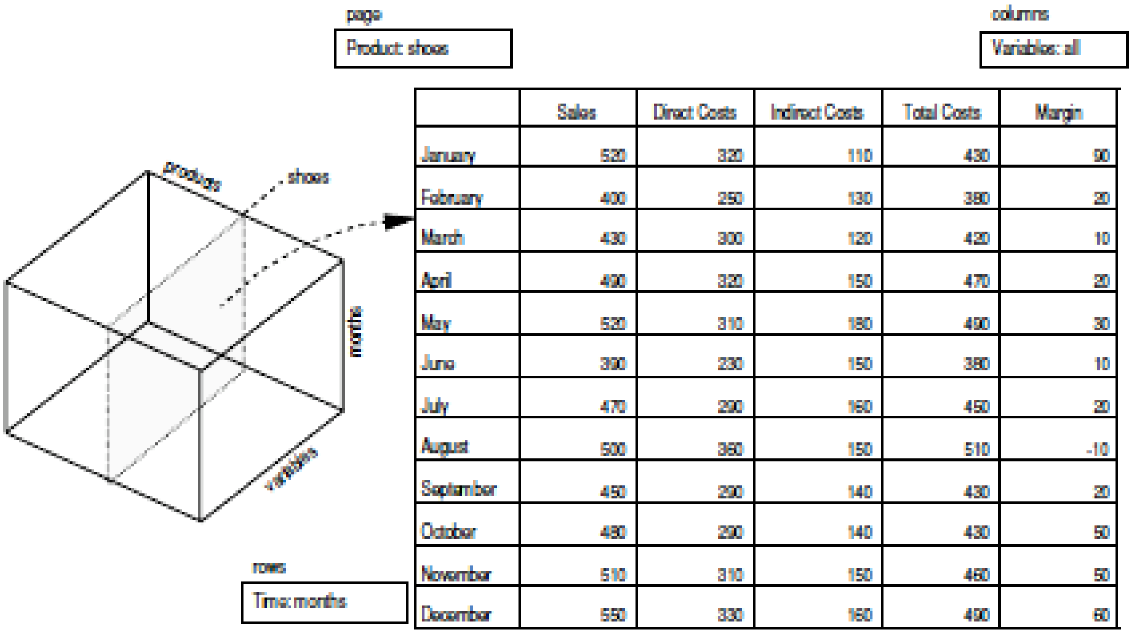
Sparse is "a fact table that has relatively few of all the possible combinations of key values" and *sparsity* is "a situation that occurs when an aggregate table is created that is not appreciably smaller than the table on which it is based" (Kimball & Ross, 2002, p. 413). Sparsity refers to the extent to which a measure contains null values. A null value can occur when there is no measure value for a particular combination of the measure's dimension and takes up a bit of storage space and in addition add to the time required to perform an aggregation (Ritmann, 2005). The opposite of the term sparsity is called *denseness*. A multidimensional database is *dense* if a relatively high percentage of the possible combinations of its dimension members contain data values (OLAP: On-Line Analytical, n.d.).

Even though the term "cube" derives from the geometric figure with three dimensions, an OLAP cube may have more dimensions than three.

OLAP Cube Operations

OLAP cube operations include *slicing* and *dicing*, *drill-down analysis, rolling up and pivoting.*

Slicing and dicing: Slicing and dicing is the process of separating and combining data in seemingly endless combinations and provide the ability to access a DW through any of its dimensions equally (Kimball & Ross, 2002) . This process allows the manager to have many different perspectives of the activities (Inmon, 2005). Slicing and dicing is a user-initiated process of navigating by calling for page displays interactively, through the specification of slices via rotations and drill down/up. (OLAP: On-Line Analytical, n.d.). Figures 11 and 12 illustrate slicing and dicing operations.



	Sales	Direct Costs	Indirect Costs	Total Costs	Margin
January	520	320	110	430	90
February	400	250	130	380	20
March	430	300	120	420	10
April	490	320	150	470	20
May	520	310	180	490	30
June	390	230	150	380	10
July	470	290	160	450	20
August	500	360	150	510	-10
September	450	290	140	430	20
October	480	290	140	430	50
November	510	310	150	460	50
December	550	330	160	490	60

Figure 11. Slice of the cube. Reprinted from *OLAP Solutions: Building Multidimensional Information Systems* (Second ed.) (p. 51), by E. Thomsen, 2002, New York: John Wiley & Sons.

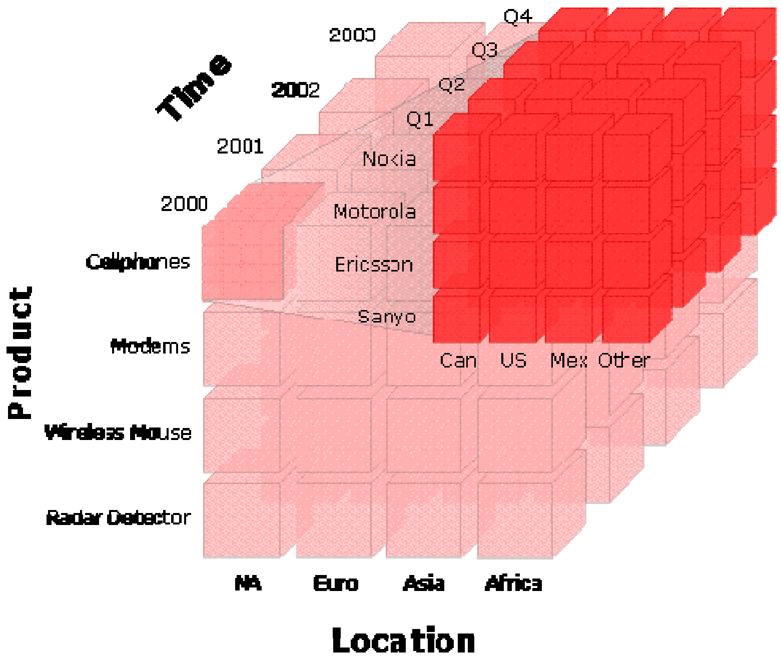


Figure 12. Dicing. Reprinted from Cubes (n. d.).*On-Line Analytical Processing (OLAP) Tutorial*. Retrieved March 5, 2011 from http://training.inet.com/OLAP/cubes.htm.

Drilling down/up: It is necessary to be able to *drill down* on data in order to do slicing and dicing. “Drilling down refers to the ability to start at a summary number and to break that summary into a successively finer set of summarizations” (Inmon, 2005, p. 243). Drilling down or up is a specific analytical technique whereby the user navigates among levels of data ranging from the most summarized (up) to the most detailed (down). The drilling paths may be defined by the hierarchies within dimensions or other relationships that may be dynamic within or between dimensions (OLAP: On-Line Analytical, n.d.). Figure 13 shows a simple example of drill-down analysis.

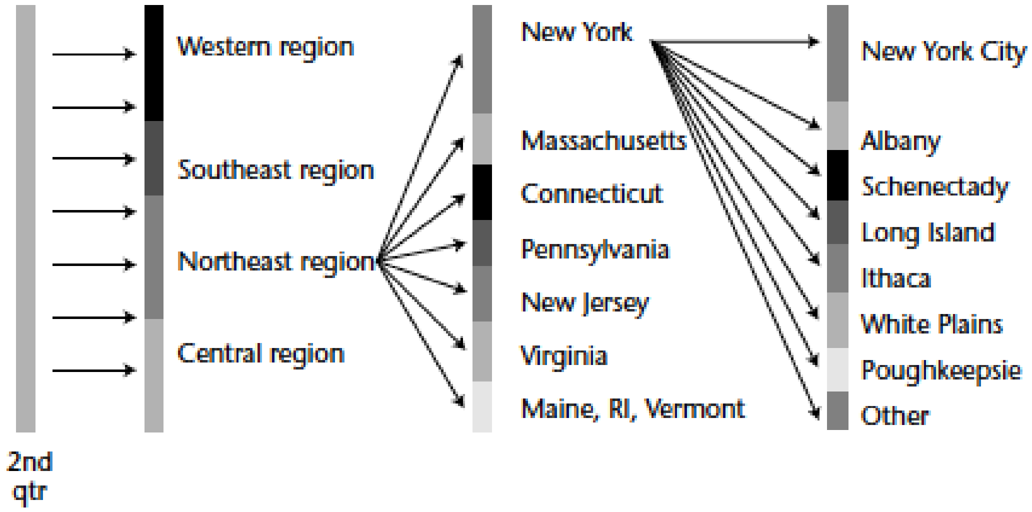


Figure 13. A simple example of drill-down analysis. Reprinted from *Building the Data Warehouse* (Fourth ed.) (p. 243), by W. H. Inmon, 2005, New York: Wiley Publishing.

Rolling up: Rolling up means to present higher levels of summarization (Kimball & Ross, 2002); to increase the level of aggregation (Chaudhuri & Dayal, 1997) and involves computing all of the data relationships for one or more dimensions. A computational relationship or formula might be defined to do this (OLAP: On-Line Analytical, n.d.). The term "aggregation" refers combining two or more data items into a single item such as summing a series of numbers (Thomsen, 2002); precalculating summary data values. *Aggregates* are physical rows, which almost always created by summing other records in the database for the purpose of improving query performance and sometimes referred to as *precalculated summary data* (Kimball & Ross, 2002). Roll-up process fills the new data into the aggregate on a cube.

Pivoting (Rotating): Pivoting refers "rearranging the orientation of logical dimensions on the screen at query time (Thomsen, 2002, p. 586). This operation is performed to provide an alternative presentation of data. Following figure illustrates pivoting operation:

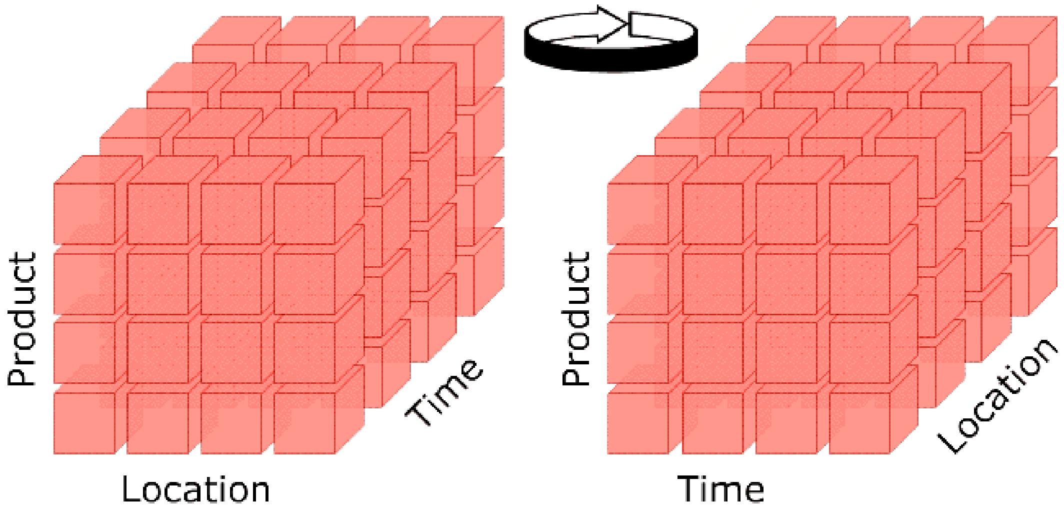


Figure 14. Pivoting. Reprinted from Cubes (n. d.).*On-Line Analytical Processing (OLAP) Tutorial*. Retrieved March 5, 2011 from http://training.inet.com/OLAP/cubes.htm.

Types of OLAP Architectures

There are three major types of OLAP tools that differ with respect to their functionality and architecture: *MOLAP, ROLAP, HOLAP*.

Multidimensional OLAP (MOLAP): MOLAP systems, as the classic form of OLAP, are "dedicated online analytical processing implementations not dependent on relational databases" (Kimball & Ross, 2002, p. 407). These structures directly support the multidimensional view of data through a multidimensional storage engine. This makes it possible to implement front-end multidimensional queries on the storage layer through direct mapping (Chaudhuri & Dayal, 1997). In MOLAP model, data are structured into proprietary formats in accordance with a client's reporting requirements with the calculations pre-generated on the cubes. These models are capable to perform complex calculations; provide users the ability to quickly write back data into a data set and optimized for fast query performance and retrieval of summarized information

(Types of OLAP, n.d.). According to Trepte (1997), MOLAP tools bring value when processing information with consistent response time regardless of level of summarization or calculations selected, avoiding many of the complexities of creating a relational database to store data for analysis and fastest possible performance are needed.

MOLAP systems are considered as less scalable due to the capability of handling only a limited amount of data (Types of OLAP, n.d.); all needed information cannot be stored in MOLAP database (Trepte, 1997). According to Chaudhuri and Dayal (1997, p. 8) "Such an approach has the advantage of excellent indexing properties, but provides poor storage utilization, especially when the data set is sparse". On the other hand, Kimball and Ross (2002, p. 407) state, "Although MOLAP systems do not scale to the sizes that relational databases systems can, they typically offer better performance and more tightly integrated tools than their relational counterparts".

Relational OLAP (ROLAP): ROLAP systems "work primarily from the data that resides in a relational database; where the base data and dimension tables are stored as relational tables. This model permits multidimensional analysis of data as this enables users to perform a function equivalent to that of the traditional OLAP slicing and dicing feature" (Types of OLAP, n.d., para. 2). ROLAP databases provide a multidimensional front end that creates queries to process information in a relational format and the ability to transform two-dimensional relational data into multidimensional information (Trepte, 1997 According to Westerlund (2008), it is better to read data from the data warehouse directly, than to use another kind of storage for the cube. Data can be aggregated and pre-calculated in ROLAP too, using materialized views, i.e. storing the aggregations physically in database tables as in MOLAP. ROLAP architecture is more flexible since it can pre-calculate some of the aggregations, but leave others to be calculated on request.

Advantages of ROLAP are better scalability, efficiency in managing both numeric and textual data. However, ROLAP applications display a slower performance as compared to MOLAP (Types of OLAP, n. d.). According to Trepte (1997), those who need to perform analysis on large volumes of data, to perform detailed what-if analysis based on multiple scenarios should use MOLAP architecture.

Hybrid OLAP (HOLAP): Thomsen (2002) considers that the relational database products are far better equipped to handle the large amounts of data typically associated with corporate data-warehousing initiatives; multidimensional databases are far better equipped to provide fast, dimensional-style calculations. Thus, most organizations need some blend of capabilities, which would be HOLAP (Hybrid OLAP).

HOLAP model attempts to incorporate the best features of MOLAP and ROLAP into a single architecture. HOLAP systems store larger quantities of detailed data in the relational tables while the aggregations are stored in the pre-calculated cubes. Some of the advantages of this system are better scalability, quick data processing and flexibility in accessing of data sources (Types of OLAP, n.d.). HOLAP architecture can provide quick and good pretreatment measurement (Qihai, Tao & Tao, 2008).

There are also less popular types of OLAP existing in the industry. One of them is *Desktop OLAP* (*DOLAP)*, which based on the idea that a user can download a section of the data from the database or source, and work with that dataset locally, or on their desktop. Another type is *Web OLAP (WOLAP)* that pertains to OLAP application, which is accessible via the web browser (Types of OLAP, n.d.).

OLAP Cubes in Banking

In recent years, OLAP Cubes are widely employed in banking all over the world. The possibility of fast analysis of multidimensional data increases the quality of managerial reporting. As shown above, OLAP cubes are considered as specifically developed tools for the queries in large databases with many relations involved, such as those in banks and optimized for short query response time. OLAP systems are utilized to discover trends and to analyze critical factors and to keep the banking managers informed about the business situation.

Kimball and Ross (2002) suggest that multidimensional OLAP products *naturally fit* for financial analysis. According to these authors, OLAP cubes are precalculated, which results in fast query performance that is critical for executive use. OLAP is well suited to handle complicated organizational roll-ups, as well as complex calculations, including inter-row manipulations (Kimball & Ross, 2002).

CPM is one of the most important tasks that OLAP Cubes promote. CPM takes a holistic approach to the implementation and monitoring of strategy. It combines business methodologies, processes and systems, which are the technology solutions. A CPM system enables a closed-loop process that starts with understanding where the organization is today, where it wants to go to, what targets should be set, and how resources should be allocated to achieve those targets. Once plans have been set, the system then monitors the performance of those plans, highlights exceptions, and provides insight as to why they occurred. The system supports the evaluation of alternatives from which decisions can be made. CPM tools deliver the right information to the right people at the right time in the right context (Coveney, 2003).

Inmon (2005) brings the function of these analytical tools in compliance auditing to the attention. According to the author, the world of banking has always had a high degree of compliance to data-integrity standards ; banks and financial institutions have always had strict procedures and controls to ensure that funds are handled properly The auditing of financial activities requires detailed, historical data be kept.. OLAP cubes can meet such kind of special needs of banks as well (Inmon, 2005).

In this Chapter, it was aimed to establish a conceptual framework to define, describe, classify and organize existing information related to subject of the thesis. The components and architecture of BI; business benefits and utilization of BI in banking; differences between relational and multidimensional DB design in BI/DW processes; OLAP concepts, features and operations; differences and similarities between relational and OLAP reporting; the benefits of using OLAP cubes in banking were examined in the context of a number of theories. Discussions on definition and description of DW processes were reviewed; two types of DW architectures, relational CIF and dimensional BUS models were compared as well in terms of performance, complexity, flexibility and efficiency. The differences between OLAP analysis and relational reporting were also examined on the basis of multidimensionality, speed, consistency, analytical aspects, query performance, maintenance and reporting capabilities along with the explanations of OLAP cube types and operations.

CHAPTER 3

LITERATURE REVIEW: RELATED WORKS

Related Academic Studies

In Chapter 2, common theoretical approaches to BI and OLAP applications were reviewed. Although several resources were utilized, a limited number of them have been based on academic researches. Most of the studies contribute commercial interests rather than academic purposes.

In this chapter, it's aimed to introduce the academic studies related the purpose of this thesis. However, it was shown that, the number of such kind of studies is very limited; actually there is no academic study that has been conducted exactly on the topic of *OLAP Cube reporting in banking.* Very few studies have particularly examined the impacts of BI implementation in banking. Even, the number of the studies examining OLAP implementations from technical perspective on such large-scale data in any business area is limited. None of them has investigated the role of OLAP cube utilization in managerial reporting for banking in technical manner. However, a number of academic studies related the issue of the thesis in general, will be presented under three categories:

1. *Studies focusing on maturity, readiness and usage levels of BI*: Certain group of academic researches examines maturity, readiness and usage levels of BI tools within various sectors in several countries.

Pirttimaki's (2007) doctoral dissertation examines BI as a tool for managing business information in fifty large Finnish companies. This study uses existing theories together with empirical material and intends to confirm theoretical framework for BI and to improve on the general knowledge of BI and its evolution, state and usage in these companies. In addition, it examines the methods for measuring BI, integration of a BI process into strategic management and utilization of human-source intelligence. The results presented in the dissertation suggest that BI is becoming an integral part of these companies' activities and they view BI not only a defensive tool to ward off perceived threats and changes but also a proactive management tool for uncovering new business opportunities and trends. However, fifty top Finnish companies feel that BI currently is not systematic or comprehensive enough. The lack of BI metrics and measurements and bulk BI investments focused on technical details at the expense of human elements disaffect with BI. Even though Pirttimaki's dissertation is a comprehensive study determining usage and limitations of BI processes in large companies, it doesn't put particular emphasis on technical issues with BI and utilization of OLAP technologies.

Another study conducted by Hindriks (2007) focuses on the usage and maturity of BI in Dutch retail sector through a theoretical review and an empirical research. The results of the research demonstrate that the maturity and usage of BI at Dutch retail sector are not sufficient to fulfil the trends that require chain wide BI. Even though almost all Dutch retail organizations involved in the study use BI, very few of them evolve BI to an integral system, which is needed to be ready for chain wide BI. The study shows that BI mostly used for reporting and OLAP. Hindriks examines the usage of OLAP in these companies, however, she doesn't provide an evaluation on the opportunities or limitations of using OLAP technologies in retail sector and doesn't approach the issue in technical manner.

On the other hand, Tabatabaei (2009) evaluates the maturity level of BI in banking industry. His master's thesis attempts to measure and introduce readiness/maturity levels of BI processes in the Iranian banks by drawing upon the concepts underlying Capability Maturity Model Integration (CMMI) and examines the key areas of improvement in BI operations, benefits gained from BI as well as the strength point of Iranian banking industry through a survey. Further, his study establishes a model for BI with the factors influencing the processes. According to empirical findings of his study, the bank characteristics (such as ownership, size, number of branches, age of the companies) have an influence on BI maturity level. The maturity level of BI as a whole process in Iranian banking industry is at level three, but DW, extraction and OLAP capabilities should be improved firstly to improve the maturity level. The usage levels of OLAP in Iranian Banks have also been examined in this study; however, implications of OLAP applications in Iranian banks have not been evaluated in detail.

2. *Studies focusing on the influence of BI tools on banks' performance*: A group of researches investigates the influence of BI implementation on the performance of the bank.

Bach, Strugar & Jaković (2007) examine the present implementation of BI tools in Croatian banking system and the possibility of their improvement for the purpose of lowering operational cost of banks and maintaining the stability of the banking system. A survey research conducted by the authors, whom concentrated on data mining and data warehousing, revealed that only forty-six percent of Croatian banks use both of the BI tools. Banks which use BI tools differ from the banks which do not have such a system in the following characteristics: size of total assets, participation of their own assets in the Croatian banking sector, size of off-balance items, size of income and capital stock and rate of capital adequacy. In other words, project results show that banks, which use BI tools, are larger and more successful. Large and successful banks invest more in information technology, especially BI in the purpose of more efficient business reporting. As shown above, the consequences of using OLAP tools are not discussed specifically in this study and it doesn't involve technical assessments on BI implementation at Croatian banks.

Zavareh (2007) investigates in his master's thesis the role of analytical CRM in maximizing customer profitability in private banking. In order to accomplish this objective, a multiple-case study was conducted, which consisted of two cases comprising two leading banks with large market share in private banking in Sweden. These banks utilize OLAP technology in analytical CRM operations. The findings of the research show that CRM deployment is positively related to facilitating profitable relationships and establishing long-term relationships. However, this study focuses only CRM operations in banking, which are not conducted directly for reporting.

3. *Studies focusing on performance and capabilities of OLAP implementations from technical perspective*: The last group of studies examines the performance issues about OLAP processes and some of them suggest particular techniques to improve query performance and/or reporting capabilities of OLAP tools.

Tam's master's thesis (1998) intends to investigate efficient methods for computing cubes and for using them to support OLAP and data mining. According to the author, MOLAP systems are generally more efficient than ROLAP systems when the sparse data cube techniques are applied to solve sparse matrix problem or when the data sets are small to medium sized. A HOLAP system has been developed, which combined better features of MOLAP and ROLAP in this study. Tam suggests depending on performance tests done during the study that HOLAP cube is faster than plain MOLAP and ROLAP systems in many cases. Another algorithm for the computation of aggregation designed as well. The thesis also discusses the OLAP mining (data mining integrated with OLAP). An OLAP mining module in the MOLAP system has also been developed to assist users to discover different kinds of knowledge from the data; to explore the data flexibility from different angles and at multiple abstraction levels. According to Tam's study OLAP-based mining has more advantages when compared with mining directly from raw data.

Westerlund (2008) also examines and compares these two different approaches, ROLAP and MOLAP by implementing a prototype to make relevant data available to all employees of a consulting company's system. According the findings of Westerlund's study, the relational database has its strengths, but it doesn't perform very well with complex queries and analysis of very large sets of data. OLAP is a requirement for large-scale data analysis. ROLAP is generally the better choice for data cubing. It is quite possible to implement data cubes directly in a relational database engine without modifications, but the performance would be unacceptable when the database is very large with many relations involved and the reporting would be limited.

Westerlund concludes that the need for multidimensional data analysis for a smaller organization with a limited database may not require all the extensive capacity of OLAP tools, which often are expensive even though there are open source alternatives for BI solutions as well.

However, Westerlund's study has been conducted only on a small-scale data and not for a bank's management reporting.

Another study by O'Neil and Quass (1997) presents a review of current indexing technology used in data warehousing, including row-set representation by Bitmaps, introduces two approaches called *Bit-Sliced indexing* and *Projection indexing* and then examines their performance for evaluating a variety of aggregate functions and (range) predicates, and showing circumstances in which type of index out-performs the others. The algorithms that become feasible with these variant index types against algorithms using more conventional indexes are compared. The analysis demonstrates important performance advantages for variant indexes in some types of SQL (Structured Query Language) aggregation, predicate evaluation, and grouping. The study concludes by introducing a new method (Groupset indexes) whereby multidimensional Group By queries, reminiscent of OLAP or Datacube queries but with more flexibility, can be very efficiently performed. As a new contribution, the authors examine how OLAP-style queries involving aggregation and grouping can be efficiently evaluated using indexing and clustering. According to their findings, using indexes, as opposed to precalculated summary tables, to evaluate OLAP-style queries allows more flexible queries, having ad-hoc selection conditions, to be evaluated efficiently. This paper constitutes the first rigorous examination of some index structures for OLAP cube queries in the literature.

One of the studies in the field of OLAP, the doctoral dissertation by Karayannidis (2003) has been carried out to investigate a data structure that would provide an efficient storage base for cubes as well as access path for the most detailed data, in order to support the processing of ad hoc OLAP queries. The objectives have been implemented into a real OLAP system (*ERATOSTHENES)* and a specific file organization (the *CUBE* file) natively supporting dimension hierarchies has also been proposed by the author. This multidimensional data structure imposes a hierarchical clustering on the data and thus is intended for speeding up queries with hierarchical restrictions, which constitute the most typical workload in OLAP. Karayannidis presents the architecture of this system and describe the implementation of various fundamental components, such as the storage manager and the processing engine. This work sets a new paradigm for the storage and processing of multidimensional data with hierarchies and according to Karayannidis, this new processing framework has opened the road for important optimizations on the evaluation of star queries such as the hierarchical pre-grouping transformation exhibiting speed-ups up to forty-two times faster than the conventional plans. With the structures like implemented CUBE file structure, the evaluation of the costly *star-join* becomes a simple multidimensional range query, which is evaluated very efficiently to the native support for many dimensions. However, the study examines the performance of cube itself not its implications in operations of an institution.

Guo's master's thesis (2009), namely "Partial Aggregation and Query Processing of OLAP Cubes" is another empirical academic study on OLAP, which has been conducted to demonstrate the techniques improving the query performance of OLAP cubes, reducing the cube size and shortening the cube building time. A SplitCube approach for partial pre-aggregation has been proposed, the guidelines in how to choose the right algorithm in different user cases are given. Observed patterns that are useful in choosing the right algorithm are as follows:

- QPRQ (Query Processing with Range Queries) and MQPSB (Merging of Query Point Set and B-tree) should always be picked as a better choice in cube construction and query processing.
- In choosing between QPRQ and MQPSB, the number of cubelets should be taken into account. If the size of the cube is huge and a big number of cubelets is

inevitable, MQPSB should be considered; otherwise, QPRQ should perform better than MQPSB in query processing.

However, these techniques do not establish a framework for banking applications utilizing OLAP.

Conclusions from the Review of Theory and Literature

A number of theories and related academic works were presented in Chapter 2 and 3. Even though none of them examines the issues that are entirely similar to the subject of this thesis, they had an important function as a basis for conceptual analysis that forms theoretical and qualitative part of the study.

The inferences, hypotheses, assumptions and discussions emerged from the review of theory and literature which constitute a frame of reference for the empirical part of the thesis can be concluded as follows:

- Different definitions and classifications for BI had been offered in literature, but, there is no clear and commonly shared approach and methods of BI, such as seen in the debate between Inmon (2005) and Kimball and Ross (2002) on BI and DW.
- According to most common definition, BI is the process of deriving and gathering valuable and meaningful information from raw data that provides people with the ability to analyze and easily share this information with others and to draw conclusions or make assumptions. Providing managers with new tools allowing them to see data in new ways empowers them to make faster and better decisions.
- The components of BI are, data collection, data integration (ETL), data warehousing, data analysis and presentation. Data is stored in DW or DMs in BI process, which are databases developed to process integrated information.

- Business benefits of BI utilization are summarized as: lowering costs, increasing revenue and improving customer satisfaction.
- BI tools meet basic needs of banking by providing necessary reporting and data analysis infrastructure such as real-time reporting and performance monitoring of key business metrics and budget performance, temporal analysis of basic business measures, effective segmentation of available information and implementation of different business scenarios, information exchange required to ensure compliance to all governmental and international regulations.
- The specific areas that BI information tools promote in banking are product/service profitability, CRM, ABC, customer profiling and segmentation, risk management, sales and marketing, fraud detection, tracking and identification of anti-money laundering.
- The term "data warehousing" refers to process of building and using DW. However, there is a strong dispute on definition of DW between the two leading authors and constructors of data warehousing. Inmon (2005) refers DW as core of the whole system where data is stored and accessed for analysis while Kimball and Ross (2002) believe DW is a complete solution for analytical processing and decision making. Due to this conceptual debate, many studies refer the processes as "DW/BI".
- Kimball and Ross (2002) introduce four separate and distinct components of DW: Operational Source Systems, Data Staging Area, Data Presentation Area and Data Access Tools and refer both storage area and ETL as "data staging".
- Kimball and Ross (2002) suggest that the data presentation area consists of *DMs*, which are directly fed from the data staging area and containing specialized data

based on a single business process that are designed for a specific user group. Inmon's approach consists of an integrated database containing all data, which then serves as a source for *independent* DMs.

- Data access tools are components of DW infrastructure, which deliver the information to business users. Those are ad hoc reporting, standard reports, scheduled reports, forecasting tools, dashboards, balanced scorecards, data mining and OLAP cubes.
- Data warehousing can be conducted in different ways such as an enterprise-wide DW (CIF, suggested by Inmon)(Inmon, 2005) or dimensional bus architecture representing the organization's key business processes (BUS, suggested by Kimball)(Kimball & Ross (2002). The *operational level*, the *atomic* (or the *DW*) *level*, the *departmental* (or the *DM*) level and the *individual levels* of data are the basis of *CIF,* a larger architecture. Kimball and Ross consider that the bus architecture is independent of technology and the database platform; all flavors of relational and OLAP-based DMs can be full participants in the DW Bus if they are designed around conformed dimensions and facts.
- *Granularity* is defined as level of detail or summarization of the units of data in the DW. The more detail there is, the lower the level of granularity. Granularity affects the volume of data that resides in the DW and the type of query that can be answered. Flexibility is another benefit of a low level of granularity according to Inmon (2005). However, a very low level of granularity creates too much data, and the system is overwhelmed by the volumes of data. A very high level of granularity is efficient to process, but precludes many kinds of analyses that need detail. To handle this issue, DW with dual levels of granularity that serve different types of

queries can be built or a living sample database where statistical processing can be created.

- *Partitioning* a DW is the second important issue. When data is partitioned it can be managed in separate, small, discrete units. This means that loading the data into the DW will be simplified; building indexes will be streamlined and archiving data will be easy.
- The primary difference between two techniques of data warehousing is the normalized data foundation. The data in the relational model exists in a form that is termed the "normalized" level. *Normalization of data* implies that the database design has caused the data to be broken down into a very low level of granularity. Another thing that separates these two approaches is the management of atomic data. With Inmon's architecture, atomic data will be stored within a normalized DW. In contrast, the Kimball method states that the atomic data should be placed within a dimensional structure.
- Widely considered models for database design is dimensional and relational model. The relational model is a central part of OLTP, which is the whole concept of managing databases in a relational manner. RDBMS begins with the organization of data into a table. Different columns are in each row of the table. The columns of data have different physical characteristics. Different columns can be indexed and can act as identifiers. Normalization and indexing are most important concepts of RDBMS. In *dimensional database model* data is presented in dimensional structure. According to Kimball the key difference between dimensional models is the *degree of normalization*. A dimensional model contains the same information as a

normalized model but packages the data in a format whose design goals are user understandability, query performance, and resilience to change.

- Inmon suggests that relational environment is shaped by the corporate or enterprise data model while dimensional model is shaped by the end-user requirements.
- Flexibility and performance issues are described as most important difference between relational and dimensional model in literature. It is considered that relational model is highly flexible but is not optimized for performance while dimensional model is good at in performance but not at flexibility. But it is considered in literature that the greater flexibility causes greater complexity.
- Inmon states that the granular data in the relational model is also used to service unknown future needs for information, not only to meet existing needs of existing user groups.
- On the other hand many authors consider that conducting the basic operations such as *complex data aggregation, comparison and consolidation* is difficult in relational method.
- Some authors suggests that dimensional model and DMs are not scalable; those are departmental and not enterprise solutions and only appropriate when there is a predictable usage pattern while the others refute these assumptions.
- Inmon (2005) argues that DM data structures are not reusable, are not flexible, are not useful as a foundation for reconciliation, and are not standing ready for a new set of unknown requirements. But the normalized granular data found in a DW are indeed all of those things.
- In *multidimensional database model* data are presented in multidimensional structure, opposing to tables in a relational database platform. *Multidimensional*

systems provide a multidimensional conceptual view of the data, including full support for hierarchies and multiple hierarchies.

- OLAP is commonly defined as *Fast Analysis of Shared Multidimensional Information* (*FASMI*), a critical category of information technology and a significant tool for high performance and multidimensional analysis of large scale business data.
- Functional requirements for OLAP are rich dimensional structuring with hierarchical referencing, efficient specification of dimensions and dimensional calculations, separation of structure and representation, flexibility, sufficient speed to support ad hoc analysis and multi-user support.
- The followings are considered as the differences between relational reporting and OLAP analysis in theory:
  - OLAP provides interaction with data while relational reporting can provide limited interactivity.
  - OLAP cubes are very good at providing very complex calculations whereas it is difficult in relational reporting.
  - Relational reporting tools are better at providing report formatting than OLAP tools.
  - OLAP provides very fast query performance when it is compared with relational model.
  - Relational reporting is better in handling textual information than OLAP.
  - OLAP may provide less maintenance since all aggregations are automatically provided within the cube. Relational reporting can be maintenance intensive if numerous "aggregation tables" are necessary.

  - Ad hoc query creation is very easy with OLAP tools when it is compared with relational reporting.

- Some of the authors suggest that a relational database is the necessary prerequisite for OLAP and it is desirable to have relational database and OLAP capabilities in one system, which could make both components more valuable to their users.
- OLAP *Cube* is a structure specially designed to retrieve, analyze and report data efficiently on the OLAP database platform. It consists of numeric facts called *measures*, which are categorized by *dimensions*. The cube *metadata* is typically created from a *star schema* or *snowflake schema* of tables in a relational database. *Measures* are derived from the records in the *fact tables* and dimensions are derived from the *dimension tables*.
- *Hierarchy* is one of the significant concepts related to OLAP Cube, which is an organization of members into a logical tree structure.
- *Sparsity* and *denseness* are also basic concepts of OLAP. *Sparse* is a situation that occurs when an aggregate table is created that is not appreciably smaller than the table on which it is based. A multidimensional database is *dense* if a relatively high percentage of the possible combinations of its dimension members contain data values.
- OLAP cube operations involve *slicing* and *dicing*, *drill-down analysis, rolling up and pivoting.*
- There are three major types of OLAP tools that differ with respect to their functionality and architecture: MOLAP, ROLAP, HOLAP. As shown through literature review, there is not an agreement between the authors on the most effective type of these OLAP architectures.

- OLAP cubes are considered as specifically developed tools for the queries in large databases with many relations involved, such as those in banks and optimized for short query response time. OLAP systems are utilized to discover trends and to analyze critical factors and to keep the banking managers informed about the business situation.
- Since OLAP cubes are precalculated, which results in fast query performance that is critical for executive use, OLAP is well suited to handle complicated organizational roll-ups, as well as complex calculations, including inter-row manipulations. So, OLAP naturally fits for banking reporting.
- The results of a research show that the banks, which use BI tools are larger and more successful. Large and successful banks invest more in information technology, especially BI in the purpose of more efficient business reporting.
- Additionally some of related academic studies suggest particular techniques to improve efficiency, query performance and/or reporting capabilities of OLAP tools. Those are:
    - Sparse datacube techniques to solve sparse matrix problem;
    - The variant index types (bit-sliced indexing and projection indexing) against algorithms using more conventional indexes;
    - A new cube structure promoting simple multidimensional range query in comparison with costly *star-join*, which is evaluated very efficiently to the native support for many dimensions;
    - A SplitCube approach for partial pre-aggregation to improve the query performance of OLAP cubes, reduce the cube size and shorten the cube building time.

CHAPTER 4

METHODOLOGY: RESEARCH DESIGN

Overview of Research Questions and Objectives

This study was conducted to determine the prospects of OLAP technology for managerial reporting in banking. Main goal of the study was to demonstrate the opportunities and limitations in use of OLAP Cubes for BI reporting in banking from technology perspective. This goal was essentially approached by comparing OLAP reporting to traditional techniques through several data collection and analysis methods.

As described in first chapter, the main research question that guided the study was:

What are the opportunities and limitations of using OLAP Cubes for BI reporting in banking?

A number of secondary research questions are also formulated for answering main research question and providing a framework for the study. These are:

RQ1. What constitute the components, layers and types of architecture of BI and DW infrastructure?

RQ2. What are the needs for BI and its utilization areas in banking?

RQ3. What are the basic differences between relational and dimensional models for database design in BI/DW process?

RQ4. What are the basic concepts, features, types and operations of OLAP and Cubes?

RQ5. How relational reporting is compared to the OLAP reporting in theory?

RQ6. In theory, what are defined as the benefits of using OLAP Cubes in banking?

RQ7. How can an OLAP Cube be utilized to improve BI reporting in banking?

RQ8. Which opportunities and limitations exist in utilization of OLAP Cube technology for BI reporting in banking?

RQ9. What are quantitative and qualitative differences between OLAP and relational reporting in banking?

RQ10. What is the common opinion among experts and users on using OLAP Cubes in banking?

RQ11. To what extent do the opportunities and limitations of OLAP utilization in banking differ from theory?

Since the main goal of the study is considerably broad, the objectives of the study are approached from several aspects and the data collected from various sources.

A literature review was performed to answer research questions 1-6; the answer for research question 3 was also sought through the analysis of the Bank's current database architecture.

Research questions 7-11 were answered through empirical research. An OLAP cube that processes and reports the relevant data of the Bank was designed and developed to obtain answers to research questions 7-9. Several queries were conducted and the performance of the cube was tested and measured through these queries in comparison with traditional/relational data reporting of the Bank to show opportunities and limitations of OLAP cube utilization and to observe the functionality of such kind of implementation.

In order to answer question 10, the opinions of BI experts on the findings of the experimental cube implementation were obtained through a questionnaire. This questionnaire also provided answer to research question 9 by describing common opinion on qualitative differences between OLAP and relational reporting in banking.

Research question 11 was answered by comparing the results of empirical study; the findings from performance tests and the questionnaire with theoretical assumptions examined in Chapter 2.

The methods and techniques used to answer all these questions are described in detail in following section of this chapter.

Type and Approach of the Study

The objectives to attain the main goal of the study on empirical base were as follows:

- Experimental design and development of a Cube on existing BI infrastructure of *the Bank.* It was aimed to show the quantitative opportunities and constraints of an OLAP Cube implementation by testing its performance and to observe OLAP queries in the context of functional properties. Current BI infrastructure, management reporting system and relevant data of the Bank were investigated, identified and analyzed prior to the experimental design.
- To examine qualitative aspects of OLAP Cube utilization in reporting and to investigate the opinions of both experts and users at the Bank in comparison with the findings of the experimental Cube implementation.

A comparative research experiment along with a questionnaire that would be conducted on the results of this experiment was chosen as the core of methodology. However, due to the nature of the research subject requiring larger analytical framework, the experimental research is integrated in a theoretical review.

Phases of the research were as follows:

- Review of theoretical background and literature to identify the nature of BI and OLAP in relation to banking operations,
- Identification and analysis of the current BI infrastructure, management reporting

system and relevant data of the Bank,

- Design and development of an OLAP cube for thesis experiment,
- Observing functional features of OLAP cube through this experimental structure
- Collecting empirical data through:
    - Organization of performance tests to test and measure the quantitative features such as speed/response time and technical efficiency of cube in comparison with traditional reporting;
    - Conducting a questionnaire among the BI experts for post-analysis, in order to investigate the common opinion on performance test results and qualitative aspects of OLAP cube utilization;
- Analysis of empirical data from experimental comparative model and the questionnaires and discussion of results;
- Producing conclusions and recommendations for future research.

On the basis of these research questions, objectives and phases, the type and approach of thesis can be described as follows.

Neuman (2007) classifies the *types of research by purpose* as:

- Exploratory research
- Descriptive research
- Explanatory research

Table 3 shows different goals of these three categories.

Table 3. The Goals of Different Research Types.

Exploratory research	Descriptive research	Explanatory research
Become familiar with the basic facts, people and concerns.	Provide a detailed, highly accurate picture.	Test a theory's predictions or principle.
Create a general mental picture of conditions.	Locate new data that contradict past data.	Determine which of several explanations is best.
Generate new ideas and conjectures, or hypothesis	Document a casual process or mechanism.	Support or refute an explanation or prediction.
Determine the feasibility of conducting research.	Report on the background or context of a situation.	Link issues or topics with a general principle.
Formulate and focus questions for future research.	Clarify a sequence of set of stages or steps.	Elaborate and enrich a theory's explanation.
Develop techniques for measuring and locating future data.	Create a set of categories of classify types.	Extend a theory or principle to new issues or topics.

Note: The goals of different research types. Adapted from *Basics of Social Research: Qualitative and Quantitative Approaches* (Second ed.). p. 16, by W. L. Neuman, 2007, London: Pearson Education-Allyn and Bacon.

According to mentioned classification, both *descriptive* and *explanatory* research methods were chosen considering the objective to obtain first hand data from the performance tests and investigate the opinions on the findings of these tests. This study is *descriptive* since it reviews the basic facts of BI, DW and OLAP technologies along with theoretical discussions in this field; attempts to describe the steps of BI and OLAP processes and tries to provide a clear view and detailed picture on usage of such kind of applications through investigation of common opinion of users and experts. In the meantime, it is an *explanatory* research due to the fact that it tests the relational and OLAP reporting processes and provides evidences through a comparative experimental analysis whether they support or refute the several arguments in theory; enriches the explanations on BI and OLAP applications in literature. Besides, these methods were appropriate for this study due to the fact that they can use several kinds of data through several data collection instruments as needed in this study.

There are different choices as well to classify (e. g. Neuman, 2007; Trochim, 2006, Greener, 2008) *research approaches* such as:

- Inductive versus deductive
- Qualitative versus quantitative

In *deductive* studies, existing theories and/or concepts are tested or confirmed. (Trochim, 2006). The researcher begins with logical relationship among concepts and move toward concrete empirical evidence. (Neuman, 2007) In *inductive* approach, the researcher begins with specific observations and measures; tries to detect patterns and regularities and develops some general conclusions or theories (Trochim, 2006).

As a matter of fact, the main approach of this thesis is *deductive* but it involves *inductive* approach at some points of the project. Because, the study mainly intends to test theoretical assumptions and provide empirical evidences on opportunities and limitations of using OLAP cube in banking, however, new patterns; new impacts of OLAP cube implementation in managerial reporting could be detected or observed as consequences of experimental analysis.

Another common division of research approaches is *qualitative* and *quantitative.* Quantitative research is empirical investigation of quantitative properties (Trochim, 2006). Qualitative researchers emphasize conducting detailed examinations of cases whereas almost all quantitative researchers emphasize precisely measuring variables and testing hypotheses (Neuman, 2007). Table 4 shows the differences between quantitative and qualitative researches.

Table 4. Differences Between Quantitative and Qualitative Research.

Quantitative Research	Qualitative Research
Test hypothesis that the researcher begins with.	Capture and discover meaning once the researcher becomes immersed in the data.
Concepts are in the form of distinct variables.	Concepts are in the form of themes, motifs, generalizations, and taxonomies.
Measures are systematically created before data collection and are standardized.	Measures are created in an ad hoc manner and are often specific to the individual setting or researcher.
Data are in the form of numbers from precise measurement.	Data are in the form of words and images from documents, observations, and transcripts.
Theory is largely causal and is deductive.	Theory can be causal or noncausal and is often inductive.
Procedures are standard, and replication is assumed.	Research procedures are particular, and replication is very rare.
Analysis proceeds by using statistics, tables, or charts and discussing how what they show relates to hypotheses.	Analysis proceeds by extracting themes or generalizations from evidence and organizing data to present a coherent, consistent picture.

Note: Differences between quantitative and qualitative research. Adapted from *Basics of Social Research: Qualitative and Quantitative Approaches*. (Second ed.) p. 88, by W. L. Neuman, 2007, London: Pearson Education-Allyn and Bacon.

Table 5 also demonstrates the main aspects of quantitative and qualitative methods (Brayman & Bell, 2003):

Table 5. Main Aspects of Quantitative and Qualitative Research.

Quantitative	Qualitative
Numbers	Words
Point of view of researcher	Points of view of participants
Researcher distant	Researcher close
Theory testing	Theory emergent
Static	Process
Structured	Unstructured
Generalization	Contextual understanding
Hard reliable data	Rich deep data
Macro	Micro
Behaviour	Meaning
Artificial settings	Natural settings

Note: Main aspect of quantitative and qualitative research. Adapted from *Business Research Methods*. p. 302, by A. Brayman & E. Bell, 2003, Oxford: Oxford University Press.

This thesis also cannot be specified as an entirely quantitative or qualitative study since both approaches were combined during data collection and analysis phases to answer research questions. Based on the objectives and research questions, selection of both of these approaches was found more appropriate to overcome their limitations. It is *quantitative* because the techniques that produce quantitative data (data in the form of numbers) used in both phases. Theoretical descriptions and assumptions on OLAP implementation for reporting large scale data are tested and represented empirically through performance tests and a questionnaire involving structured questions. It is qualitative as well since a broad review of the theory and literature and the observations generating verbal information conducted to answer certain research questions.

Very few studies have particularly examined the impacts of BI implementation in banking. Even, the number of the studies examining OLAP implementations from technical perspective on such large-scale data in any business area is limited as shown in Chapter 3. None of them has investigated the role of OLAP cube utilization in managerial reporting for banking in technical manner. Additionally, there are still limited number of studies in the academic literature that analyze BI and OLAP; most of the studies in this area contribute commercial interests rather than academic purposes. Therefore, this thesis has to adopt such multipurpose method and approach in order to better address the research questions, contribute future academic studies besides business productivity through an empirical research.

To conclude, type of the study and approach to research issue involves several directions and methods. Such kind of methodology is referred as "triangulation". Triangulation is the "rationale for using multiple sources of evidence" (Yin, 2003, p. 97) "where different methods of data collection and analysis will both enrich and confirm the picture (…) of the situation" (Greener, 2008, p. 36). It can be conducted on the basis of data sources, methods, theories and investigators (Yin, 2003). Triangulation is considered as an approach that increases the validity of the research and provides a check of findings from a particular method (Greener, 2008)

Data Collection and Measurement

Yin (2003) classifies research strategies applied to collect empirical evidence under five categories: Experiments, surveys, archival analysis, histories and case studies (see Table 6). According to Yin (2003, p. 5) the three conditions distinguishing the strategies are "(a) the type of research question posed, (b) the extent of control an investigator has over actual behavioural events, and (c) the degree of focus on contemporary as opposed to historical events". Categorization scheme for the type of research questions are "who", "what", "where", "how" and "why" (Yin, 2003).

Table 6. Relevant Situations for Different Research Strategies.

Strategy	Form of Research Question	Requires Control of Behavioral Events?	Focuses on Contemporary Events?
Experiment	how, why?	Yes	Yes
Survey	who, what, where?	No	Yes
Archival analysis	who, what, where?	No	Yes/No
History	how, why?	No	No
Case study	how, why?	No	Yes

Note: Adapted from R. K. Yin (2003) Case Study Research: Design and Methods. Third edition. (p. 5). Sage Publications. Thousand Oaks:CA.

According to Neuman (2007), data collection techniques fall into two categories based on whether the data being gathered are quantitative or qualitative. He describes these techniques as follows (Neuman, 2007, p. 20-21):

Quantitative Data Collection Techniques:

- *Experiment*, in which the researchers create situations and examine their effects on participants,
- *Survey,* in which the *researcher* asks people questions in a written questionnaire or during an interview in a short time period; acquires a picture of what people think and generalizes the results from samples or small groups to a larger group,
- *Content analysis*, which is a technique for examining information, or content, in written or symbolic material,
- *Existing statistics* research, in which the researcher locates previously collected information or previously conducted surveys, then reorganizes or combines the information in new ways to address a research question.

Qualitative Data Collection Techniques:

- *Field research*, in which the researcher conducts case studies observing a small group of people in detail over a length of time,

- *Historical-comparative research*, in which the researchers examine aspects of social life in a past historical era or across different cultures; compare one or more cultures or historical periods.

Neuman (2007) also classifies data collection methods by *time dimension of research* as follows:

- *Cross-sectional research*, which examines a single point in time or take a one-time snapshot approach,
- *Longitudinal research*, which examines features of people or other units at more than one time,
- *Case study*, which examines, in depth, many features of a few cases over a duration of time with very detailed, varied and extensive data, often in a qualitative form.

A classification for scientific research (Capri & Egger, 2008) describes the data gathering methods under following categories:

- *Experimentation*: Experimental methods are used to investigate the relationship between two or more variables (a condition or a parameter) when at least one of those variables can be intentionally controlled or manipulated. The resulting effect of that manipulation can then be measured on another variable or variables.
- *Description*: Description is used to gather data regarding natural phenomena and natural relationships and includes observations and measurements of behaviors.
- *Comparison*: Comparison is used to determine and quantify relationships between two or more variables by observing different groups that either by choice or circumstance are exposed to different treatments.
- *Modeling*: Both physical and computer-based models are built to mimic natural

systems and then used to conduct experiments or make observations.

Considering quite broad objectives and the nature of the research questions of the thesis, it would not be suitable to apply a single method to collect empirical data. Both quantitative and qualitative techniques were used and data were gathered from several resource. In accordance with available research environment and material, following data collection methods were chosen to formulate rational conclusions and recommendations for the study:

1. *Conceptual analysis* through the review of theoretical background and literature: A conceptual framework was established to define, describe, classify and organize existing information related to subject of thesis before quantification and measurements. Chapter 2 and 3 of the thesis are based on conceptual analysis, which examine the components and architecture of BI; utilization of BI in banking; differences between relational and multidimensional DB design in BI/DW processes; OLAP concepts, features and operations; differences and similarities between relational and OLAP reporting; the benefits of using OLAP cubes in banking and related academic studies. This involved the qualitative part of the study.
2. *A comparative experiment* through design and development of an OLAP Cube. This method was used for experimental verification and quantification of theoretical predictions. Current BI infrastructure, management reporting system and relevant data of the Bank were investigated and identified prior to this experimental design. These data were gathered through meetings and discussions with the experts at the Bank headquarters and by conducting on-site examinations of existing BI infrastructure.

Several performance tests by a series of queries and technical observations were conducted during experimental phase of the study. It was aimed to measure, examine and compare certain aspects of relational and OLAP reporting on existing data structure of the Bank, to obtain objective evidence about the opportunities and limitations of using OLAP cubes in banking reporting. Main steps of this phase of the study were:

- Identification of relevant data of the Bank being used for management reporting
- Preparation of the cube development environment
- Design and development of the Cube's source database
- Preparation of the mapping documents
- Design and development of ETL processes
- Loading data into the Cube DM
- Design and development of the Cube
- Preparation of the static reports
- Conducting performance tests and comparisons

Briefly, the bases of comparisons for queries through which quantitative properties of two different reporting environments were measured by duration and/or response time in several scenarios were:

- Report processing
- Source system data update
- Cube processing
- Response to new requirements on source system
- Response to change requests

All steps of this phase of the study will be explained in detail in Chapter 6.

3. *Survey* through *a questionnaire* based on *cross-sectional technique*.

It was aimed to provide the possibility of control on the results of performance tests that have been previously conducted. In short, questionnaire was chosen as a method to verify the findings of the experiment and to collect additional data to assess qualitative aspects of the Cube.

Trochim (2006) groups surveys into two categories: *Questionnaires* and *interviews.* According to Neuman's (2007) classification, the types of survey are: *Mail and self-administered questionnaires, web surveys, telephone interviews,* and *face-to face interviews.*

Questionnaire is a collection of written queries, which is arranged putting all the essential variables for the research and can be completed by respondents in presences, in absentia, directly or indirectly (Cooper & Schindler, 2003).

Some of the authors divide the interviews into three categories: Structured, semi-structured and unstructured (Greener, 2008). In unstructured or in-depth interviews, a questionnaire or interview guide is not used. Semi-structured interviews are based on a question guide, however, interviewee is allowed to go where they want with the questions. Both of these methods are qualitative whereas structured interviews in which clear questions are asked in a consistent way are quantitative methods similar to administration of self-administered questionnaires (Neuman, 2007; Greener, 2008).

In this study, a structured questionnaire has been chosen as a survey method based on *cross-sectional technique*, which examined a single point in time; took a one-time snapshot of opinions of participants. Selected participants answered a questionnaire structure in *Likert* format consisting of *close-ended* questions with multiple choice options in accordance with the purpose of the survey toward confirming previously obtained information (Greener, 2008). The participants were also asked to submit additional comments at the end of each group of questions, which might clarify their responses.

The population (literacy, cooperation), sampling (location, frame, response rates), questions (type, complexity), content (need for knowledge of participants) and administrative (costs, facilities, time) issues were considered for selection of survey method in this study (Trochim, 2006).

Specifically, a total of eleven respondents among *readily available and convenient* experts (who develop and/or construct BI applications) of the Bank's IT department were selected to make up the sample. Thus, the chosen method was *convenience sampling* since it wasn't aimed to produce highly generalizable results *directly* from the questionnaire that highly represent the population. The major goal of conducting this questionnaire was to evaluate the findings of the experimental results of the study; to verify or refute the opportunities and limitations of OLAP Cube utilization for BI reporting of the Bank that were determined through the performance tests and observations. However, experience and sufficient time factors have been considered in making the list of participating experts. Response rate was a hundred percent, because the number of respondents was relatively low and all questionnaires were administered in presence of the interviewer. *Interviewer-administered questionnaire method* was chosen due to very limited number of the respondents, in order to decrease the number of "undecided" responses and to have highest response rate (see Neuman, 2007, p. 190). Besides, a detailed background of the study could be provided by interviewer while conducting questionnaires.

The questionnaire was containing thirty questions derived from the results of performance tests, the observations during multiple queries and certain theoretical assumptions on functional properties of two reporting system. The questions were in the format of five-level Likert rating scale.

Scales are common in situations where a researcher wants to measure how an individual feels or thinks about something. The objects are text statements, usually statements of attitude or belief in scaling. Scaling produces quantitative measures and can be used with other variables to test hypotheses. (Trochim, 2006; Neuman, 2007). Therefore, it was most appropriate method for the aim of study, for verifying experimental results. Most common scaling methods are (Trochim, 2006):

1. Thurstone (or Equal-Appearing Interval) Scaling,
2. Gutmann (or Cumulative) Scaling,
3. Likert (or Summative) Scaling.

Likert scale, which allows the respondents to mark a numerical scale in response to a question (Greener, 2008) has been chosen in this study since it provides the researcher with ease of construct; enables the respondents to answer the questionnaire easily and gives the opportunity to use of statistics efficiently for data interpretation.

Each respondent was asked to rate each question or statement on the response scale with five choices. The scale was ranging from *strongly disagree* (-2) to *strongly agree* (+2). The choices represented the extent of agreement of respondent on the issue. Analysis method for the results of these measurements will be explained in the next section of this Chapter.

Data Analysis

In this study three different procedures were needed to analyze data gathered through three different instruments:

1. Analysis of relevant data of the Bank,
2. Analysis of the results from performance tests and comparisons,
3. Analysis of the results from questionnaires.

With the conducted interviews, the BI infrastructure of the Bank was understood and a high level model was drawn (see Figure 15). Then the management reporting system of the Bank was analyzed and the information currently reported within the system was extracted from the scheduled static reports. The reported fields were listed and then grouped under the sets of dimensions and measures in order to provide a reference model for the Cube design.

The results obtained from the performance tests for OLAP Cube were analyzed by comparing them with the performance of relational reporting system of the Bank, which redeveloped on the DM. As mentioned above, during these tests, run durations evaluated on a set of eight reports that differ in report complexity, number of joined tables, row count of the queried source tables, row count of the result set and the number of queried terms. Each query was run both on the OLAP Cube and the RDBMS for five times to calculate average durations or in other words, to produce *mean values* that indicate mid-points in each measurement. Comparing *means* was regarded as a robust technique, because there was no outlier obtained, which deviated markedly from the other results and might cause imprecision (see Neuman 2007; Greener, 2008). These measurements were compared and presented in tables created from spreadsheets.

Besides, the "additional" cost of owning and maintaining such a Cube for the MI reporting was calculated by generating some scenarios; response to new development needs, daily and monthly DM update durations, total effort needed to process the Cube, necessary disc size for adding a new term to the DM and the Cube were measured. There were a number of sub-tasks, which would ease observing the total time, necessary for each modification scenario. The tests that measure duration in this phase were mainly related to necessary human effort required to accomplish the tasks and the calculations were mostly made on the basis of Man/Hour (MH). The duration of each task was calculated by the average of five trials and *mean values* were also presented in comparison tables. The learning effect was not considered as a problem since the implementer has enough experience for accomplishing all the necessary tasks. The results from the trials also verify this assertion by very close durations. Another problem about the generalizability of results from the trials might emerge since these trials were conducted by only one person. In order to overcome this problem, a series of questions asked to participants within the questionnaire to verify the results from this experiment.

The questionnaires were in Likert format through which respondents' attitudes have been measured by asking the extent to which they agree or disagree with the particular questions or statements (Strongly disagree= -2, disagree= -1, undecided = 0, agree= +1, strongly agree= +2). *Mean* was used also for the interpretation of these data, to compute the answers to each question as a technique that analyzes *central tendency* in a data set (see Neuman, 2007; Trochim, 2006). This technique was chosen among several options for analysis of Likert scaling such as mode, frequency distribution and chi-square. Because, it was aimed to include every value in the data set as part of the calculation.

Reliability and Validity

The term *reliability* refers the "consistency", "repeatability" or “dependability” of the measures used in the research. It means that the numerical results produced by an indicator do not vary because of characteristics of the measurement process or measurement instrument itself (Neuman, 2007). A measure is considered reliable if it would give us the same result over and over again (Trochim, 2006) or at least the method is clear enough to instill confidence in the reader that the results were not fudged in any way (Greener, 2008).

In theory, several methods to increase the reliability of the research are recommended such as clear conceptualization of constructs, using a precise level of measurement, using multiple sources of evidence; using pilot-tests and triangulation in which different data collection, measurement and analysis methods are utilized (Neuman, 2007; Yin, 2003; Greener, 2008).

Even though it is rare to have perfect reliability, in this study, most of these methods were taken into consideration to support sufficient reliability. The approach and methods used to collect, measure and analyze data involve several directions and techniques. It was tried to obtain evidences from several sources and to conduct measurements at the most precise level possible and to verify the results from a particular method using another technique.

The reliability of comparison of relational and OLAP Cube reporting and their performances should be sufficient due to the several reasons. The same data sets were used as the basis for reporting. Additionally, to prevent potential performance advantage of the source of the existing powerful reporting system, selected reports were redeveloped in the same formats to run on the new relational DM. Besides, the reports prepared on both systems produced the same results, excluding insignificant differences in presentation of data due to the different architectures of the relational and multidimensional environments. However, the information interpreted from the data was exactly the same. The tests to measure the performance of two different reporting systems were repeated several times and the average values calculated for adequate precision. Finally, the results obtained from these tests were verified and reevaluated through structured interviews with users and experts.

Validity is another quality criterion for research and concerned whether the research actually examines the intended issues or not. In theory, there are several ways of characterizing and measuring validity of a research (Greener, 2008; Neuman, 2007; Trochim, 2006; Yin, 2003): *Face validity* is a judgment by the scientific community that the indicator really measures the construct, even a non-researcher can broadly see that this is a valid method of researching this question. *Construct validity* refers establishing correct operational measures. *Internal validity* is the approximate truth about inferences regarding cause-effect or causal relationships and only relevant in studies that try to establish a causal relationship. *External validity (generalizability)* is the ability to generalize findings from a specific setting and to a broad range of settings. High external validity means that the results can be generalized to many situations.

The following approach was adopted in order the increase the validity of this study: First of all, the subject of the study was examined in both theory and practice; theoretical framework of BI, OLAP and related concepts reviewed systematically along with their applications in practice based on empirical findings to gain an in-depth understanding. Secondly the methods of collecting, measuring and analyzing data were carefully investigated and discussed and selected. Utilization of multiple data sources and methods was chosen also as a way of improving validity of the study. Third, the research questions, the features of reporting systems that would be tested and compared and survey questions were determined considering theoretical aspects of BI and OLAP reporting and by asking the opinions of the experts in this field. The questionnaires were applied after the experts confirmed that the questions were precise enough. The results from this study can reasonably be expected to apply to future researches on such large scale data, on the other hand it is unclear whether they could apply in small data sets since it was not aimed to determine the opportunities and limitations of using OLAP cubes for small data sets. However, the considerably broad scope and the extent of the study will promote the generalizability.

CHAPTER 5

DESIGN AND DEVELOPMENT OF THE CUBE

Review of Current BI Infrastructure and Management Information Reporting System of the Bank

The Bank is one of the biggest banks in Turkey with its more than twenty million customers and over twenty thousand workers. Being serving this much of people inevitably leads the organization to have numerous operational systems and terabytes of data stored in several sources.

There is a huge mainframe, which is built up on a hierarchical database management system and handles most of the operations. Operational open system applications also exist within the IT infrastructure of the Bank, which are based on relational database architecture. In order to manage these data from separate sources and establish relations between them, the Bank has developed a BI system in 1999.

The current status of the BI infrastructure of the Bank is demonstrated in Figure 15:

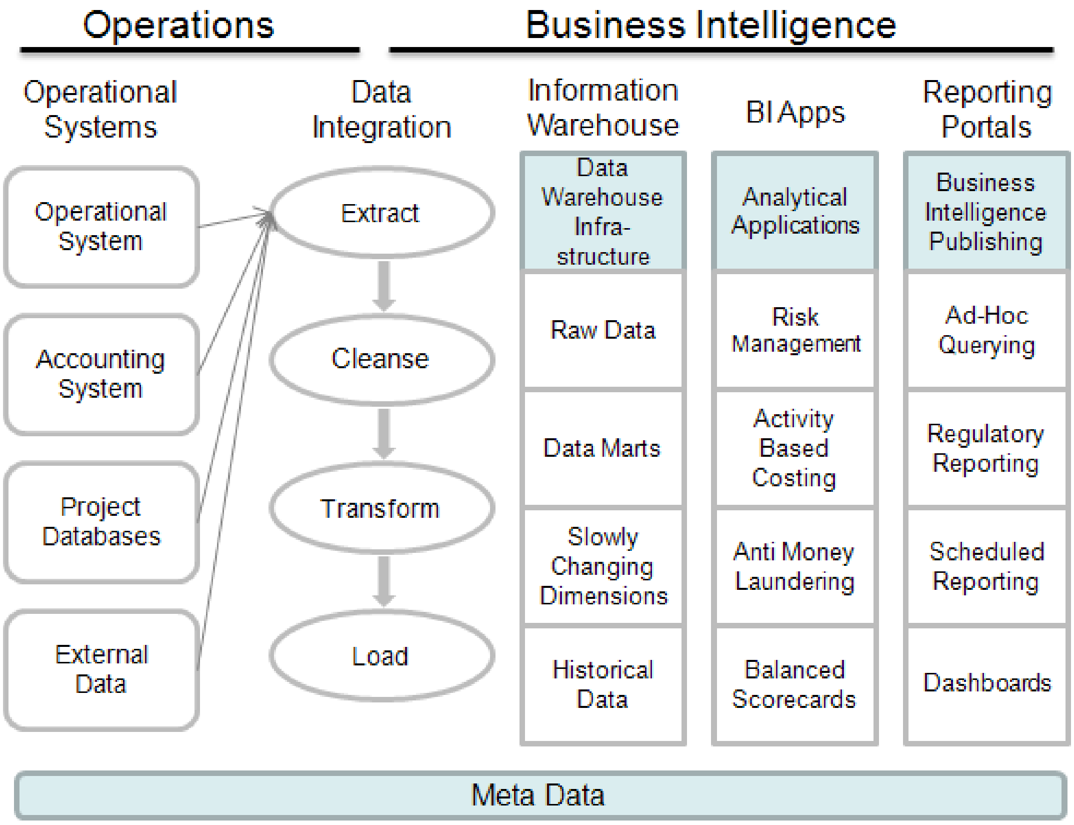


Figure 15. BI infrastructure of the Bank. Adapted from "Business Intelligence Infrastructure" by M. Robinson, 2008, *Information Management Online*. Retrieved February 16, 2011 from http://www.information-management.com/specialreports/20020521/5211-1.html

At the data source level, besides the operational systems running on mainframes and open systems, there are also external data coming from the foreign branches, the third party information providers and national organizations. These data exist in many different formats like hierarchical, relational and text based.

At the data integration level, there is an ETL server, which connects to these heterogeneous sources and extracts, cleanses and transforms the data in order to load them into the corporate DW. After the data is loaded into the DW, some further transformations take place; some new fields are added; data are prepared for being queried and loaded into the DMs. The termly data are stored in the form of monthly snapshots, which are cumulated for fifteen years. Some technical properties of the Bank's DW are as follows:

- The relational database management system is DB2 9.5.
- The system runs on IBM P7 series servers and utilizes AIX operating system.
- The total size of this DW is over 30 TB growing at a rate of twenty percent per year.
- 400 GB of data is loaded into more than one thousand five hundred tables everyday (raw data, DMs, BI applications)
- More than a hundred thousand lines of code running (SQL & UNIX Shell scripts)

The next level of the Bank's BI architecture is the *Application Layer*. There are more than twenty BI applications running over the DW along with the other external applications using the DW as source. Some of these applications are:

- Balanced Scorecards
- Activity Based Costing
- Customer Relationship Management
- Anti Money Laundering
- BASEL II
- Fraud Detection

Those which run on the DW are developed by combining UNIX Shell and DB2 SQL scripts and the external ones developed by using several programming languages and third party software.

At the final layer, the data are presented to the business users. Several methods are utilized to accomplish this task, which can be summarized as:

- Business Objects (BO) Universes: For ad-hoc purposes

- Scheduled Reporting: For delivering static reports to specified users/branches in a specified time (Business Objects / Raw SQL scripts)
- Regulatory Reporting: To provide regulatory information to the governmental agencies (Business Objects and UNIX&SQL scripts)
- OLAP Reporting: To obtain reports from the Balanced Scorecard and Branch Targets applications
- Dashboards: To visually demonstrate a very aggregated and summarized information to the top level managers (SAP Xcelcius)
- Custom Applications: Some information screens are developed using open system programming languages to present the data in a structured way to the operational level business workers.

The current MI reporting system of the Bank is based on scheduled reports, which involve some critical measures of the corporation in a highly aggregated form. The level of aggregation starts from the most granule dimension of the Bank, the *account*; and goes to some important grouping sets such as *branches*, *regions* and *segments*. There are more than forty reports running daily and monthly, which contain different grouping options and different important KPIs. All of these reports were prepared using SAP Business Objects 6.5 and scheduled by an application developed on .NET platform. The results from these reports are published to a server and the business users can access them through a web portal.

Identification of Relevant Data
of the Bank Being Used for Management Reporting

In this study, the information currently reported within the Management Reporting System of the Bank were summarized by analyzing daily and monthly static reports and the fields were grouped under the sets of dimensions and measures in order to provide a more clear reference model for the Cube design. These dimensions and measures are:

- *Segment dimension*; keeps the information on the definitions of the segments that are derived from the CRM system of the Bank.
- *Product dimension*; keeps the information about the definitions of the valid products, product groups and product categories.
- *Branch dimension*; keeps the information of the Bank's more than 1200 branches with the regions that they locate in.
- *Region dimension*; keeps the region definitions.
- *Balance measure*; keeps information on balances of all the deposit and loan products of the customers. These data are collected from thirteen different product tables whose sources are the aggregation of the daily transactions that take place within the mainframe of the Bank.
- *Channel usage measure*; keeps information about activity costs, capacities and transaction counts of the communication channels on the basis of customers' products. These data were obtained from the Activity Based Costing system of the Bank.
- *Profit measure*; keeps information about account based incomes, expenses, average balances and profits. These data were collected as well from the Activity Based Costing system.

Preparation of the Cube Development Environment

It was not appropriate to develop such kind of OLAP Cube on the DW environment due to the following reasons:

- Processing the Cube would use a great amount of system resources during querying the necessary data within the source database. This would slow down the other jobs on the server.
- The DW was utilizing DB2 9.5 and there was no practical way to keep the contemporary and historical data together within partitioned tables due to inefficient partitioning capability of the RDBMS software.
- Performance observations could not be performed precisely since the load on the corporate DW was uncertain.

Therefore, it was decided that the Cube's source database should be on another and more convenient environment rather than the DW. Better RDBMS software and a separate server were needed to start the development.

Oracle was selected as the RDBMS; two servers for test and production levels have been arranged and necessary software were installed in order to develop the Cube. The configurations of the new DM and Cube environments are given in the following table:

Table 7. Configurations of Data Mart and Cube Environments of the Bank

Server	Processor Type	CPU #	Memory	Operating System	RDBMS
Cube Server (Prod Env.)	IBM P-Series 5	4	16 GB	Windows Server 2003 SP2 64bit	-
Cube Server (Dev Env.)	Intel Xeon	2	4 GB	Windows Server 2003 SP2 64bit	-
Source Database Server (Prod Env.)	IBM P-Series 5	16	24 GB	Operating System IBM AIX 5.3	Oracle 10.2.0.4.

Design and Development of the Cube's Source Database

The DW model was developed using CA ERwin Data Modeller, which is a graphical tool for logical database design. This tool was used to create a conceptual model of the DW. Since the performance was a major concern for the management reporting processes, the tables have already been denormalized within the DW and modelled in the form of a star schema. So, the model of the new DM had a similar design with the previous one. In order to make termly groupings, a *Date* dimension table was added to the model, which would keep several date fields. Figure 16 illustrates the final DM model. A copy of the image with better resolution can also be found in Appendix C. The Balance table was demonstrated as a single table, but it was formed by combining thirteen different balance tables, which are demonstrated in Appendix D.

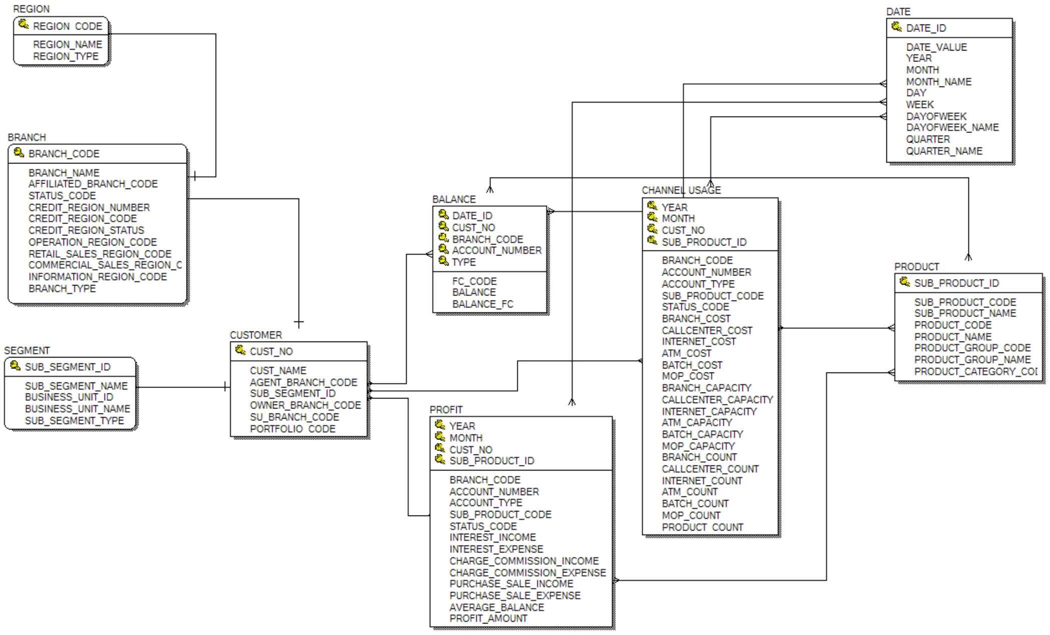


Figure 16. Final data mart model.

When the design phase was completed, the modelling tool transformed the database design into SQL DDL statements, which were necessary to build the physical database objects. By using these statements, the tables and necessary indexes were created on both development and production levels. Since the amount of data to be stored in the tables was too large, most of the tables were created with monthly partitions in order to increase performance and make the maintenance easier.

Design and Development of ETL Processes

Next stage was the preparation of the ETL jobs, which would feed the Cube DM.

The Bank uses IBM Data Stage 8, a powerful ETL tool for data integration that can connect to almost all data sources of the corporation except the mainframe. All ETL operations are carried out on an IBM P7 server.

First step of developing ETL jobs is to identify which source column would be loaded into which target column with what necessary transformation. In order to do that, the data mapping documents were prepared which describe the relation between the DW and the Cube DM. These mapping documents can be found in Appendix E.

When the mapping documents became ready, ETL jobs were prepared accordingly. A separate job was designed parametrically for each table in order to fill each specified partition of the tables. Finally, they were brought together within a job sequence.

After the jobs were tested in the development environment, they were deployed to the production level. The jobs were run multiple times and the DM was filled with the termly data beginning from December 2006 to the present day. The total amount of data loaded into each table is shown in Table 8.

Table 8. Amount of Data Loaded into Tables.

	Current Partition		Termly Partitions	
	Row Count	Total Size (MB)	Row Count	Total Size (MB)
FACT_PROFIT (Channel + Profit)	51,146,990	2300.00	2,474,129,776	177,000.00
FACT_BALANCE_DEMAND_DEPOSIT_TL	17,052,984	577.00	756,592,220	24,000.00
FACT_BALANCE_CREDIT_CARD	8,115,973	525.00	332,323,031	18000.00
FACT_BALANCE_INVESTMENT_ACCOUNT	4,701,166	157.00	208,275,080	6,400.00
FACT_BALANCE_DEMAND_DEPOSIT_FC	3,717,107	118.00	184,703,049	5,500.00
FACT_BALANCE_OVERDRAFT	2,244,903	79.00	101,818,204	4,600.00
FACT_BALANCE_CONSUMER_LOAN	1,303,644	86.00	59,769,337	2,100.00
FACT_BALANCE_TIME_DEPOSIT_TL	983,176	41.00	45,173,996	1,500.00
FACT_BALANCE_TIME_DEPOSIT_FC	320,682	15.00	17,430,548	618.00
FACT_BALANCE_LETTEROF_GUARANTEE_YP	182,131	5.20	7,552,414	214.00
FACT_BALANCE_LETTEROF_GUARANTEE_TL	462,597	11.00	5,218,766	184.00
FACT_BALANCE_DEBIT_ACCOUNT	55,558	2.80	2,358,339	104.00
FACT_BALANCE_FC_LOAN	11,127	1.00	508,505	20.00
FACT_BALANCE_FOREIGN_BRANCHES	6,258	0.50	118.15	20.20
Fact Total	90,304,296	3,919	4,196,071,415	240,260
DIM_CUSTOMER	24,491,350	1,372.00	86,165,449	4,800.00
DIM_BRANCH	1,835	10.00	-	-
DIM_DATE	4,657	0.40	-	-
DIM_SEGMENT	29	0.30	-	-
DIM_PRODUCT	71	0.30	-	-
DIM_REGION	98	0.10	-	-
Dimension Total	24,498,040	1,383	86,165,449	4,800
Grand Total	114,802,336	5,302	4,282,236,864	245,060

About 250 GB of data were transferred into the DM, which filled fifty-one partitions at the initial load phase. For the continuity of the data transfer, these jobs were had to run periodically, so they were scheduled with an application called Tivoli Work Scheduler (TWS). It is a tool from the family of IBM workload automation products that plan, execute and track jobs. The Bank uses this tool for most of the open system scheduling purposes. For daily and monthly runs, the job was scheduled separately

Design and Development of the Cube

The data cube was built using MS SQL Server 2005: Analysis Services (SSAS). The first step of building a SSAS Cube was to define a data source, which in existing case was the new Oracle DM. After the connection with the relational environment was established, a data source view (DSV) had to be developed. That was a logical data model containing the metadata of the database objects, which would be used in the actual cube design. It defines the relationship between these objects. It’s also possible to define new objects by using named queries to simplify the data source structure. The actual purpose of defining a DSV is to isolate the development of the Cube from the relational data source so that the development process can still be carried out without an active connection to the database. In this study, most of the tables were imported into the DSV in the same format as they were modelled in the DM since they were already been structured and optimized for Cube development purpose. A modification was conducted during the definition of the Balance Fact Table within a single view, which was specified with a named query combining all of the thirteen products’ balance tables.

Another modification followed this: There has been a Customer table in the existing database model that wasn’t reported within the existing system but was a candidate for being a Cube dimension. However, it was not identified as one and used only for combining fact tables with the Segment and Branch dimensions due to the following reasons:

- Adding a dimension with more than twenty-four million rows would strain the process of the Cube as the MOLAP functionality attempts to pre-aggregate all the data. Theoretically, the number of calculations can simply be calculated as “Number of Combinations of all Rows in Dimensions x Number of Rows in Measures". At the current situation, it was about 10^9 x 10^9 aggregations. If customer dimension has been added, this number would become 10^{15} x 10^9.

- The necessary disk space would be tremendous while trying to store all the aggregated data within the data files.
- It was impossible to report the data with that much of rows through any convenient reporting utility, even with MS Excel 2010, which can handle up to hundred million rows.

The next step was to build the actual cube model. The Cube was built using the model shown in Figure 17 (see Appendix F for more detailed view). It is a screenshot from the actual SSAS project.

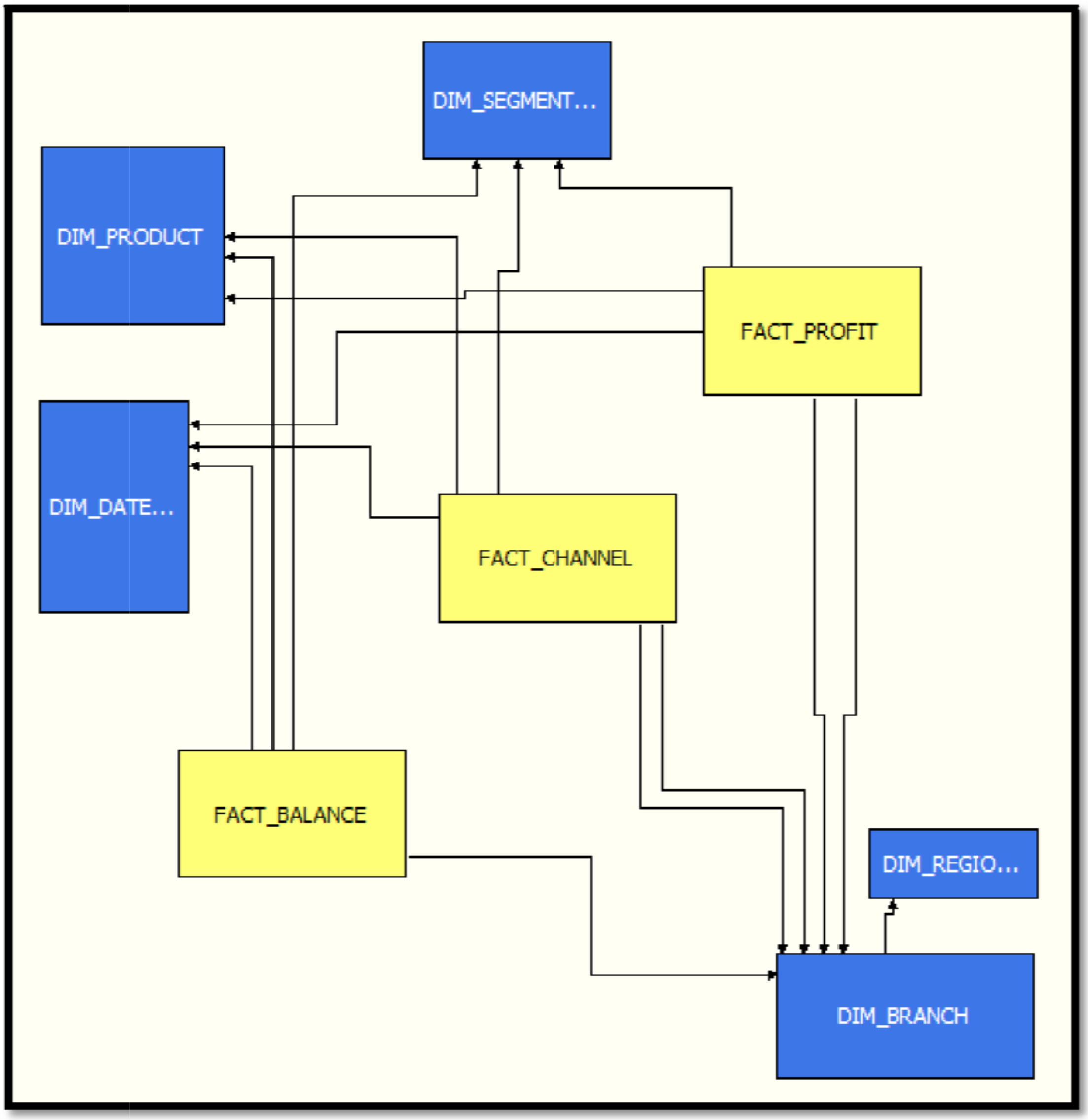


Figure 17. A high-level design of the Cube.

After the model was generated, the dimensions and their hierarchies were identified, the dimension usages, which define the relationship with the measure groups, were specified and the measure partitions were created. A separate partition for each month in the measure groups had to be created within the Cube. That is because there were a great amount of data in the fact tables and maintenance process of the termly data would be painful during each update in the source. The Cube was developed with MOLAP functionality while the query response times were the most prior concern of the study.

Data would be stored in multidimensional files and any change in the structure of the source DB or the data contained in the DB would be reflected to the reports only if the related partition of the Cube was reprocessed. The final step was to create the user roles to organize the privileges of the users. At that moment, only one fully authorized user was defined since the Cube would not be used by anyone during the study.

Following the completion of the design phase, the Cube was deployed to the production server, and all the dimensions and measure partitions were fully processed. In order to keep the cube up-to-date, a job was developed on the SQL Server Management Studio and scheduled to run daily to process the dimensions and current partitions of the measures. For monthly process, another job was created as well; however, it wasn't scheduled due to the manual interference requirements. This job had to be run manually at the end of each month due to reasons, which will be explained in Chapter 6.

The final view of the BI architecture of the Bank after development of the Cube was demonstrated by the Figure 18.

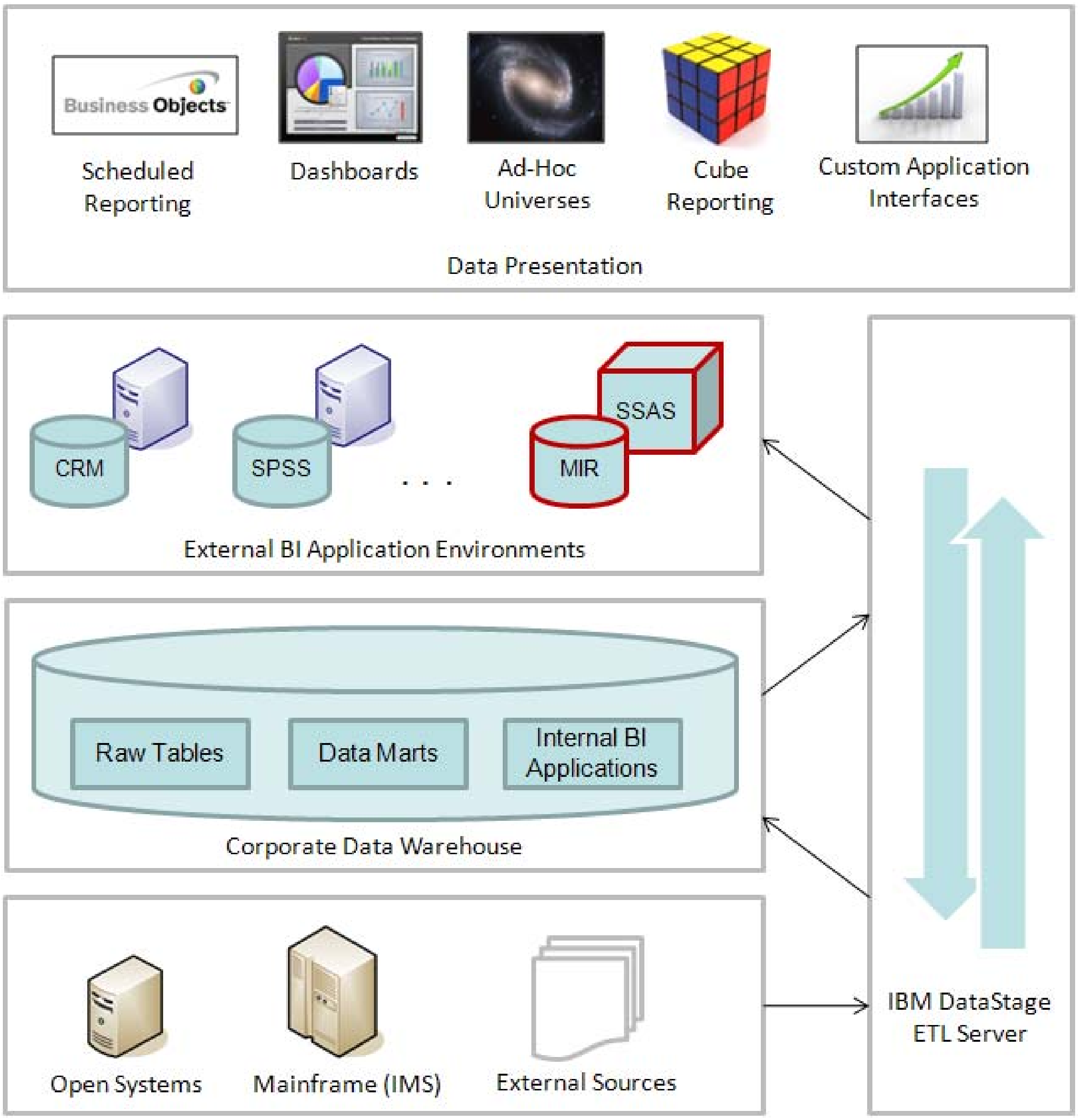


Figure 18. The final view of the BI architecture of the Bank after development of the Cube.

CHAPTER 6

PERFORMANCE TESTS

This chapter introduces the methods used for the performance comparisons of the relational and multidimensional reporting environments in detail.

All the tests were carried out on previously mentioned production servers. The test subjects were categorized under following steps:

- Performance Tests: Relational vs. OLAP
- Observation of the additional requirements to have an OLAP Cube
    - Update source DM: Daily and monthly update
    - Response to new development needs
        - Add a new Fact Field to the system
        - Add a new Dimension Table to the system
        - Delete an existing Dimension Field from the system
        - Delete an existing Fact Table from the system
    - Process the Cube: Daily, monthly and full process
    - Total amount of storage necessary to maintain the Cube

Performance Tests: Relational vs. OLAP

First test subject was the comparison between the run durations of the reports on relational and multidimensional environments. As mentioned previously, the current MI reporting system of the Bank was based on scheduled reports, which have been developed using SAP Business Objects 6.5 and utilized the corporate DW as the source database. For the performance comparisons, a set of eight reports with distinct characteristics was chosen from the current reporting system, which differed in the following features:

- *Report complexity*: Query execution cost, which is an estimate made by the relational database agent and describes the number of data blocks required to complete the query.
- *The number of joined tables*: The number of table joined within the query.
- *Row count of the queried tables*: Small or large number of rows queried.
- *Row count of the result set*: Small or large number of rows returned.
- *Number of queried terms*: Query a single term or construct trend relation between several terms (EOM's).

The actual report definitions with their characteristics were shown in the below table.

Table 9. The Report Definitions.

Report ID	Report Name	Report Properties				Characteristic
		Joins	Rows Queried	Rows Returned	Terms Queried	
1	Deposit Balance Total - Segment Base	Low	Low	Low	Low	No Aggregation - Medium Source Data
2	Deposit Balance Total - Segment & Region Base	Low	Low	High	Low	Medium Aggregation - Medium Source Data
3	Account Count - Business Unit & Product & Segment Base	High	Low	Low	Low	Many Joins - Medium source data - Medium Aggregation
4	Account Count - Segment & Region Base	High	Low	High	Low	Many Joins - Medium source data - High Aggregation
5	Customer Profitability - Segment & Product Base	Low	High	Low	Low	Very large source data - Medium Aggregation
6	Customer Profitability - Segment & Region Base	Low	High	High	Low	Very large source data - High Aggregation
7	Balance Total - Region & Segment Base (Trend)	High	High	High	Med	Medium Trend - High Aggregation
8	Customer Count - Segment Base (Trend)	High	High	Low	High	High trend relation - Medium Aggregation

To prevent potential performance advantage of the powerful DW of the Bank, chosen reports were redeveloped in the same formats on the new relational DM. On the other hand, MS SQL Server 2005: Reporting Services (SSRS) were used to prepare the static reports on the OLAP Cube. While the utilized reporting software were susceptible to mislead the results by adding a lag for presenting the data in the desired format, SQL and MDX (MultiDimensional eXpressions; used for querying SSAS Cube) scripts were exported from the report designs and then run on the relational and OLAP environments in order to observe only the execution durations of the queries on the servers. The SQL queries were run with the SQL*Plus, which is Oracle's basic command line utility and the MDX queries were run by using Reporting Services. These queries didn't produce the exact same result due to the architectural differences between them. MDX is multidimensional by nature and the results set can be compared to a spreadsheet with column and row labels whereas SQL queries produce tables that only have column labels. However, the information that can be interpreted from the data was the same. The durations of the MDX queries were calculated using SQL Server Profiler, a tool that logs all events, which occur during any operation on the Cube, with timestamp and duration. On the other hand, durations of the SQL queries were calculated using the timestamps automatically taken before and after the query runs.

Additional Requirements to Have an OLAP Cube

At the next phase of the performance tests, the proportion of the "additional" progress needed to accomplish reporting by using an OLAP Cube was aimed to be observed. In order to do that, some scenarios have been generated and they were tested. For observing the additional requirements to build and maintain an OLAP Cube, the following subjects were evaluated:

- Update source DM: Daily and monthly update
- Response to new development needs
    - Add a new Fact Field to the system
    - Add a new Dimension Table to the system
    - Delete an existing Dimension Field from the system
    - Delete an existing Fact Table from the system
- Process the Cube: Daily, monthly and full process
- Total amount of storage necessary to maintain the Cube

Update Source DM: Daily and Monthly Update

As mentioned at the previous chapter, the amount of data loaded into the DM for one term was about 5 GB, which consist of over a hundred million rows loaded into twenty tables. These numbers were valid for both monthly and daily table updates. The necessary effort for this update process was tried to be observed in this section. The statistics were gathered from the DataStage job logs and the average values were calculated.

Response to New Development Needs

There were a number of sub-tasks, which would ease observing the total time necessary for each modification scenario that were examined in this part of the tests. A brief explanation of the works conducted under each task is provided in the following table.

Table 10. Tasks for Accomplishment of the Scenarios in Relational and Cube Environments.

Task Name	Task Definition	Classification
Alter database logical model	In order to manage the metadata and the logical DM model, the modifications should firstly be done in the Erwin model.	Optional for Relational and OLAP since it was required by the regulation of the Bank.
Alter database physical model	Add or remove the new row/table to the Cube DM.	Necessary for Relational and OLAP.
Modify ETL job design.	To alter the source system, ETL jobs should be modified in order to fit the new DB structure.	Necessary for Relational and OLAP.
Deploy ETL jobs	Jobs are modified at the development level and then deployed to the production DataStage Server.	Optional for Relational and OLAP since it was required by the regulation of the Bank.
Re-run ETL jobs	To fill the new table/row. Details were given in the Update source DM section.	Optional for Relational since necessity may come from the emergence of reflection of the changes to the system. Necessary for OLAP since any model change necessitates Full Process of the Cube and the source tables had to filled before that.
Alter Cube model	In order to reflect the changes to the Cube environment, the Cube DSV and the model should be changed.	Necessary only for OLAP.
Deploy Cube	After the Cube model was changed, it needs to be deployed to the production server.	Necessary only for OLAP.
Process Cube	After a model change, the Cube becomes inoperative and need to be processed to reoperate.This task will be introduced in next section.	Necessary for OLAP since any model change necessitates Full Process of the Cube.
Modify report template	Static SSRS and BO reports should be modified and deployed to the production environments.	Necessary for Relational and OLAP.
Process report	Reports should re-run to publish the changes to the users.	Necessary for Relational. Unnecessary for OLAP since the reports run for a short time and can be accessed directly from the Cube.

Tasks for accomplishment of the scenarios in relational and Cube environments are classified as ‘Necessary’, ‘Optional’ or “Completely Unnecessary”. “Necessary” is a step that must be fulfilled in order to implement the scenario while “Optional” is a step without which the scenario can still be completed. However, in order to see the results immediately or to obey the development rules of the Bank, these steps may still be needed. Table 11 summarizes the environment based necessities to apply a change to the Relational and OLAP environments.

Table 11. Environment Based Necessities to Apply a Change to the Relational and OLAP Environments.

	Relational		OLAP	
	Necessary	Optional	Necessary	Optional
Alter database logical model		x		x
Alter database physical model	x		x	
Modify ETL job design	x		x	
Deploy ETL jobs		x		x
Re-run ETL jobs and fill the new table/row		x	x	
Alter Cube model			x	
Deploy Cube			x	
Process Cube			x	
Modify report template	x		x	
Process report		x		

The tests in this section that measure duration were mainly related to necessary human effort required to accomplish the tasks. Therefore, the calculations were mostly made on the basis of MH (Man/Hour) as mentioned in the Chapter 4 and the results were rounded to the nearest quarter of an hour. It should also be mentioned that each task was valid and would be evaluated for each scenario; however, the durations of the tasks would differ by the complexity level of the scenario.

Process the Cube: Daily, Monthly and Full Process

As stated previously, Cube processing is necessary to see the results reflected in the Cube while it was modelled with the MOLAP functionality. This requirement can be divided into three sections:

1. *Processing the current partitions* with the scheduled SQL Server Management Studio job or manually if the modification needs to be reflected immediately.

Daily process of the Cube was scheduled with the SQL Server Management Studio. The job runs daily without any manual interference and processes the current partitions without controlling whether the source tables were updated or not. The duration was calculated through the Management Studio event logs from a sample of ten randomly selected runs. If there was an urgent need to reflect a change in the model, the current partitions could be processed manually.

2. *Processing a monthly partition* when an EOM data need to be filled. There are necessary tasks that should be accomplished manually before processing the monthly partitions, which can be described as;
    - A new partition should be added to all of the measure groups and the binded query should be modified in order to retrieve the new partitions' data from the source DM.
    - As the DW of the Bank does not contain consistent data (that means different tables may include data belonging to different dates), all the source tables should be controlled before processing the monthly partition to ensure that all the tables were filled with the new term's data successfully.
    - New partitions have to be processed manually.

3. *Processing the whole Cube (Full Process)* would be *necessary in following situations:*

    - When there is a change in the source data; generally in the dimension tables; the Cube may wanted to be re-processed for the previous term in order that they involve the measures calculated around the new dimensions. A change in the segments of customers may be an example for such kind of changes. In such situations Cube is reachable and dirty read is enabled during the full process. Processed partitions are up to date and other ones contain the former data. The refreshed partitions begin to contain new data one by one. This type of full process is also valid for the changes applied to the formulation of a calculated variable within the Cube.
    - When there is a change in the metadata, which the Cube was built over; in other words in the DSV. In this type of situations, the Cube becomes inoperative, which makes it unreachable. When all the dimensions and at least one partition of each measure group were processed, the Cube becomes reachable again, but only the processed partitions contain data differently from previous method. The partitions are processed one by one and brought into use. All the partitions need to be processed in order Cube to fully operate again.

In this study, the full process of the cube includes manually processing all the dimensions and the partitions of the measure groups since the data that they contain are very large.

Total Amount of Storage Necessary to Maintain the Cube

This section introduces all storage requirements necessary to physically implement and maintain an OLAP Cube with such large size while the aim of this part of the thesis was to demonstrate the additional requirements of having an OLAP Cube. An approximate total value was calculated by summing the sizes of the tables at the source DM and the actual Cube. The additional storage capacity required to add a new term's data into the Cube was also evaluated.

After these tests, another scenario was applied in order to see the reflections of the additional duration to modify the Cube environment in comparison with the duration of running the reports at the relational side. A sample change request was evaluated in order to identify the trade-off between the Cube's model modification and relational report processing tasks. Then a break even analysis was performed to find out when the Cube becomes advantageous. The details of this analysis are demonstrated in next chapter.

CHAPTER 7

EXPERIMENTAL RESULTS AND COMPARISONS

This chapter introduces the empirical results gathered from performance tests, makes comparisons between relational and OLAP reporting systems and presents the findings from questionnaires.

The results of this study obtained using several methods could be grouped under two categories regarding the type of the information they contain as described in Chapter 4: Quantitative and qualitative. The qualitative results were attained by conceptual analysis; through the review of theory and literature and concluded at the end of the Chapter 3. The results from empirical phases of the study (from the experiment and questionnaires) were obtained by analyses of quantitative data.

In brief, quantitative results involve:

1. The findings obtained through performance tests and comparisons between two systems,
2. The findings on additional requirements to have an OLAP cube,
3. Results from the questionnaire, which was conducted to verify or refute both experimental findings and some conclusions from theory concerning functional properties of two systems that can be found in Appendix G.

Results from Performance Tests: Relational vs. OLAP

This part of the results describes the findings obtained through the performance tests that were explained in detail at the previous chapter. As mentioned, empirical evaluation was accomplished by making comparisons between two systems.

Eight reports, which differ by their complexities (query cost), number of joined tables, row count of the queried tables, row count of the result sets and number of queried terms were chosen for performance evaluation. The characteristics of these reports were described verbally in Chapter 6 and the actual values of these features are provided in the Table 12.

Table 12. Characteristics of the Reports Chosen for Performance Evaluation.

Report ID	Query Cost (Data Blocks)			Number of Rows		Number of Joined Tables	Number of Terms Queried
	Cost of Data Read	Cost of Aggregation	Total Cost	Queried (Raw Data)	Result Set (Aggr. Data)		
1	346,604	2,302	348,906	21,765,266	19	7	1
2	354,824	482,490	837,314	21,662,215	12,972	8	1
3	402,736	486,517	889,253	36,085,108	479	21	1
4	400,149	616,620	1,016,769	35,643,458	13,084	21	1
5	559,547	985,943	1,545,490	78,196,544	23	6	1
6	520,842	915,546	1,436,388	78,090,185	11,532	6	1
7	1,123,588	4,886,493	6,010,081	-	13,303	39	2
8	1,482,516	1,487,708	2,970,224	349,373,327	766	39	12

In this case, the most costly operations occur while reading the data from the database and then aggregating them hierarchically into several dimensions. In order to observe the complexity of these tasks individually, the costs were given for Data Read and the following Aggregation processes, separately and demonstrated within the Figure 19. Report 7 was excluded because of its different characteristic that will be explained later.

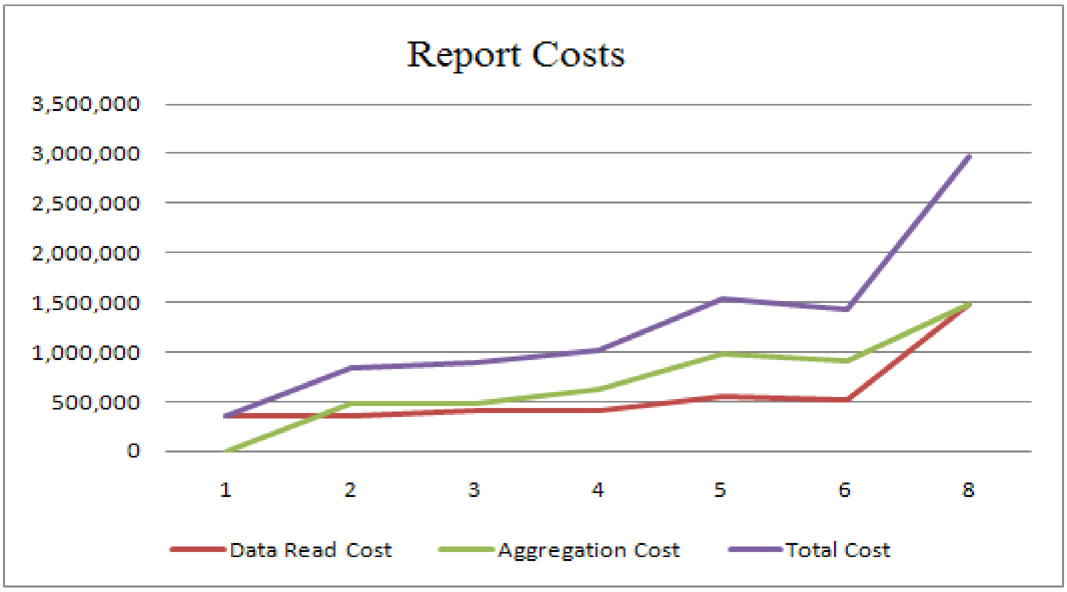
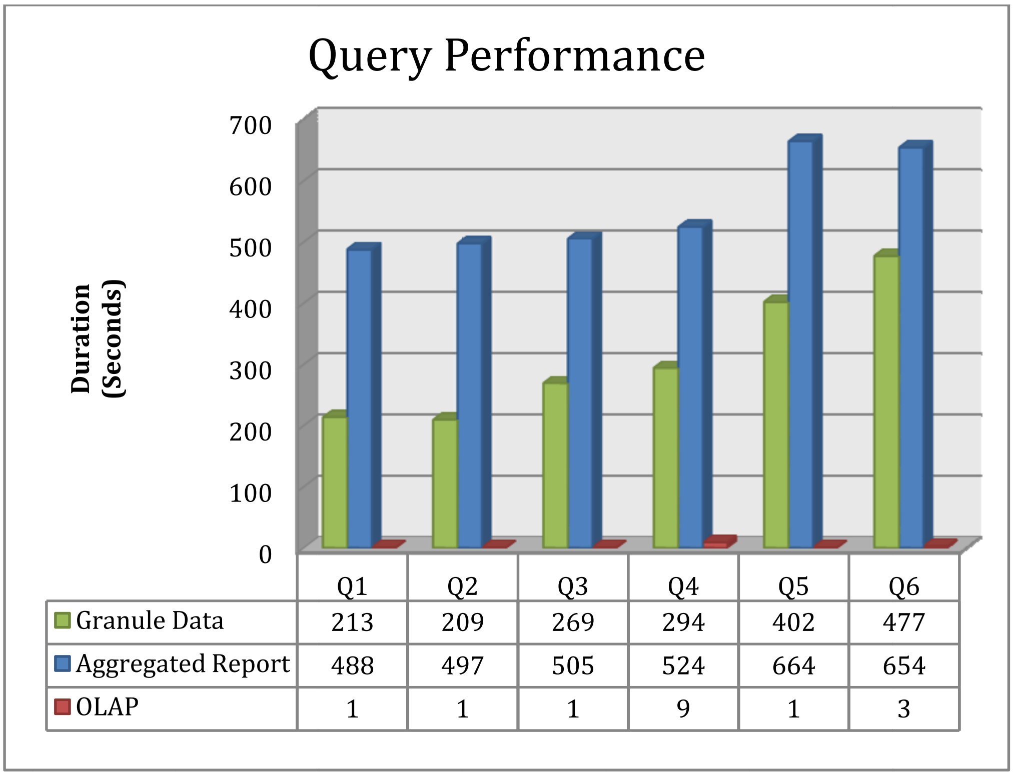


Figure 19. Report costs.

Data Read describes the cost of reading data from the DM, joining the tables and presenting them granularly to the database engine, which will apply further aggregations. The *Aggregation* costs represent the additional cost that arises from the grouping and sorting necessities of the reports. In other words, if the reports would not attempt to aggregate and summarize all that data into dimensions with hierarchies, the total cost would be equal to the Data Read cost. However, Aggregation cost is a must to undertake since the distinctive feature of the MI reports is their specialty to present information to the top managers in a very aggregated way.

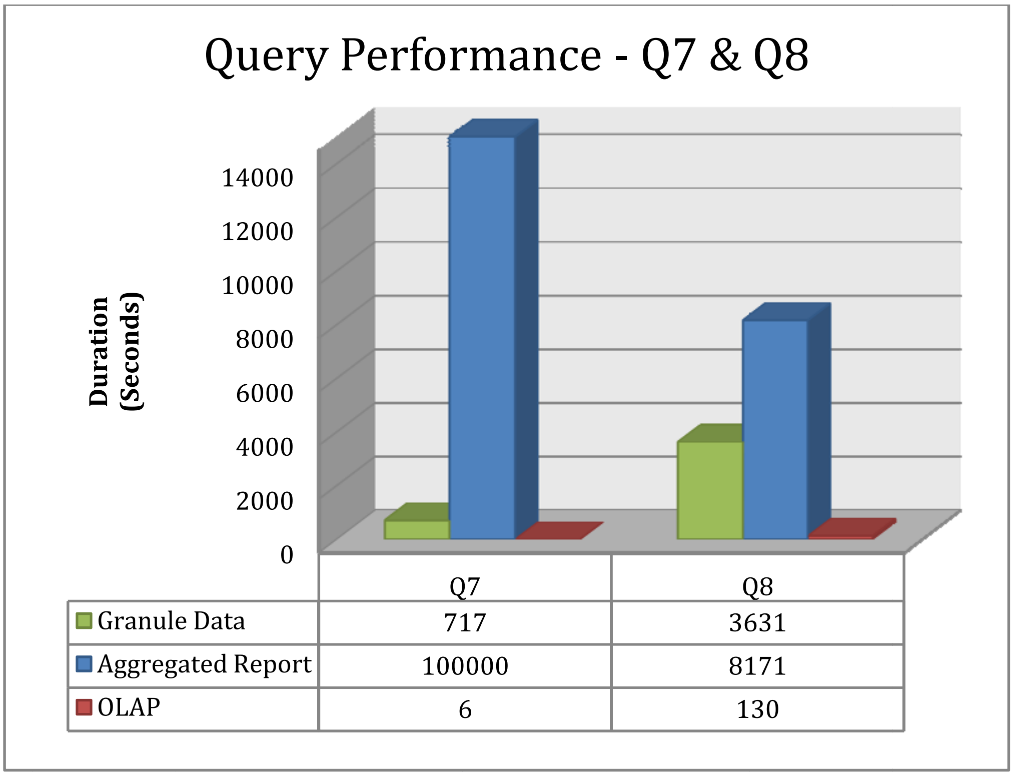
The number of joins in the SQL statement and the number of rows queried/resulted were also provided in Table 12. Number of queried rows represents the raw data used in the MI reporting, which has not yet been aggregated. On the other hand, the number of rows at the result set gives the exact counts of the actual reports.

All the reports were run on both systems each for five times and produced the results demonstrated in Figure 20 and Figure 21 (For the duration of each trial, see Appendix H). In order demonstrate the durations of each report more precisely, results of the trendy reports were presented separately. Aggregated report duration of the R7 was given as an estimate since it didn’t return any result and this issue will be discussed later.



	Q1	Q2	Q3	Q4	Q5	Q6
🟩 Granule Data	213	209	269	294	402	477
🟦 Aggregated Report	488	497	505	524	664	654
🟥 OLAP	1	1	1	9	1	3

Figure 20. Results of the performance tests for first six queries (Query Performance of R1 – R6).



	Q7	Q8
🟩 Granule Data	717	3631
🟦 Aggregated Report	100000	8171
🟥 OLAP	6	130

Figure 21. Results of the performance tests of seventh and eighth queries (Query Performance of R7 – R8).

As mentioned above, querying the data from the DM and aggregating them under dimensions are two most costly operations that occur during the execution of the queries. Therefore, the query run durations on the RDBMS with (*Aggregated Report*) and without *(Granule Data*) aggregating the data are presented separately in order to make better conclusions.

Findings from the Observations on the Additional Requirements to Have an OLAP Cube

As stated previously, OLAP is a system that is built over a relational database. The additional requirements to have such a system were tried to be observed quantitatively by the results given in the following steps, whose contents were explained in detail in the previous chapter.

Update Source DM: Daily and Monthly Update

The Bank's experts stated during the analysis of existing system that DM update process takes place daily and monthly. The quantity of the data loaded within both processes was nearly same, only the source tables were different. DM update durations and some other information are given in Table 13.

Table 13. Daily and Monthly Data Mart Update Durations.

	Tables Loaded	Rows Loaded	Data Size	Trigger	Duration (min)
DM Daily Update	20	> 100M	5 GB	Automatic	105
DM Monthly Update	15	> 100M	5 GB	Manual	180

Monthly job was triggered manually, because the Profit measure table's data were loaded into the DW at an indefinite time each month whereas all other tables were loaded at the first day of month. Thus, it needs to be checked and then the monthly transfers should get started.

On the other hand, for each month, a new partition was created in the measure tables automatically while ETL jobs were being performed and about 5 GB of data were stored collectively within these partitions. As mentioned before, the DM was containing fifty-one partitions with a total size of 250 GB at that time.

Response to New Development Needs

Four different scenarios were identified regarding the directions form experts of the Bank, which represent change requests as in the real-life environment for such kind of reporting. These scenarios have been divided into sub-tasks and each of them was explained in detail in Chapter 6. At this phase of the study, these tasks were evaluated and the average durations were calculated for each to accomplish and then, a greater induction was performed to see the duration of each scenario. Table 14 presents the results. Appendix I and Appendix J also contains the duration of each trial in detail.

Table 14. Response Times to New Development Need.

Duration (MH)	S1		S2		S3		S4		Average	
Task Name	Rel.	OLAP	Rel.	OLAP	Rel.	OLAP	Rel.	OLAP	Rel.	OLAP
Alter Database logical model	0.1	0.1	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1
Alter Database physical model	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Modify ETL job design	0.3	0.3	0.5	0.5	0.1	0.1	0.1	0.1	0.3	0.3
Deploy ETL jobs	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Re-run ETL jobs	0.8	0.8	0.3	0.3	0.0	0.0	0.0	0.0	0.3	0.3
Alter Cube Model	-	0.2	-	0.8	-	0.1	-	0.5	-	0.4
Deploy Cube	-	0.1	-	0.1	-	0.1	-	0.1	-	0.1
Process Cube	-	60.0	-	60.0	-	0	-	0	-	30.0
Modify report template	0.2	0.5	0.3	0.8	0.1	0.3	0.2	0.5	0.2	0.5
Process report	0.4	-	0.4	-	0.4	-	0.4	-	0.4	-
Total Duration	0.60	62.00	0.90	62.55	0.30	0.65	0.40	1.30	0.6	31.63
(Only Necessary Tasks)										
Total Duration	2.00	62.20	1.90	62.85	0.90	0.85	1.00	1.50	1.45	31.85
(All Tasks)										

As mentioned previously, the results shown in above table were calculated on the MH (man/hour) basis. Each result was an average of five trials and rounded to the nearest decimal point. Coloured cells represent the optional tasks, which have been introduced in Chapter 6.

The followings are some assumptions made while evaluating these tasks:

- It was stated that any change in the DSV requires full processing of the Cube. While implementing S3 and S4, DSV was not modified, only table/field definitions were removed from the Cube model. Therefore, there was no need to full process the Cube.

- Re-running the ETL jobs involves only transferring the modified table and for the current term.
- Relational reports were run only for the current term.
- Processing the Reports on the relational side was marked as optional depending on the assumption that they would run on next day as they were scheduled. It must be accomplished in order that the changes can be reflected to the reports.
- However only the current term was required for the reports, in order the Cube to fully operate, all the partitions were processed.

Process the Cube: Daily, Monthly and Full Process

The types of processes were described in Chapter 6 with their sub-tasks. Durations of four types of process were given in the below table.

Table 15. Durations of Four Types of Process.

Process Type	Process Method	Duration
Current Partition	Automatic	137 min
Current Partition	Manual	2.5 MH
Monthly Partition	Manual	6 MH
Full Process	Manual	60 MH

The duration of the daily automatic process was gathered through the SQL Server Management Studio job logs. The durations of the other manual processes were calculated by taking the average of five trials performed for each process (two for Full Process).

Processing the Cube has been done with a parallelism level of four, which determines the maximum number of partitions that can be processed in parallel at a time.

Total Amount of Storage Necessary to Maintain the Cube

For both OLAP and relational reporting, a DM had to be developed. In the existing case, it was containing data about 250 GB and being increased at a rate of 5 GB per month.

The OLAP Cube built on this relational DM had a size of 250 GB and it grew again with a rate of 4.5 – 5 GB per month.

OLAP reports do not need to pre-run and be stored in a particular place since they are very fast and can be reached any time from the Cube. On the other hand, relational reports run for long durations, therefore, they need to be stored in a server in order that users can reach them quickly. Size of these eight reports would be negligible when compared to the sizes of the DM and cube data files. However, the actual number of reports at the current system is about forty and storing all of them for every term engages considerable data sizes.

Summary of Results from the Experiment

In this section, the results are summarized and a comparison between both systems is conducted.

At the first phase of the tests, durations for the report process were calculated. The results showed that OLAP had a big advantage in report processing compared to the relational system. Relational reports were evaluated with and without aggregating the data. These were resulted in an average duration of 776 seconds for the non-aggregate queries and 1,643 seconds for the aggregated versions. On the other hand, the average report duration of OLAP was nineteen seconds. When compared to the average duration of the OLAP reports, the relational ones lasted 8,547 percent longer.

The reports were grouped under four sets according to their characteristics. Each set consisted of two reports.

First group of reports were containing queries (R1, R2) with the simplest characteristics. Queries had a relatively low number of rows read from DB, they were not highly complex and a very limited number of tables were inquired. Due to these features, mentioned reports run fastest among the others. However, they still lasted an average of 492 seconds.

The second set of queries (R3, R4) joined lots of tables, which also contain relatively low number of rows. The duration of these reports were close to the results from first group of queries and it seems that using many joins do not affect the query costs too much if the total number of rows in all the source tables were small.

The third set (R5, R6) queried smaller number of tables, which contain large amount of data. The results of these tests show that data reading lasts long, but the duration necessary to carry out the aggregations is also very close to the results of first two sets. There is a positive relationship between the queried rows and the data-read part of the query duration that can be seen in Figure 22. On the other hand, it seems that aggregating a higher number of rows under the same dimensions as used at the previous two sets do not affect the aggregation cost.

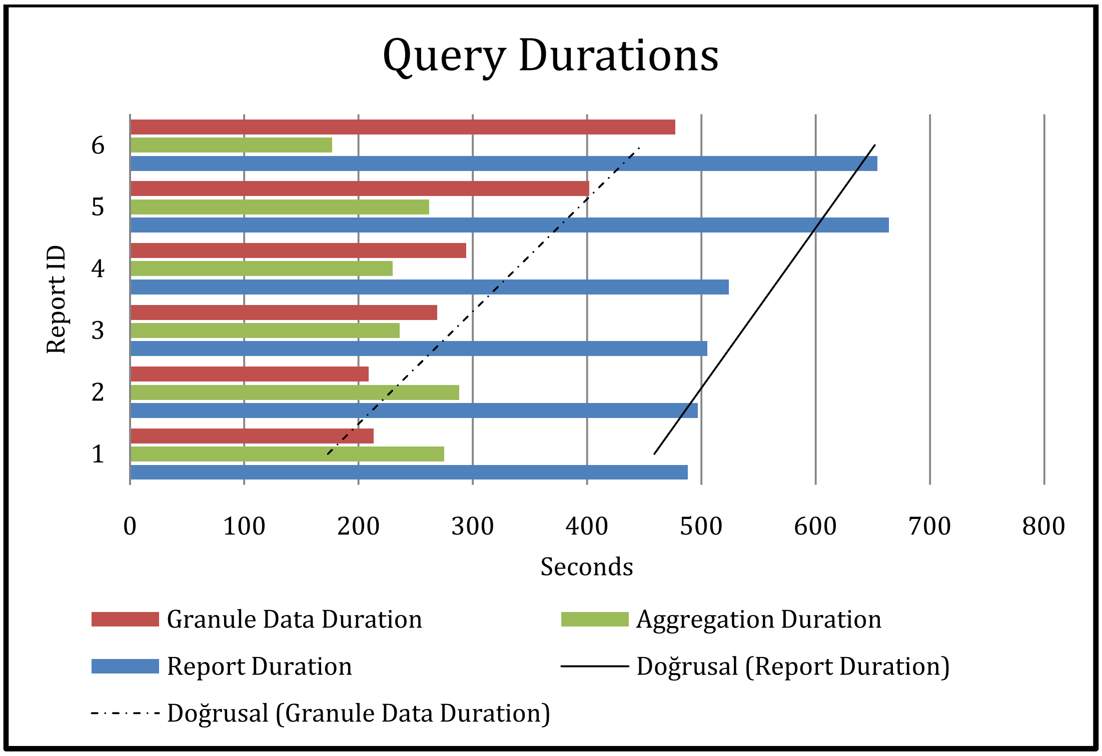


Figure 22. Relationship between the queried rows and the data-read part of the query cost.

Last set of reports (R7, R8) involved the most complex queries. They composed trend relationship between several terms; inquired a very large number of rows by making many table joins and read very large amounts of data from the DB. R8 was queried a period of twelve months and attempted to show the trend relationship between them. However, it was tried to group the measures under only one set of dimensions. This query lasted for more than two hours and aggregated around 350 million rows under the Segment dimension. On the other hand, R7 of this group, which queried only two terms, but tried to group the measures under two sets of dimensions that have thirteen thousand different combinations. At first, it was run without aggregating the data and lasted 717 seconds. The duration was shorter than R8 since the number of queried terms was smaller. Then, the actual report with group by expression was tried to be run several times, however it didn't return any result. This shows that there is a limitation at the relational side; a query can compose aggregations up to a maximum level and for the further needs, it does not work at all. However this implication should further be tested in order to prove its validity, which would be a subject to future woks.

Based on the above results, it can be seen that the number of queried rows has a direct effect on the query cost and run duration. When it increases, the cost and duration increase as well almost with the same ratio. It was also seen that the number of rows returned and the number of joined tables do not have much effect to the cost and duration.

On the other hand, increasing the number of terms queried and trying to conduct a trend relationship between several terms increases the cost and durations dramatically. Another significant criterion is the number of dimensions queried. It also increases the costs by multiplying the amount of aggregations that have to take place during the grouping operations.

All these characteristics are essential for the MI reports and as seen in above results, some of them become critical weaknesses of the relational environment when reporting such kind of information compared to the OLAP Cube.

At the second phase of the tests, the additional requirements of having an OLAP Cube were evaluated in the context of four topics; updating the source DM, total amount of storage necessary to maintain the Cube, processing the Cube and response to new development needs.

1. A DM has already existed on the DW for the current relational reporting environment of the Bank. Within this study, this DM was replicated on an external

server in order to develop the Cube reliably. This process brought a load of building the same DM on another source and transferring the data from DW to that source regularly.

2. It was expected that the Cube would have a smaller size since it kept only the summary data at the source DM. However, the total amount of storage necessary to maintain the Cube was almost equal to the size of the DM itself. Since sparsity was not an issue for the current Cube, it seems that the number of combinations of the dimensions and grouping the large amount of data in the fact tables around those dimensions enlarged the size of the Cube.
3. Processing the Cube was one of the most costly operations due to the long process durations and manual interference needs during monthly and full processes, which could not be automated because of the size and structure of the Cube. Further practices might try to accomplish that.
4. Response to the new development needs was investigated in detail at the previous sections. The scenarios have been selected with the directions of the experts of the Bank. It has been seen that a possible change requests would most probably necessitate a model change at the Cube, which requires the Cube to be full processed.

The results show that, without including the process of the Cube, the total average durations to implement a change in OLAP are very close to those in the relational environment. The changes in the relational system lasted for 0.6 MH whereas they lasted 1.63 MH in OLAP without including the optional steps. On the other hand, when the optional tasks were included, this duration becomes 1.45 MH to 1.85 MH. It can be said that except the processing needs of the Cube, the change durations are close for both systems so that modelling with both seems to have equal flexibility. However, when the processing of the Cube was included, the results became 1.45 MH to 31.85 MH, which makes OLAP less flexible to changes.

The scenarios that have been generated regarding the opinions of the experts and investigated within this study were concerning the changes to only be reflected to the current and the forthcoming terms. This was the case for the scenarios with the current system up to now, because reflecting any change to the previous terms was very hard to implement. In order to accomplish such a scenario; the DM tables had to be refilled with the previous terms' data and static reports required to be run all over again. Especially rerunning the reports was meant to be a very painful operation since it would last for the duration of the longest report multiplied by the number of previous terms. Also the number of the static reports was about forty and rerunning each of them would mean rerunning about two thousand reports.

On the other hand, since the OLAP reports do not need to run beforehand (since they can be taken quickly from the Cube), when the Cube is full processed, all the reports become available for all the previous terms. In order to clarify the reflections of the additional duration to modify the Cube environment in comparison with the duration of running the reports at the relational side, another scenario was evaluated.

As mentioned previously, to get the reports from the Cube, it is not necessary to wait for all the partitions to be processed. Desired terms can be queried by processing only the necessary partitions. Full processing was only needed in order to operate the Cube fully. Considering this feature, a sample change request was evaluated in order to identify the trade-off between the Cube's model modification and relational report processing tasks. To accomplish that, below scenario was generated and the durations were tried to be adapted to the findings from the previous part, where the additional cost of having an OLAP Cube was tried to be observed. Parameters of the scenario and the results of the evaluation are provided in Appendix K and accordingly a break even analysis was performed to find out when the Cube start to be advantageous, which is demonstrated in Figure 23.

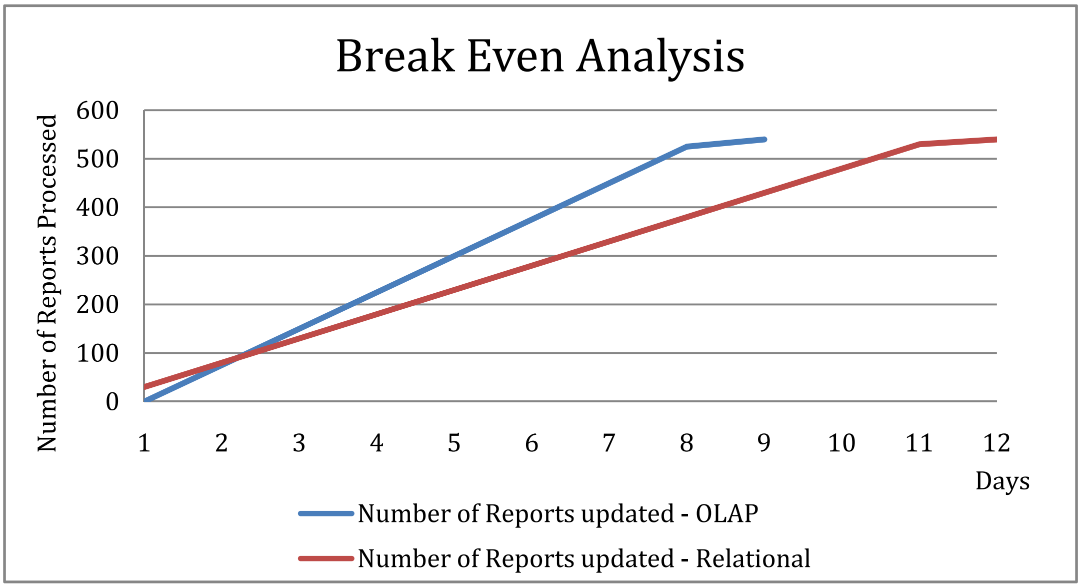


Figure 23. Results of break even analysis.

With the given scenario, the break even point was calculated as 2.2 days. Up to that time, more reports were generated within the relational environment since processing the partitions of the OLAP Cube could be started after one day. This was because the model changes and report modifications at the OLAP side lasted longer than the relational system. On the other hand, the total time to generate all the reports from the OLAP system took nine days whereas this duration was twelve days for the relational side.

The reason that the analysis was conducted by using the number of reports generated was to be able to generalize the findings into other possible scenarios intending to change something in some reports and to process them for a certain number of terms.

Several extremes can be introduced just to give idea about the possible situations. For example a change request may necessitate to add a new field only into one report, which runs for only ten minutes and to process it for only one month. Another change may require modifying all reports and processing all of them for all the terms. For the OLAP side, both of these scenarios would take the same time to implement. On the other hand, parameters used in above example can be modified in numerous ways and the necessary duration to accomplish each scenario can be evaluated. However, according to BI experts whose opinions were taken into consideration to determine test parameters, those extreme scenarios would not probably be requested during banking operations.

As mentioned above, all the reports become available for all the previous terms when the Cube was once fully processed since they do not need to run beforehand. Thus, making past projections of the newly added measures would become possible. This new perspective might invert the weakness of OLAP in flexibility towards changes, with respect to long lasting process needs, into an advantage compared to the relational environment. Business requirements would change with such an opportunity and trends of the new measures would be observed easily. Feasibility of this issue would be subject to the further studies on this field.

Results from the Questionnaires

The above findings from the experiment were tried to be verified or refuted through a questionnaire conducted to gather the opinions of IT experts in the field of banking, which is provided in Appendix G with the results. The questionnaire was divided into four parts as follows:

3. Personal information
4. Questions about performance issues of relational and/or OLAP Cube reporting
5. Questions about additional requirements to have an OLAP Cube
6. Questions about general features and/or functional properties of relational and/or OLAP Cube reporting.

There was also an "additional comments" section at the end of each group of questions to clarify the responses of the experts verbally.

In the first part, it was attempted to measure the profile of eleven participants who responded to the questionnaires and their experience on the field of BI. The profile of the participants was composed of several areas of BI as there were; two BI directors, a BI project manager, a database administrator, several BI developers, several reporting specialist and an ETL developer. The average experience of these experts was given below:

• Number of years experience with BI:	4.73
• Number of BI projects participated in:	9
• Number of OLAP projects participated in:	1.82

These numbers show that the respondents answering this questionnaire consist of experts on the field of BI and most of them had experience on the OLAP projects. Even though the number of participants was limited, their experienced profile provided significant results.

The following parts of the questionnaire were analysed by calculating mean values of the answers to the questions, which were on the Likert scale. The intervals shown in Table 16 describe the agreement levels of the participants to the questions. All the questions were highlighted with the corresponding colour according to the mean values of the answers.

Table 16. Intervals of Likert Scale.

Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree
2.00 <---> 1.21	1.20 <---> 0.41	0.40 <---> -0.40	-0.41 <---> -1.20	-1.21 <---> -2.00

The second group of questions (statements) were about the findings from the performance comparison of the reporting process durations of both systems (see Table 17).

Table 17. Results from the Questionnaire on Performance of OLAP.

1	Execution cost of a query generally is a good estimate, which describes the performance and complexity level of a query.	1.36
2	Number of queried rows, number of rows at the result set and the number of tables joined within the query are some other important features that determine the complexity of the queries.	1.45
3	For the relational reporting, grouping (aggregation) operations can be more costly than reading the data from the database.	1.27
4	Operations such as complex data aggregation, comparison and consolidation are more difficult in relational method than OLAP.	1.18
5	OLAP provides very high query performance when it is compared with relational model.	1.55
6	A good opportunity for using an OLAP cube in comparison with relational reporting is time intelligence (availability to make trend analysis).	1.55
7	In relational reporting, even the most powerful hardware and the most optimized database cannot provide the performance of OLAP.	-0.36
8	Running the SQL and MDX queries rather than the formatted reports provides better results on performance since data presentation duration would be eliminated.	1.00

First three questions were asked to evaluate whether the method used to determine the complexity levels of the queries was reasonable. Ninety-four percent of the participants agreed to all three statements that describe the complexity of a query which would be determined by its execution cost, number of queried rows and the number of tables joined within the query.

Eighty-one percent of them also endorsed the idea considering that aggregating, comparing and consolidating data within the relational system is harder than doing those operations in an OLAP system. Additionally, all of them supported that OLAP performs better in conducting trend analysis.

About sixty-five percent of the participants strongly agreed that the OLAP systems have much greater reporting performance compared to the relational system.

However, they didn’t agree with the statement, which proposes that there is no way to catch the performance of the OLAP queries through a relational system. They declared in additional comments section that, recently, very powerful DW appliances such as Teradata, Netezza and Exadata were developed. These machines combine the power of DB and hardware by using special data distribution systems and parallelism techniques. These tools also provide methods to improve analytical ROLAP performance such as materialized views or aggregate join indexes (AJI). Some of the respondents surprisingly added that these systems could reach the same performance of MOLAP cubes by using ROLAP models.

Finally, ninety-one percent of the participants agreed that running the raw SQL and MDX queries rather than running the actual formatted reports, which were developed via some high level reporting applications was a better way to observe the actual duration of the queries running on both systems. Because they thought that placing an application layer would mislead to make some imprecise calculations.

Additional requirements of having an OLAP Cube were inquired within the third part of the questionnaire which contains the questions given in Table 18.

Table 18. Results from the Questionnaire on Additional Requirements of Having an OLAP Cube.

9	Building a data mart is necessary for relational systems in order to efficiently report such kind of information since there are complex aggregations and queries need to run on optimized data models	1.27
10	Building an external data mart is necessary for OLAP systems since the intensive use of the database during the Cube processing would negatively affect other usages of the system.	-0.09
11	Maintenance of OLAP systems is generally harder than the relational model.	0.82
12	The necessary effort for daily and monthly update of OLAP cube is greater than the relational system because of OLAP's processing needs.	1.27
13	An OLAP Cube of this size necessitates manual interference for monthly and full processing	0.18
14	Any change in OLAP model or the major changes at the source data necessitates full processing of the Cube	1.55
15	OLAP Cube generally requires longer duration than relational system to respond the new development needs.	1.09
16	If full processing the Cube was not necessary, a change can be implemented approximately within the same duration with the relational system.	0.36
17	Due to long process durations of OLAP Cubes, up to dateness with the DW may not be maintained.	0.82
18	The necessary disc capacity to have an OLAP Cube is almost equal to the data size of the relational database.	-0.09

The current management reporting system of the Bank was utilizing a relational DM, which has been modelled as a star schema. Having a separate DM for such kind of reporting was also suggested by ninety-one percent of the participants. They also shared the idea of having a DM in order to build the Cube; however, they were not agreeing with the statement suggesting having that DM externally due to the performance issues. They stated that the Cube process should be scheduled into nighttimes in order not to cause any performance problems. However, performance issue was not the only reason to create an external DM therefore; it should be well decided whether to build a DM internally or externally while implementing such a Cube.

Results from the experiment related to the maintenance needs of both systems were tried to be assessed through the questions 11, 12 and 17. Results of the experimental phase of the study have shown that, in general, maintenance of an OLAP system is harder than the relational system. Most of the participants (sixty-three percent) agreed with this proposition. Also, fifty-five percent of them agreed that the necessary effort to update the Cube is greater than relational system because of its long process durations. They added that, if the Cube was modelled with ROLAP functionality, these durations would be equal. Finally, it was stated that the disadvantage caused by the process requirements of the Cube would prevent it from being up to date with the DW; which was also the common opinion among the participants. On the other hand, some of the participants suggested that by building the Cube directly on the source transactional systems rather than the DW this situation would be prevented. However, the feasibility of such an implementation was open to questions.

Twenty-seven percent of the participants disagreed that there is an inevitable need to have manual interference for monthly and full processing of the Cube. Forty-five percent of them were undecided on this issue. Some of the participants stated that the product based limitations may exist, but in practice, those processes should also be carried out automatically. Another participant added that manual interference is only needed for complex DM designs, which was the situation in existing case; to follow tables loaded properly or in order not to start cube process.

In Chapter 6, it was stated that any change in OLAP model or the major changes at the source data necessitate full processing of the Cube. The participants confirmed that this necessity was existed for the other OLAP products at the market and it is an inevitable need. A hundred percent of them agreed with this statement.

During the implementation of a change request to both systems, OLAP takes a longer time because of its full process need. On the other hand, the results stated that, except the process duration of the Cube, the response times of both systems to the development needs were very close to each other. Eighty-two percent of the participants agreed that the OLAP would last longer to respond to a new development need. However, they added as that the reason was not only the process duration of the Cube, but depending on the software used, modelling would also last longer.

Regarding the capacity issues, the results indicated that OLAP had nearly an equal size with the whole source DM, but the participants suggested that this could not be generalized. They stated that, in theory, OLAP should read granule data, aggregate them among dimensions and store only the summarized data. So, they said that the Cube should have a smaller size compared to the DM. There might be unnecessary aggregations within the implemented Cube and it would be expanding its size. By making some further optimizations, the size would be tried to be reduced as a further task of this study.

As the final question of this part, the response times of the relational and the multidimensional environments to the new development needs that have been calculated within the experiment were presented to the participants and their opinions were asked to verify or refute whether the found durations are reasonable. Seventy-five percent of them agreed that the durations would be generalized to the real-life practices. They also added that the durations of the tasks, Modify ETL job design and Alter Cube Model would change according to the complexity of the scenario. Lastly, they stated that the OLAP Cube structures should not be changed frequently if they were modelled correctly.

The final part of the questionnaire was composed of the questions about the functional features of both OLAP and relational environments, which have been identified within the theoretical research (see Table 19). Some implications from the performance evaluation were also asked for generalization.

Table 19. Results from the Questionnaire on Functional Features.

20	Ad hoc query creation is easier and more flexible with OLAP tools when compared to the relational reporting.	0.82
21	Short query response times and capabilities like drilling-down, slicing & dicing of OLAP provide interactivity with the data for users while it is limited for the relational system.	1.73
22	Relational reporting tools are better than OLAP reporting tools at developing formatted static reports.	1.00
23	A good opportunity for using an OLAP Cube in comparison with relational reporting is simplicity to model complex environments.	0.73
24	OLAP is more appropriate than relational model for managerial reporting in banking.	1.27
25	The granular data in the relational model is also used to service unknown future needs for information while OLAP cube only meet existing needs of existing user groups.	1.18
26	Flexibility of OLAP systems can be considered as a limitation when compared to the relational model.	1.18
27	OLAP cube and data marts are not scalable; those are departmental and not enterprise solutions and only appropriate when there is a predictable usage.	1.00
28	OLAP Cubes are not proper tools for rapidly changing environments.	0.91
29	Total cost of ownership of OLAP for MI reporting is more than the relational system	0.18
30	Querying with MDX language is more difficult than querying with SQL because of the technical complexity of the MDX language.	0.91

Regarding the reporting purposes; the participants (eighty-two percent of them) agreed that it is more flexible to create ad-hoc reports through OLAP and this flexibility of the OLAP comes from the Microsoft Excel application’s ability to connect to most of the OLAP products. On the other hand, according to the general opinion (eighty-two percent) of the participants the development of structured reports with relational reporting tools was easier than OLAP reporting tools.

Seventy-three percent of the participants strongly agreed that the fast query response times and several features of OLAP, which promote the presentation of the information from several perspectives, provide interactivity with the data.

The respondents (ninety-one percent) agreed that OLAP is more appropriate than relational model for managerial reporting in banking. On the other hand, they stated that OLAP would be used to serve for only a specific purpose, which in the current case is the MI reporting, whereas it is easier to serve further needs through the relational DM since it also keeps the granule data. Keeping the granule data also within the Cube was suggested to solve this problem. However, this idea might not be feasible when working such kind of large data sets.

The experts agreed that the MDX is a more difficult language than SQL, since SQL is a standard language for relational DB querying while MDX is only a vendor specific multidimensional querying language. Other cube vendors have different cube querying languages such as Essbase-Maxl and different expertises have to be gained for each of the products.

Some of the experts refuted the proposition stating that the total cost of ownership of OLAP for MI reporting is greater than the relational system. They said regarding query performance issues of relational queries that, waiting duration for each set of queries would be much higher than OLAP, especially when trying to run them for the previous terms. This would make development flexibility cost of the OLAP worthy. Also, by using Cubes, some very complex reports would start to be taken, which have not been possible within the current system.

On the other hand, participants agreed (ninety-one percent) that OLAP Cubes are not proper tools for rapidly changing environments; however, one of the experts who have developed the current MI DM years ago, mentioned that the current system was not changed much within the years and stated that only the number of reports were increased. As a general opinion, participants stated that using OLAP is more appropriate than relational model for managerial reporting in banking since the several perspectives of data were needed to be reported in a very aggregated way.

CHAPTER 8

SUMMARY, CONCLUSIONS AND DISCUSSION

Research Questions

As stated in the beginning, main goal of the study was to demonstrate the opportunities and limitations of using OLAP Cubes for BI reporting in banking, which are considered to promote fast and multidimensional analysis of information on the business variables in this field.

The study was guided by a main research question that has been specified as: *What are the opportunities and limitations of using OLAP Cubes for BI reporting in banking?*

A number of secondary research questions have been formulated as well, which were answered in several chapters of the study. First six research questions, which compose the theoretical part of the thesis subject, were as follows:

RQ1. What constitute the components, layers and types of architecture of BI and DW infrastructure?

RQ2. What are the needs for BI and its utilization areas in banking?

RQ3. What are the basic differences between relational and dimensional models for database design in BI/DW process?

RQ4. What are the basic concepts, features, types and operations of OLAP and Cubes?

RQ5. How relational reporting is compared to the OLAP reporting in theory?

RQ6. In theory, what are defined as the benefits of using OLAP Cubes in banking?

These research questions were answered through the review of theoretical background and literature. The findings from this phase of the study can be summarized as follows:

The components of BI are, data collection, data integration (ETL), data warehousing, data analysis and presentation. *Raw data* are collected from several operational systems, administrative processes or from external sources. These data are transformed into meaningful information through *data integration*. *Integration layer* involves profiling to evaluate validity and reliability of data along with extracting, transformation, staging, cleansing, merging and loading processes. Then, data are *stored* in DM or in DW which are databases developed to organize and process integrated information on confirmed variables. *Data analysis* provides the evaluation and interpretation of current circumstances of business activities. In last stage, the information that has been previously transformed and analyzed are *displayed* to business users through several presentation techniques that enables them to identify, query and analyze business variables of the organization.

There is a conceptual debate in literature on definition of DW. One of the approaches refers DW as the core of the whole system where data are stored and accessed for analysis while a second believes that DW is a complete solution for analytical processing and decision making.

On the basis of these different approaches, data warehousing can be conducted in different ways such as an enterprise-wide DW (CIF) based on relational model or dimensional bus architecture representing the organization’s key business processes (BUS).

The primary difference between two techniques of data warehousing is the normalized data foundation. Normalization of data implies that the database design has caused the data to be broken down into a very low level of granularity. Another thing that separates these two approaches is the management of atomic data. According to first approach, atomic data will be stored within a normalized DW whereas the second approach suggests a method in which the atomic data should be placed within a dimensional structure.

Business benefits of BI are summarized in literature as lowering costs and expenses, reducing risks, increasing revenue, improving productivity, operational performance, profitability and customer satisfaction

The main areas that BI tools promote in banking are:

- Product/service profitability
- CRM
- ABC
- Customer profiling and segmentation
- Risk management
- Sales and marketing
- Fraud detection
- Tracking and identification of anti-money laundering
- Corporate performance management

Widely considered models for database design is dimensional and relational model. Normalization and indexing are most important concepts of relational model. In *dimensional database model* data are presented in dimensional structure.

A school of thought suggests that relational environment is shaped by the corporate or enterprise data model while dimensional model is shaped by the end-user requirements. According to this approach, granular data in the relational model is also used to service unknown future needs for information, not only to meet existing needs of existing user groups as in dimensional model.

Flexibility and performance issues are described as most important difference between the two models in literature. It is considered that relational model is highly flexible but is not optimized for performance while dimensional model is good at in performance but not at flexibility. On the other hand, many authors of second group consider that conducting the basic operations such as *complex data aggregation, comparison and consolidation* is difficult in relational method.

Some authors suggests that dimensional model and DM are not scalable; those are departmental and not enterprise solutions and only appropriate when there is a predictable usage pattern while the others refute these assumptions.

OLAP is commonly defined as a category of software technology that enables analyst and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user. FASMI is another definition that refers OLAP as *Fast Analysis of Shared Multidimensional Information* a critical category of information technology and a significant tool for high performance and multidimensional analysis of large scale business data.

The functional requirements for OLAP are:

- Rich dimensional structuring with hierarchical referencing
- Efficient specification of dimensions and dimensional calculations
- Separation of structure and representation
- Sufficient speed to support ad hoc analysis
- Multi-user support

OLAP achieves the multidimensional functionality by using a structure called Cube. Cube, as the core of OLAP, is a structure specially designed to retrieve, analyze and report data efficiently on the OLAP database platform. An OLAP Cube consists of numeric facts called measures, which are categorized by dimensions. The cube metadata is typically created from a star schema or snowflake schema of tables in a relational database. Measures are derived from the records in the fact tables and dimensions are derived from the dimension tables. Measures are the items to be counted, summarized and aggregated. Dimensions are the terms or variables, which are utilized to organize and summarize measures.

OLAP cube operations include slicing and dicing, drill-down analysis, rolling up and pivoting.

The major types of OLAP tools that differ with respect to their functionality and architecture are MOLAP, ROLAP and HOLAP. There is not an agreement between the authors on the most effective type of OLAP architectures. According to some authors, MOLAP systems, as the classic form of OLAP, are dedicated online analytical processing implementations not dependent on relational databases. These structures directly support the multidimensional view of data through a multidimensional storage engine. It's possible to implement front-end multidimensional queries on the storage layer through direct mapping with MOLAP. These models are capable to perform complex calculations; provide users the ability to quickly write back data into a data set and optimized for fast query performance and retrieval of summarized information. MOLAP systems are considered as less scalable due to the capability of handling only a limited amount of data, all needed information cannot be stored in MOLAP database. ROLAP databases provide a multidimensional front end that creates queries to process information in a relational format and the ability to transform two-dimensional relational data into multidimensional information. The main idea here is that it is better to read data from the DW directly, than to use another kind of storage for the cube. ROLAP architecture is more flexible since it can pre-calculate some of the aggregations, but leave others to be calculated on request. Advantages of ROLAP are better scalability, efficiency in managing both numeric and textual data. However, ROLAP applications display a slower performance as compared to MOLAP.

According to an approach, those who need to perform analysis on large volumes of data, to perform detailed what-if analysis based on multiple scenarios and up-to-the-minute data that is loaded on a near real-time basis should use MOLAP architecture.

HOLAP systems store larger quantities of detailed data in the relational tables while the aggregations are stored in the pre-calculated cubes. Some of the advantages of this system are better scalability, quick data processing and flexibility in accessing of data sources.

Databases, which uses relational model, can perform large amounts of small transactions, keeping the database available and the data consistent at all time. The *normalization* helps keeping the data consistent, but it also introduces a higher degree of complexity to the database, which causes huge databases to perform poorly when it comes to composite aggregation operations. In the context of business it is desirable to have historical data covering years of transactions, which results in a vast amount of database records to be analyzed. The performance issues will arise when processing analytical queries that require complex joining on such databases. Another issue with doing analysis with the relational model is that it requires complex queries, particularly composed for each request. Conducting most basic operations of analysis such as *comparisons* is also difficult in relational model. OLAP has been proposed to handle these issues.

The followings are considered as the differences between relational reporting and OLAP analysis in theory:

- OLAP provides interaction with data while relational reporting can provide limited interactivity.
- OLAP cubes are very good at providing very complex calculations whereas it is difficult in relational reporting.
- Relational reporting tools are better at providing report formatting than OLAP tools.
- OLAP provides very fast query performance when it is compared with relational model.
- Relational reporting is better in handling textual information than OLAP.
- OLAP may provide less maintenance since all aggregations are automatically provided within the cube. Relational reporting can be maintenance intensive if numerous "aggregation tables" are necessary.
- Ad hoc query creation is very easy with OLAP tools when it is compared with relational reporting.
- Conducting trend relationship between several terms of data is easier in OLAP

systems with respect to the relational ones.

OLAP cubes are considered as specifically developed tools for the queries in large databases with many relations involved, such as those in banks and optimized for short query response times. OLAP systems are utilized to discover trends and to analyze critical factors and to keep the banking managers informed about the business situation.

Since OLAP cubes are precalculated, which results in fast query performance that is critical for executive use, OLAP is well suited to handle complicated organizational roll-ups, as well as complex calculations, including inter-row manipulations. With these features, OLAP naturally fits for banking reporting.

The theoretical concepts were investigated up here. Through the following research questions, the empirical part of the study was carried out:

RQ7. How can an OLAP Cube be utilized to improve BI reporting in banking?

RQ8. Which opportunities and limitations exist in utilization of OLAP Cube technology for BI reporting in banking?

RQ9. What are quantitative and qualitative differences between OLAP and relational reporting in banking?

This group of questions was answered experimentally; an OLAP Cube was designed and developed using relevant data of the Bank to show the quantitative opportunities and constraints of an OLAP Cube by testing its performance through several queries in comparison with relational MI reporting system of the Bank. Additionally, functional properties of OLAP queries were evaluated within this phase of the study.

First test subject was the comparison between the run durations of the reports on relational and multidimensional environments; on a set of eight reports chosen from the current reporting system, which differed in complexity (query cost), row count of the queried tables, row count of the result set, number of queried dimensions, number of tables joined and number of queried terms. The results obtained from these tests show that:

- OLAP has a big advantage in terms of report processing duration when compared to the relational system.
- Using many joins doesn't affect the query cost too much if the total number of rows in the source tables were small.
- There is a positive relationship between the queried rows and the data-read part of the query cost and run duration.
- Aggregating low or high number of rows under the same number of dimensions does not affect the aggregation query cost and duration.
- The number of rows returned at the result set does not have much effect to the query cost and duration.
- There is a limitation at the relational side; a query can compose aggregations up to a maximum level and for the further needs, it does not work at all.
- Increasing the number of terms queried and trying to conduct a trend relationship between several terms increases the cost and durations dramatically.
- The number of dimensions queried increases the costs by multiplying the amount of aggregations that have to take place during the grouping operations.
- Above characteristics are essential for the MI reports and some of them become critical weaknesses of the relational environment while reporting such kind of information when compared to the OLAP Cube.

At the second phase of the tests, the additional requirements of having an OLAP Cube were evaluated in the context of updating the source DM, total amount of storage necessary to maintain the Cube, processing scenarios of the Cube and response to new development needs. The results from this phase show that:

- The total amount of storage necessary to maintain the Cube was almost equal to the size of the DM itself. It seems that the number of combinations of the dimensions and grouping the large amount of data in the fact tables around those dimensions enlarge the size of the Cube since sparsity was not an issue for the current implementation.
- As the Cube was developed with the MOLAP functionality, processing it was one of the most costly operations due to the long process durations and manual interference needs during monthly and full processes, which could not be automated because of the size and structure of the Cube.
- *Full Process of the whole Cube is necessary when* there is a change in the metadata and/or the source data or in the formulation of a calculated variable of the Cube.
- A possible change request in the real life environment most probably necessitates a model change at the Cube, which requires the Cube to be full processed.
- Without including the process duration of the Cube, the total average time to implement a change in OLAP is very close to that in relational environment.
- Due to the full process requirement, it can be said that OLAP model is less flexible than the relational model.
- The break even analysis showed that full processing requirement of the Cube was not a big disadvantage when the changes need to be applied to the reports because it do not need to process all the partitions. Reflecting changes to multiple terms of

several reports could be done more rapidly within OLAP system than the relational one.

- Scalability was a limitation for both OLAP and relational systems while dealing with such large data sets.

Another research question was:

RQ10. What is the common opinion among experts and users on using OLAP Cubes in banking?

This question was answered through a questionnaire. The actual purpose of using this method was to investigate the opinions of the experts of this kind of applications in banking sector in order to verify or refute the findings of the experiment and examine qualitative/functional aspects of using OLAP Cubes as well.

A great majority of the participants confirmed that number of joined tables, row count of the queried tables, row count of the result sets and number of queried terms were correct parameters to determine the complexity level (cost) of a query, which have been chosen for performance evaluation within the experiment.

A great majority of the respondents agreed that the OLAP systems have much greater reporting performance compared to the relational systems.

According to most respondents, aggregating, comparing and consolidating data within the relational system is harder than conducting those operations in an OLAP system. A big proportion of them supported the idea considering that OLAP also performs better in conducting trend analysis.

Common opinion among the participants was that the maintenance of an OLAP system is harder than the relational system and necessary effort to update the Cube is greater than relational system because of its long process durations. This disadvantage would prevent OLAP from being up to date with the DW.

All of the respondents confirmed that, for all OLAP tools any change in the source model or the major changes at the source data necessitate full processing of the Cube as determined during the experiment.

The participants confirmed that during the implementation of a change request to both systems, OLAP takes a longer time because of its full process need. On the other hand, except the process duration of the Cube, the response times of both systems to the development needs are very close to each other.

The response times of the relational and the multidimensional environments to the new development needs that have been calculated within the experiment were also confirmed by most of the respondents.

The great majority of the participants agreed that it is more flexible to create ad-hoc reports through OLAP. On the other hand, according to the common opinion, development of structured reports through relational reporting tools was easier than OLAP reporting tools.

Most of the respondents stated that fast query response times and several features of OLAP, which promote the presentation of the information from several perspectives, provide interactivity with the data.

The respondents (ninety-one percent) agreed that OLAP is more appropriate than relational model for managerial reporting in banking. On the other hand, it is easier to serve further needs through the relational DM.

The experts agreed that the MDX is a more difficult language than SQL, since SQL is a standard language for relational DB querying while MDX is only a vendor specific multidimensional querying language. Other cube vendors have different cube querying languages such as Essbase-Maxl and different expertises have to be gained for each of the products

Some of the experts refuted the proposition stating that the total cost of ownership of OLAP for MI reporting is greater than the relational system. They said that modification and processing costs of OLAP become worthy when the report durations in the relational side was considered. Waiting duration for each set of queries would be much higher than OLAP, especially when trying to run them for the previous terms. Also, they added that, by using Cubes, some very complex reports would start to be taken, which have not been possible within the current system.

A great majority of the participants agreed that OLAP Cubes are not proper tools for rapidly changing environments; however, according to common opinion among them, OLAP is more appropriate than relational model for managerial reporting in banking since the several perspectives of data were needed to be reported in a very aggregated way.

The last research question was formulated to compare the theory with the practice. It was:

RQ11. To what extent do the opportunities and limitations of OLAP utilization in banking differ from theory?

To answer this question, the results obtained through the experimental Cube design and questionnaires were compared to theoretical assumptions.

As shown through review of theory, a group of authors suggest that relational environment is shaped by the corporate or enterprise data model while dimensional model is shaped by the end-user requirements. According to this approach, granular data in the relational model is also used to service unknown future needs for information, not only to meet existing needs of existing user groups as in dimensional model. The results of this study showed that the theoretical assumptions were true as dimensional model specializes in particular points and serves only a specific purpose, however, reporting flexibility of the dimensional model is much better, thus, it should be well decided which model must be used in accordance with the purpose. In theory, some authors suggests that OLAP model and DMs are not scalable; those are departmental and not enterprise solutions and only appropriate when there is a predictable usage pattern while the others refute these assumptions.

Flexibility and performance issues are described as most important differences between the two models in literature. It is considered that relational model is highly flexible but is not optimized for performance while dimensional model is good at in performance but not at flexibility. On the other hand, many authors of second group consider that conducting the basic operations such as *complex data aggregation, comparison and consolidation* is difficult in relational method. Experimental evidences provided by this study demonstrate that in order to operate these kinds of applications, the relational model could not provide sufficient performance. A relational model is a necessity to store the granule data and to serve the future needs but dimensional DMs should be built in order to develop BI applications effectively. Both architectures should be used together within a BI infrastructure.

Regarding functional features of OLAP, sufficient speed to support ad hoc analysis, unrestricted cross dimensional operations, intuitive data manipulation and unlimited dimensions and aggregation levels have been specified in theory as the opportunities of OLAP reporting. The findings of this study showed that OLAP is actually appropriate for ad-hoc reporting and provides interactivity with the data through the mentioned features.

There was not an agreement between the authors on the most effective type of OLAP architectures, MOLAP, ROLAP, HOLAP. MOLAP by nature had a greater performance than ROLAP since it precalculates all the results and store them in the Cube. MOLAP method was also chosen for this study since the main goal was to investigate whether OLAP improves the reporting performance. On the other hand, ROLAP had advantages like flexibility, scalability and easy maintenance since it doesn't require processing the Cube beforehand. Opinions of the experts brought a new perspective to the OLAP concept by indicating that recently developed DW appliances may provide the performance of OLAP by using relational systems. By having such a high-end technology, MOLAP performance may be achieved when ROLAP model is used.

In literature, OLAP cubes are considered as specifically developed tools for the queries in large databases with many relations involved, such as those in banks and optimized for short query response times. OLAP systems are utilized to discover trends and to analyze critical factors and also well suited to handle complicated organizational roll-ups, as well as complex calculations. So, OLAP naturally fits for banking reporting. This study verified this statements, however, it has been seen that handling the new development needs in rapidly changing environments can be painful when using Cubes.

So, OLAP systems should be preferred to relational ones especially for the environments, which have static structures and reporting needs.

Contributions of the Study

It can be said that, this study provides meaningful insight and valuable information on the opportunities and limitations of using OLAP cubes for BI reporting in banking, since the proper methods such as clear conceptualization of the subject matter, triangulation; using multiple sources of evidence; verification of the results by using different methods were taken into consideration to support sufficient reliability, validity and generalizability of the results.

In spite of some limitations of the study, such as quite broad objectives of research; coverage of almost whole BI reporting research area, conducting an experiment on a very large scale of data in a short period of time and scarce literature to utilize especially for the experiment, a set of results that can be considered as reliable information were obtained.

As a theoretical contribution, this thesis extends the knowledge base that currently exists in that field; clarifies the framework for BI, OLAP and related concepts, which have been confused in literature; provides comprehensive information for both theory and practice.

As mentioned previously and shown through the review of theory and literature, very few studies have particularly examined the impacts of BI implementation in banking. Even, the number of the studies examining OLAP implementations from technical perspective on such large-scale data in any business area is limited. None of them has investigated the role of OLAP cube utilization in managerial reporting for banking in technical manner. Academic literature analyzing BI and OLAP is very limited. Therefore, this study provides a useful source of information for academicians working in the field of BI; it hopefully will contribute to future studies and encourage the academicians for further investigation of OLAP cube utilization in banking and/or on such large-scale data.

Meanwhile, the concept of OLAP Cube is relatively new to the Turkish banking industry. Empirical findings of this thesis can be utilized by developers and users of BI applications. Even though the scope of the thesis is limited to technical perspective, this study is also significant for banking institutions in respect to business operations as it explores the benefits and constraints of using such technology; it may contribute to business productivity by providing managers with several findings on use of these technologies in decision-making.

Directions for Further Research

Although the results of the study can be considered useful and reliable, not every possible opportunity and limitation of using OLAP Cubes for BI reporting in banking was answered due to the broad scope and extent of the study. Additionally, a number of new questions have been identified in some areas and several issues for further research have arisen while researching the topic. Some of these issues are provided below.

The results from this study can reasonably be expected to apply to future researches on such large scale data. On the other hand it is unclear whether the opportunities and limitations of using OLAP Cubes determined within this study are valid also for small data sets. Another research can be conducted to investigate this subject matter.

It was shown that, most limitations of the Cube detected in the study, have been caused due to using MOLAP model in order to achieve better query performance. On the other hand, the experts whose opinions were gathered through the questionnaire stated that, recently developed DW appliances have extremely high query performance levels and can reach the same performance with MOLAP cubes by using ROLAP models. Therefore, a further investigation can be conducted to explore whether the same system would be implemented by using these appliances. Thus, the following issues, which could not be detected in this study, can be examined by using ROLAP:

- Ability of conducting queries involving customer dimension,
- Ability of operating Cube without any manual interference,
- Flexibility and scalability features, which were considered in theory as the advantages of ROLAP.
- Presentation of up to date data with DW.

A disadvantage caused by the process requirements of the Cube was determined during the experiment, which would prevent it from being up to date with the DW. This limitation would be eliminated by building the Cube directly on the source transactional systems rather than the DW. On the other hand, keeping the granule data within the Cube was suggested by the experts during the study in order that OLAP Cube can serve further needs as well, besides some specific purposes. Both of these issues might not be feasible when working with such kind of large data sets and need to be investigated. Also within a further research, different optimization methods can be applied to reduce the size, process duration and manual interaction requirements of the Cube.

It was seen in this study that, since OLAP reports do not need to run beforehand, all the reports become available for all the previous terms when the Cube is full processed. This new perspective might invert the weakness of OLAP into an advantage when compared to the relational environment. Business requirements would be changed with such an opportunity and trends of the new measures would be observed easily. A further research may also be conducted to investigate the possible advantages of this technology from a business viewpoint.

To conclude, the need to continue the studies on the methods toward more efficient reporting from both academic and business point of view is obvious due to the significant role of managerial reporting in banking in rapidly changing business environment.

APPENDICES

Appendix A: Dimensional Modelling Myths (by Kimball & Ross)

Despite the general acceptance of dimensional modeling, some misperceptions continue to be disseminated in the industry. We refer to these misconceptions as *dimensional modeling myths*.

Myth 1. *Dimensional models and data marts are for summary data only*. This first myth is the root cause of many ill-designed dimensional models. Because we can't possibly predict all the questions asked by business users, we need to provide them with queryable access to the most detailed data so that they can roll it up based on the business question at hand. Data at the lowest level of detail is practically impervious to surprises or changes. Our data marts also will include commonly requested summarized data in dimensional schemas. This summary data should complement the granular detail solely to provide improved performance for common queries, but not attempt to serve as a replacement for the details. A related corollary to this first myth is that only a limited amount of historical data should be stored in dimensional structures. There is nothing about a dimensional model that prohibits the storage of substantial history. The amount of history available in data marts must be driven by the business's requirements.

Myth 2. *Dimensional models and data marts are departmental, not enterprise, solutions*.
Rather than drawing boundaries based on organizational departments, we maintain that data marts should be organized around business processes, such as orders, invoices, and service calls. Multiple business functions often want to analyze the same metrics resulting from a single business process. We strive to avoid duplicating the core measurements in multiple databases around the organization. Supporters of the normalized data warehouse approach sometimes draw spiderweb diagrams with multiple extracts from the same source feeding into multiple data marts. The illustration supposedly depicts the perils of proceeding without a normalized data warehouse to feed the data marts. These supporters caution about increased costs and potential inconsistencies as changes in the source system of record would need to be rippled to each mart's ETL process. This argument falls apart because no one advocates multiple extracts from the same source. The spiderweb diagrams fail to appreciate that the data marts are process-centric, not department-centric, and that the data is extracted once from the operational source and presented in a single place. Clearly, the operational system support folks would frown on the multiple extract approach. So do we.

Myth 3. *Dimensional models and data marts are not scalable*. Modern fact tables have many billions of rows in them. The dimensional models within our data marts are extremely scalable. Relational DBMS vendors have embraced data warehousing and incorporated numerous capabilities into their products to optimize the scalability and performance of dimensional models. A corollary to myth 3 is that dimensional models are only appropriate for retail or sales data. This notion is rooted in the historical origins of dimensional modeling but not in its current-day reality. Dimensional modeling has been applied to virtually every industry, including banking, insurance, brokerage, telephone, newspaper, oil and gas, government, manufacturing, travel, gaming, health care, education, and many more. In this book we use the retail industry to illustrate several early concepts mainly because it is an industry to which we have all been exposed; however, these concepts are extremely transferable to other businesses.

Myth 4. *Dimensional models and data marts are only appropriate when there is a predictable usage pattern*. A related corollary is that dimensional models aren't responsive to changing business needs. On the contrary, because of their symmetry, the dimensional structures in our data marts are extremely flexible and adaptive to change. The secret to query flexibility is building the fact tables at the most granular level. In our opinion, the source of myth 4 is the designer struggling with fact tables that have been prematurely aggregated based on the designer's unfortunate belief in myth 1 regarding summary data. Dimensional models that only deliver summary data are bound to be problematic. Users run into analytic brick walls when they try to drill down into details not available in the summary tables. Developers also run into brick walls because they can't easily accommodate new dimensions, attributes, or facts with these prematurely summarized tables. The correct starting point for your dimensional models is to express data at the lowest detail possible for maximum flexibility and extensibility.

Myth 5. *Dimensional models and data marts can’t be integrated and therefore lead to stovepipe solutions*. Dimensional models and data marts most certainly can be integrated if they conform to the data warehouse bus architecture. Presentation area databases that don’t adhere to the data warehouse bus architecture will lead to standalone solutions. You can’t hold dimensional modeling responsible for the failure of some organizations to embrace one of its fundamental tenets.

Appendix B: Twelve Rules to Evaluate OLAP Systems (by Codd, Codd & Salley)

Multidimensional Conceptual View
A user-analyst's view of the enterprise's universe is multidimensional in nature. Accordingly, the useranalyst's conceptual view of OLAP models should be multidimensional in nature. This multidimensional conceptual schema or user view facilitates model design and analysis, as well as inter and intra dimensional calculations through a more intuitive analytical model. Accordingly user-analysts are able to manipulate such multidimensional data models more easily and intuitively than is the case with single dimensional models. For instance, the need to "slice and dice," or pivot and rotate consolidation paths within a model is common. Multidimensional models make these manipulations easily, whereas achieving a like result with older approaches requires significantly more time and effort.

Transparency
Whether OLAP is or is not part of the user's customary front-end (e. g., spreadsheet or graphics package) product, that fact should be transparent to the user. If OLAP is provided within the context of a client server architecture, then this fact should be transparent to the user-analyst as well. OLAP should be provided within the context of a true open systems architecture, allowing the analytical tool to be embedded anywhere the user-analyst desires, without adversely impacting the functionality of the host tool. Transparency is crucial to preserving the user's existing productivity and proficiency with the customary front-end, providing the appropriate level of function, and assuring that needless complexity is in no way introduced or otherwise increased. Additionally, it should be transparent to the user as to whether or not the enterprise data input to the OLAP tool comes from a homogenous or heterogeneous database environment.

Accessibility
The OLAP user-analyst must be able to perform analysis based upon a common conceptual schema composed of enterprise data in relational DBMS, as well as data under control of the old legacy DBMS, access methods, and other non-relational data stores at the same time as the basis of a common analytical model. That is to say that the OLAP tool must map its own logical schema to heterogeneous physical data stores, access the data, and perform any conversions necessary to present a single, coherent and consistent user view. Moreover, the tool and not the end-user analyst must be concerned about where or from which type of systems the physical data is actually coming. The OLAP system should access only the data actually required to perform the indicated analysis and not take the common "kitchen sink" approach which brings in unnecessary input.

Consistent Reporting Performance
As the number of dimensions or the size of the database increases, the OLAP user-analyst should not perceive any significant degradation in reporting performance. Consistent reporting performance is critical to maintaining the ease-of-use and lack of complexity required in bringing OLAP to the end-user. If the user-analyst were able to perceive any significant difference in reporting performance relating to the number of dimensions requested, there would very likely be compensating strategies developed, such as asking for information to be presented in ways other than those really desired. Spending one's time in devising ways of circumventing the system in order to compensate for its inadequacies is not what enduser products are about.

Client-Server Architecture
Most data currently requiring on-line analytical processing is stored on mainframe systems and accessed via personal computers. It is therefore mandatory that the OLAP products be capable of operating in a client-server environment. To this end, it is imperative that the server component of OLAP tools be sufficiently intelligent such that various clients can be attached with minimum effort and integration programming. The intelligent server must be capable of performing the mapping and consolidation between disparate logical and physical enterprise database schema necessary to effect transparency and to build a common conceptual, logical and physical schema.

Generic Dimensionality
Every data dimension must be equivalent in both its structure and operational capabilities. Additional

operational capabilities may be granted to selected dimensions, but since dimensions are symmetric, a given additional function may be granted to any dimension. The basic data structure, formulae, and reporting formats should not be biased toward any one data dimension.

Dynamic Sparse Matrix Handling
The OLAP tools' physical schema must adapt fully to the specific analytical model being created to provide optimal sparse matrix handling. For any given sparse matrix, there exists one and only one optimum physical schema. This optimal schema provides both maximum memory efficiency and matrix operability unless of course, the entire data set can be cached in memory. The OLAP tool's basic physical data unit must be configurable to any subset of the available dimensions, in any order, for practical operations within large analytical models. The physical access methods must also be dynamically changeable and should contain different types of mechanisms such as:
1. direct calculation;
2. B-trees and derivatives,
3. hashing;
4. the ability to combine these techniques where advantageous.

Sparseness (missing cells as a percentage of possible cells) is but one of the characteristics of data distribution. The inability to adjust (morph) to the data set's data distribution can make fast, efficient operation unobtainable. If the OLAP tool cannot adjust according to the distribution of values of the data to be analyzed, models which appear to be practical, based upon the number of consolidation paths and dimensions, or the size of the enterprise source data, may be needlessly large and/or hopelessly slow in actuality. Access speed should be consistent regardless of the order of cell access and should remain fairly constant across models containing different numbers of data dimensions or varying sizes of data sets.

For example, given a set of input data from the enterprise database which is perfectly dense (every possible input combination contains a value, no nulls), it is possible to predict the size of the resulting data set after consolidation across all modeled data dimensions.

For example, in a particular five-dimensional analytical model, let us suppose that the physical schema size after model consolidation is two-and-one-half times the size of the input data from the enterprise database.

However, if the enterprise data is sparse, and has certain distribution characteristics, then the resulting physical schema might be one-hundred times the size of the enterprise data input. But, given the same size data set, and the same degree of sparseness, but with different data distribution, the size of the resulting physical schema might be only two-and-one-half times the size of the enterprise data input as in the case of the perfectly dense example. Or, we could experience anything in between these two extremes. "Eyeballing" the data in an attempt to form an educated guess is as hopeless as using conventional statistical analysis tools to obtain crosstabs of the data.

Because conventional statistical analysis tools always compare only one dimension against one other dimension, without regard for the other, perhaps numerous, data dimensions, they are unsuitable to multidimensional data analysis. Even if such tools could compare all dimensions at once (which they can't), the resulting crosstab would be the size of the product of all the data dimensions, which would be the maximum size of the physical schema itself.

By adapting its physical data schema to the specific analytical model, OLAP tools can empower user-analysts to easily perform types of analysis which previously have been avoided because of their perceived complexity. The extreme unpredictability and volatility in the behavior of multidimensional data models precludes the successful use of tools which rely upon a static physical schema and whose basic unit of data storage has fixed dimensionality (e.g., cell, record or two-dimensional sheet). A fixed, physical schema which is optimal for one analytical model, will typically be impractical for most others. Rather than basing a physical schema upon cells, records, two dimensional sheets, or some other similar structure, OLAP tools must dynamically adapt the model's physical schema to the indicated dimensionality and especially to the data distribution of each specific model.

Multi-User Support
Oftentimes, several user-analyst's have a requirement to work concurrently with either the same analytical model or to create different models from the same enterprise data. To be regarded as strategic, OLAP tools must provide concurrent access (retrieval and update), integrity, and security.

Unrestricted Cross-Dimensional Operations
The various roll-up levels within consolidation paths, due to their inherent hierarchical nature, represent in outline form, the majority of 1:1, 1:M, and dependent relationships in an OLAP model or application. Accordingly, the tool itself should infer the associated calculations and not require the user-analyst to explicitly define these inherent calculations. Calculations not resulting from these inherent relationships require the definition of various formulae according to some language which of course must be computationally complete.

Such a language must allow calculation and data manipulation across any number of data dimensions and must not restrict or inhibit any relationship between data cells regardless of the number of common data attributes each cell contains.

For example, consider the difference between a single dimensional calculation and a cross-dimensional calculation. The single dimensional calculation: Contribution = Revenue -Variable Cost defines a relationship between attributes in only one data dimension, which we shall call *D_ACCOUNTS*. Upon calculation, what occurs is that the relationship is calculated for all cells of all data dimensions in the data model which possess the attribute *Contribution*.
A cross-dimensional relationship and the associated calculations provide additional challenges. For example, given the following simple five-dimensional outline:

D_Accounts	*D_Corporate*	*D_Fiscal Year*	*D_Products*	*D_Scenario*
Sales	United Kingdom	Quarter1	Audio	Budgeted
Overhead	London	January	Video	Actual
Interest Rate	York	February	et cetera	Variance
et cetera	France	March		et cetera
	Paris	Quarter2		
	Cannes	April		
	et cetera	May		
		June		
		et cetera		

Sample Five-Dimensional Outline Structure
The formula to allocate corporate overhead to parts of the organization such as local offices (Paris, Cannes, et cetera) based upon their respective contributions to overall company sales might appear thus:
Overhead equals the percentage of total sales represented by the sales of each individual local office multiplied by total corporate overhead

1 "D_" is used to indicate that this top most aggregation level is the dimension.

Here is another example of necessary cross-dimensional calculations. Suppose that the user-analyst desires to specify that for all French cities, the variable *Interest Rate* which is used in subsequent calculations, should be set to the value of the *BUDGETED MARCH INTEREST RATE* for the city of Paris for all months, across all data dimensions. Had the user-analyst not specified the city, month and scenario, the attributes would alter and stay consistent with the month attributes of the data cell being calculated when the analytical model is animated. The described calculation could be expressed as:

If the value within the designated cell appears within the consolidation path D_Corporate, beneath the consolidation level France, then the global interest rate becomes the value of the interest rate for the month of March which is budgeted for the city of Paris

Intuitive Data Manipulation

Consolidation path re-orientation, drilling down across columns or rows, zooming out, and other manipulation inherent in the consolidation path outlines should be accomplished via direct action upon the cells of the analytical model, and should neither require the use of a menu nor multiple trips across the user interface. The user-analyst's view of the dimensions defined in the analytical model should contain all information necessary to effect these inherent actions.

Flexible Reporting

Analysis and presentation of data is simpler when rows, columns, and cells of data which are to be visually compared are arranged in proximity or by some logical grouping occurring naturally in the enterprise. Reporting must be capable of presenting data to be synthesized, or information resulting from animation of the data model according to any possible orientation. This means that the rows, columns, or page headings must each be capable of containing/displaying from 0 to N dimensions each, where N is the number of dimensions in the entire analytical model.

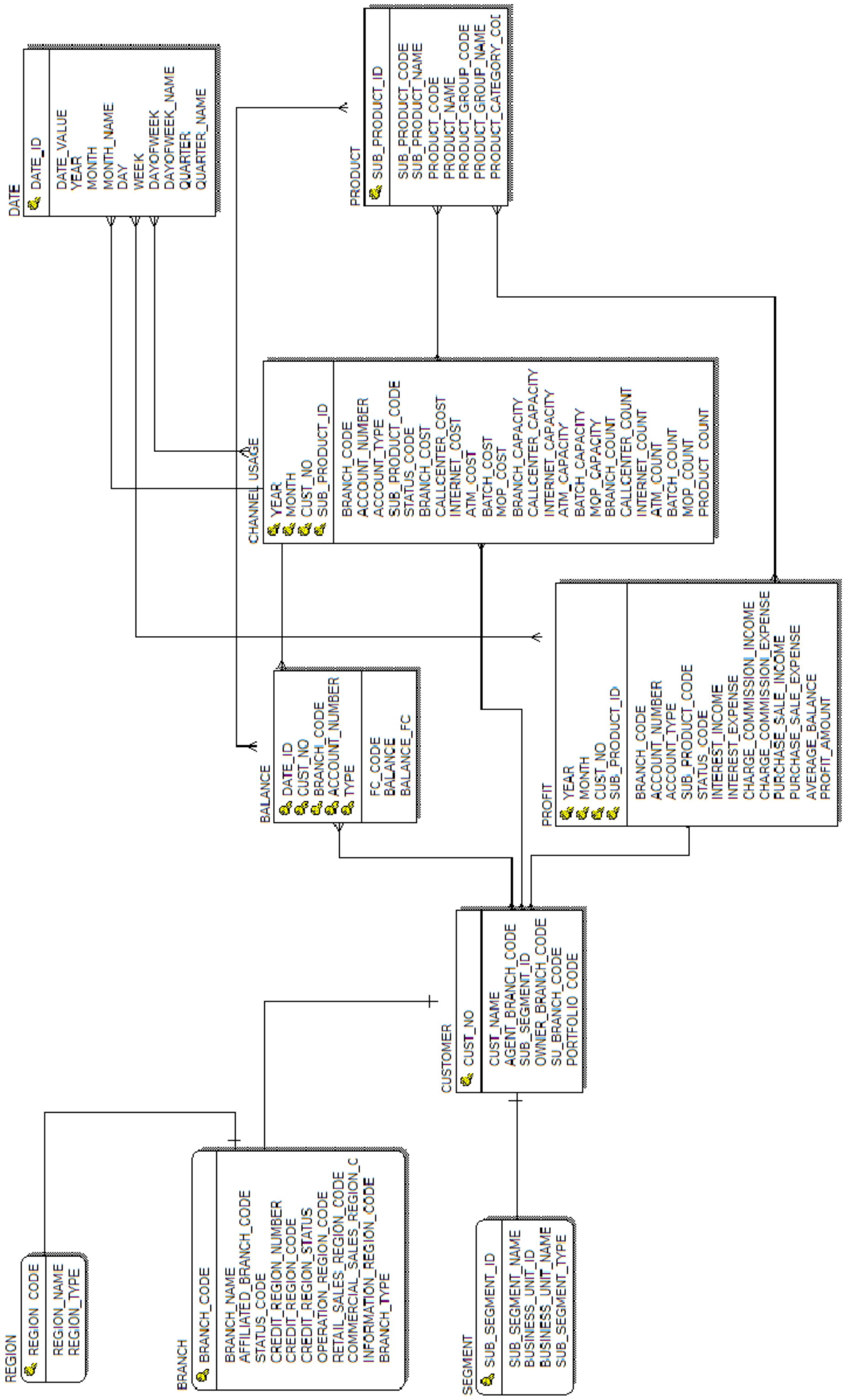
Additionally, each dimension contained/displayed in one of these rows, columns, or page headings must itself be capable of containing/displaying any subset of the members, in any order, and provide a means of showing the inter-consolidation path relationships between the members of that subset such as indentation.

Unlimited Dimensions and Aggregation Levels

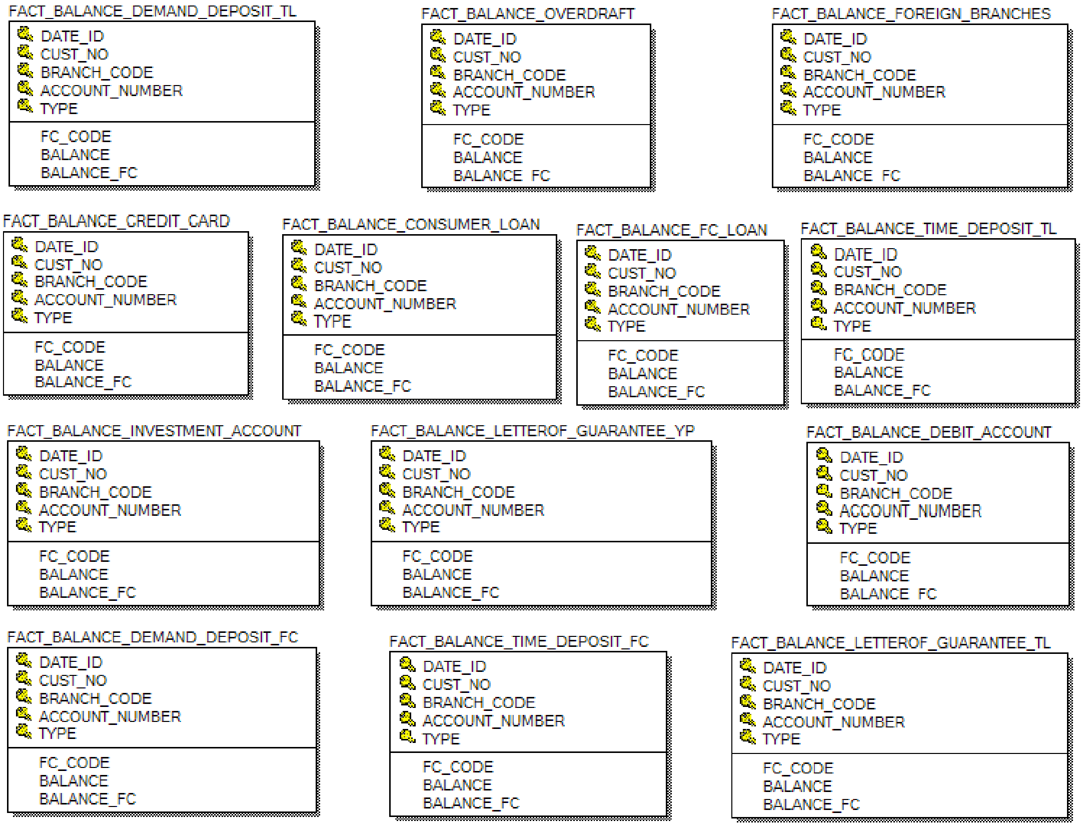
Research into the number of dimensions possibly required by analytical models indicates that as many as nineteen concurrent data dimensions (this was an actuarial model) may be needed. Thus the strong recommendation that any serious OLAP tool should be able to accommodate at least fifteen and preferably twenty data dimensions within a common analytical model.

Furthermore, each of these generic dimensions must allow an essentially unlimited number of user-analyst defined aggregation levels within any given consolidation path.

Appendix C: Final Data Mart Logical Model



Appendix D: Balance Tables of Each Product at the Source DM



FACT_BALANCE_DEMAND_DEPOSIT_TL
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

FACT_BALANCE_OVERDRAFT
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE FC

FACT_BALANCE_FOREIGN_BRANCHES
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE FC

FACT_BALANCE_CREDIT_CARD
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

FACT_BALANCE_CONSUMER_LOAN
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

FACT_BALANCE_FC_LOAN
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

FACT_BALANCE_TIME_DEPOSIT_TL
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

FACT_BALANCE_INVESTMENT_ACCOUNT
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

FACT_BALANCE_LETTEROF_GUARANTEE_YP
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

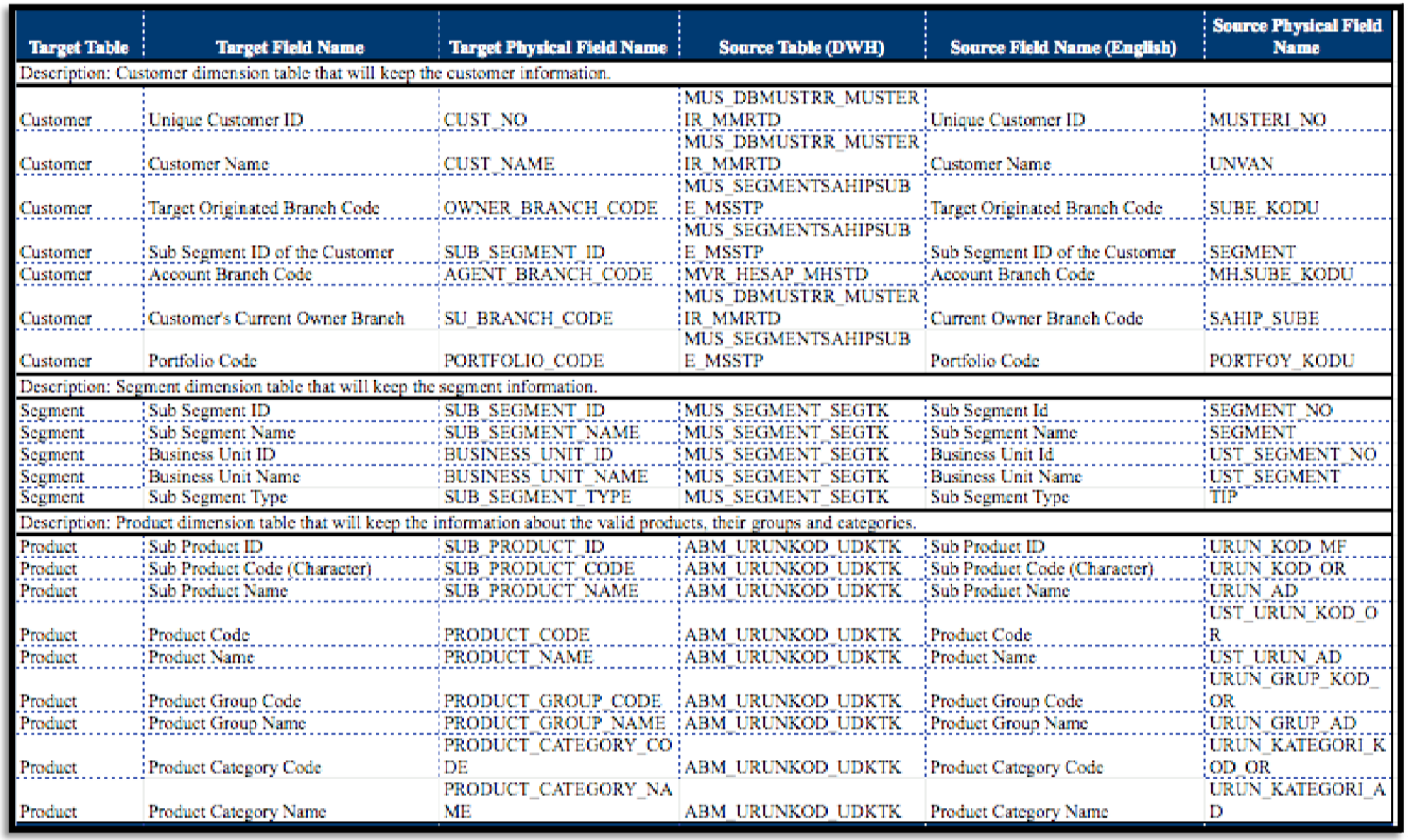
FACT_BALANCE_DEBIT_ACCOUNT
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE FC

FACT_BALANCE_DEMAND_DEPOSIT_FC
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

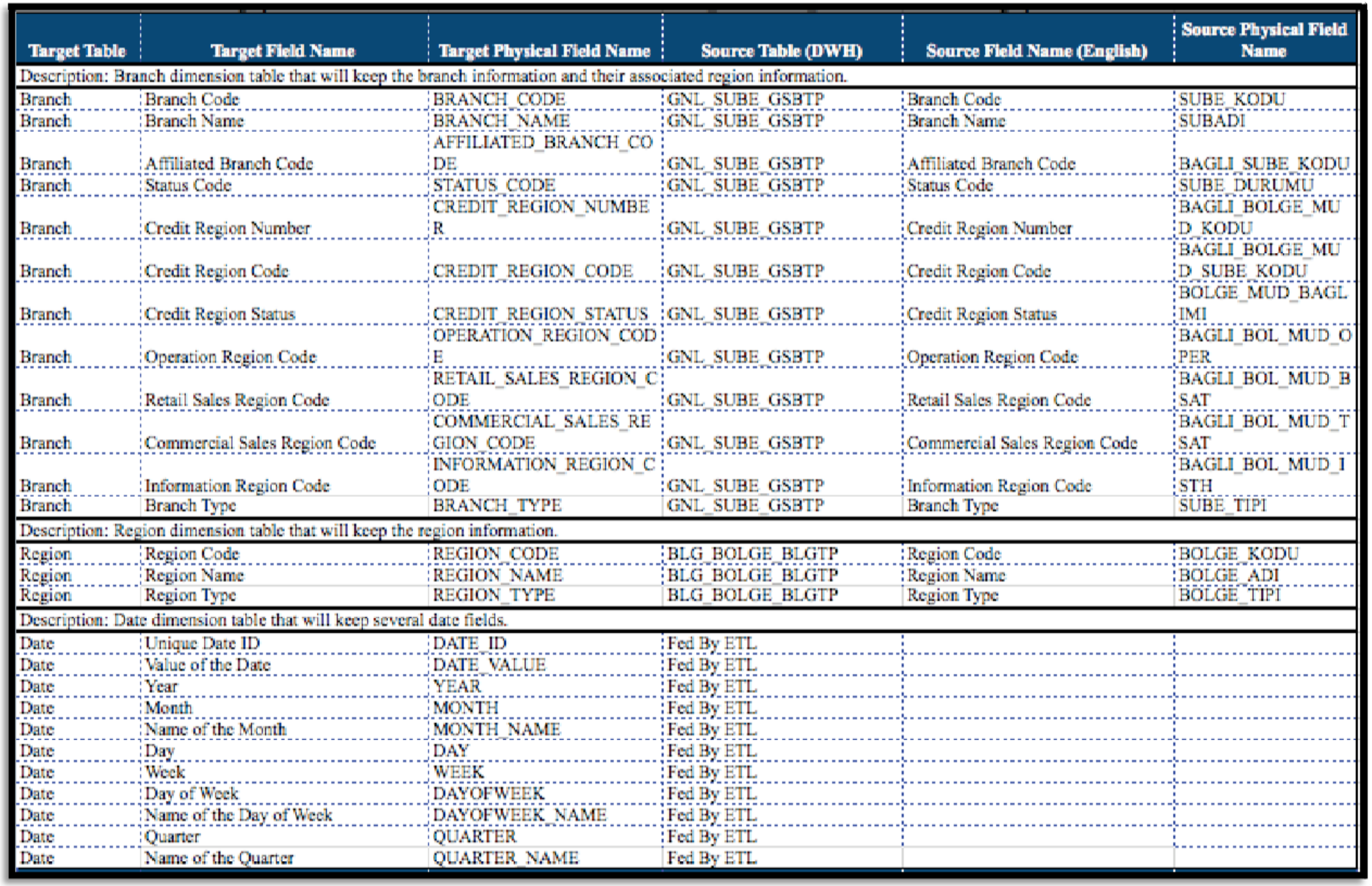
FACT_BALANCE_TIME_DEPOSIT_FC
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

FACT_BALANCE_LETTEROF_GUARANTEE_TL
DATE_ID
CUST_NO
BRANCH_CODE
ACCOUNT_NUMBER
TYPE
FC_CODE
BALANCE
BALANCE_FC

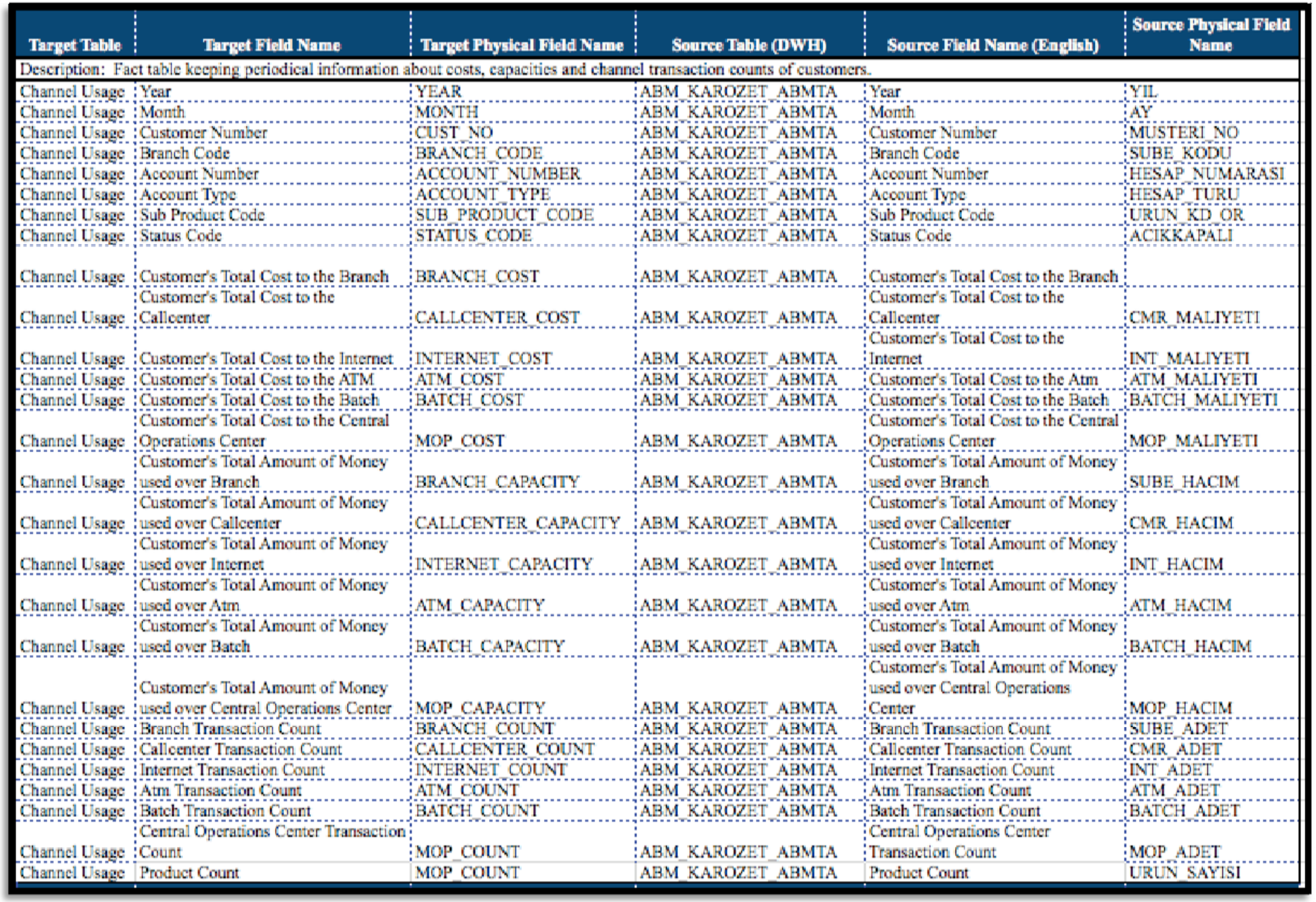
Appendix E: Data Warehouse to Data Mart ETL Design Map



Target Table	Target Field Name	Target Physical Field Name	Source Table (DWH)	Source Field Name (English)	Source Physical Field Name
Description: Customer dimension table that will keep the customer information.					
Customer	Unique Customer ID	CUST_NO	MUS_DBMUSTRR_MUSTERIR_MMRTD	Unique Customer ID	MUSTERI_NO
Customer	Customer Name	CUST_NAME	MUS_DBMUSTRR_MUSTERIR_MMRTD	Customer Name	UNVAN
Customer	Target Originated Branch Code	OWNER_BRANCH_CODE	MUS_SEGMENTSAHIPSUBE_MSSTP	Target Originated Branch Code	SUBE_KODU
Customer	Sub Segment ID of the Customer	SUB_SEGMENT_ID	MUS_SEGMENTSAHIPSUBE_MSSTP	Sub Segment ID of the Customer	SEGMENT
Customer	Account Branch Code	AGENT_BRANCH_CODE	MVR_HESAP_MHSTD	Account Branch Code	MH.SUBE_KODU
Customer	Customer's Current Owner Branch	SU_BRANCH_CODE	MUS_DBMUSTRR_MUSTERIR_MMRTD	Current Owner Branch Code	SAHIP_SUBE
Customer	Portfolio Code	PORTFOLIO_CODE	MUS_SEGMENTSAHIPSUBE_MSSTP	Portfolio Code	PORTFOY_KODU
Description: Segment dimension table that will keep the segment information.					
Segment	Sub Segment ID	SUB_SEGMENT_ID	MUS_SEGMENT_SEGTK	Sub Segment Id	SEGMENT_NO
Segment	Sub Segment Name	SUB_SEGMENT_NAME	MUS_SEGMENT_SEGTK	Sub Segment Name	SEGMENT
Segment	Business Unit ID	BUSINESS_UNIT_ID	MUS_SEGMENT_SEGTK	Business Unit Id	UST_SEGMENT_NO
Segment	Business Unit Name	BUSINESS_UNIT_NAME	MUS_SEGMENT_SEGTK	Business Unit Name	UST_SEGMENT
Segment	Sub Segment Type	SUB_SEGMENT_TYPE	MUS_SEGMENT_SEGTK	Sub Segment Type	TIP
Description: Product dimension table that will keep the information about the valid products, their groups and categories.					
Product	Sub Product ID	SUB_PRODUCT_ID	ABM_URUNKOD_UDKTK	Sub Product ID	URUN_KOD_MF
Product	Sub Product Code (Character)	SUB_PRODUCT_CODE	ABM_URUNKOD_UDKTK	Sub Product Code (Character)	URUN_KOD_OR
Product	Sub Product Name	SUB_PRODUCT_NAME	ABM_URUNKOD_UDKTK	Sub Product Name	URUN_AD
Product	Product Code	PRODUCT_CODE	ABM_URUNKOD_UDKTK	Product Code	UST_URUN_KOD_OR
Product	Product Name	PRODUCT_NAME	ABM_URUNKOD_UDKTK	Product Name	UST_URUN_AD
Product	Product Group Code	PRODUCT_GROUP_CODE	ABM_URUNKOD_UDKTK	Product Group Code	URUN_GRUP_KOD_OR
Product	Product Group Name	PRODUCT_GROUP_NAME	ABM_URUNKOD_UDKTK	Product Group Name	URUN_GRUP_AD
Product	Product Category Code	PRODUCT_CATEGORY_CODE	ABM_URUNKOD_UDKTK	Product Category Code	URUN_KATEGORI_KOD_OR
Product	Product Category Name	PRODUCT_CATEGORY_NAME	ABM_URUNKOD_UDKTK	Product Category Name	URUN_KATEGORI_AD



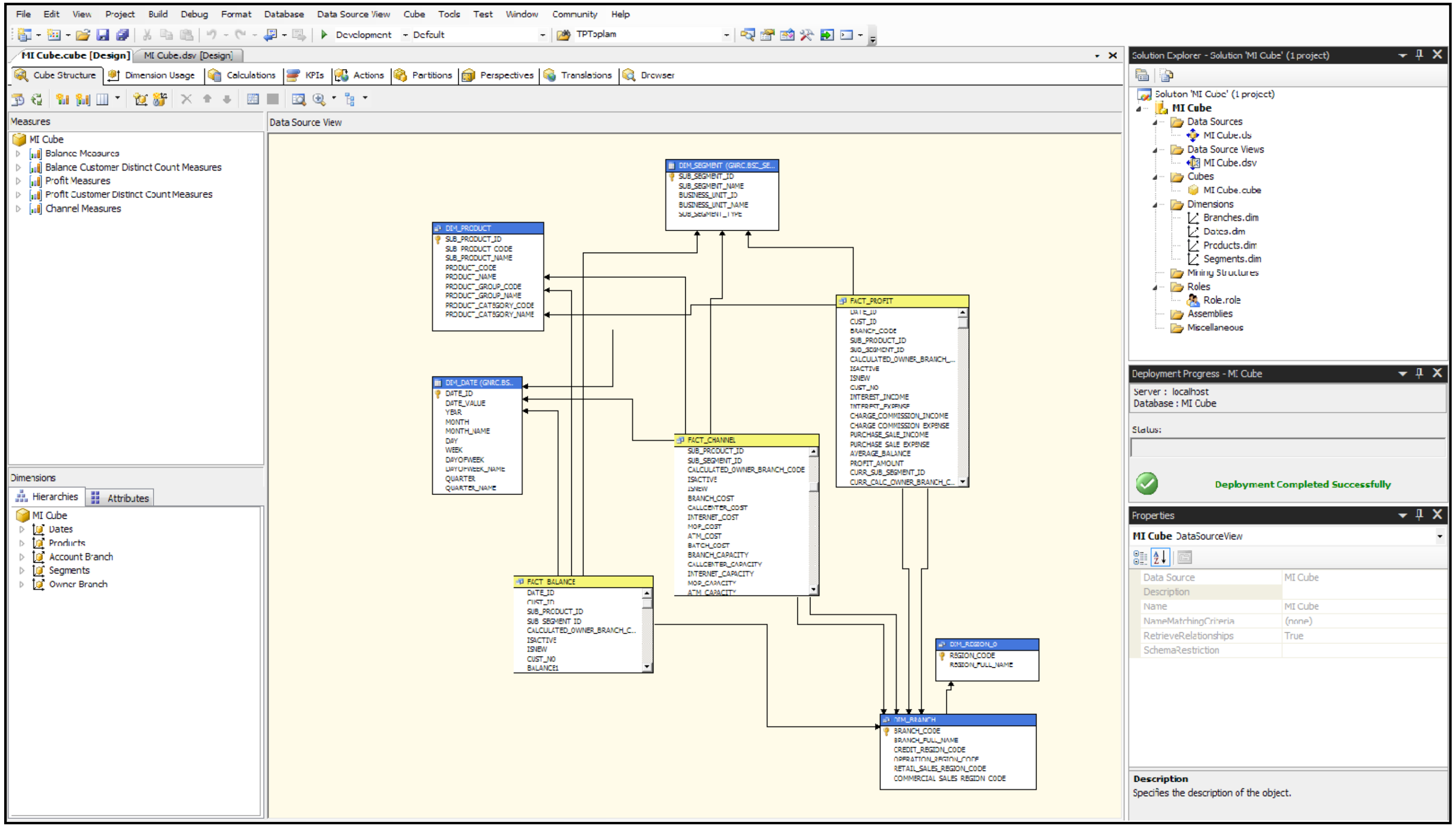
Target Table	Target Field Name	Target Physical Field Name	Source Table (DWH)	Source Field Name (English)	Source Physical Field Name
Description: Branch dimension table that will keep the branch information and their associated region information.					
Branch	Branch Code	BRANCH_CODE	GNL_SUBE_GSBTP	Branch Code	SUBE_KODU
Branch	Branch Name	BRANCH_NAME	GNL_SUBE_GSBTP	Branch Name	SUBADI
Branch	Affiliated Branch Code	AFFILIATED_BRANCH_CODE	GNL_SUBE_GSBTP	Affiliated Branch Code	BAGLI_SUBE_KODU
Branch	Status Code	STATUS_CODE	GNL_SUBE_GSBTP	Status Code	SUBE_DURUMU
Branch	Credit Region Number	CREDIT_REGION_NUMBER	GNL_SUBE_GSBTP	Credit Region Number	BAGLI_BOLGE_MUD_KODU
Branch	Credit Region Code	CREDIT_REGION_CODE	GNL_SUBE_GSBTP	Credit Region Code	BAGLI_BOLGE_MUD_SUBE_KODU
Branch	Credit Region Status	CREDIT_REGION_STATUS	GNL_SUBE_GSBTP	Credit Region Status	BOLGE_MUD_BAGLIMI
Branch	Operation Region Code	OPERATION_REGION_CODE	GNL_SUBE_GSBTP	Operation Region Code	BAGLI_BOL_MUD_OPER
Branch	Retail Sales Region Code	RETAIL_SALES_REGION_CODE	GNL_SUBE_GSBTP	Retail Sales Region Code	BAGLI_BOL_MUD_BSAT
Branch	Commercial Sales Region Code	COMMERCIAL_SALES_REGION_CODE	GNL_SUBE_GSBTP	Commercial Sales Region Code	BAGLI_BOL_MUD_TSAT
Branch	Information Region Code	INFORMATION_REGION_CODE	GNL_SUBE_GSBTP	Information Region Code	BAGLI_BOL_MUD_ISTH
Branch	Branch Type	BRANCH_TYPE	GNL_SUBE_GSBTP	Branch Type	SUBE_TIPI
Description: Region dimension table that will keep the region information.					
Region	Region Code	REGION_CODE	BLG_BOLGE_BLGTP	Region Code	BOLGE_KODU
Region	Region Name	REGION_NAME	BLG_BOLGE_BLGTP	Region Name	BOLGE_ADI
Region	Region Type	REGION_TYPE	BLG_BOLGE_BLGTP	Region Type	BOLGE_TIPI
Description: Date dimension table that will keep several date fields.					
Date	Unique Date ID	DATE_ID	Fed By ETL		
Date	Value of the Date	DATE_VALUE	Fed By ETL		
Date	Year	YEAR	Fed By ETL		
Date	Month	MONTH	Fed By ETL		
Date	Name of the Month	MONTH_NAME	Fed By ETL		
Date	Day	DAY	Fed By ETL		
Date	Week	WEEK	Fed By ETL		
Date	Day of Week	DAYOFWEEK	Fed By ETL		
Date	Name of the Day of Week	DAYOFWEEK_NAME	Fed By ETL		
Date	Quarter	QUARTER	Fed By ETL		
Date	Name of the Quarter	QUARTER_NAME	Fed By ETL		



Target Table	Target Field Name	Target Physical Field Name	Source Table (DWH)	Source Field Name (English)	Source Physical Field Name
Description: Fact table keeping periodical information about costs, capacities and channel transaction counts of customers.					
Channel Usage	Year	YEAR	ABM_KAROZET_ABMTA	Year	YIL
Channel Usage	Month	MONTH	ABM_KAROZET_ABMTA	Month	AY
Channel Usage	Customer Number	CUST_NO	ABM_KAROZET_ABMTA	Customer Number	MUSTERI_NO
Channel Usage	Branch Code	BRANCH_CODE	ABM_KAROZET_ABMTA	Branch Code	SUBE_KODU
Channel Usage	Account Number	ACCOUNT_NUMBER	ABM_KAROZET_ABMTA	Account Number	HESAP_NUMARASI
Channel Usage	Account Type	ACCOUNT_TYPE	ABM_KAROZET_ABMTA	Account Type	HESAP_TURU
Channel Usage	Sub Product Code	SUB_PRODUCT_CODE	ABM_KAROZET_ABMTA	Sub Product Code	URUN_KD_OR
Channel Usage	Status Code	STATUS_CODE	ABM_KAROZET_ABMTA	Status Code	ACIKKAPALI
Channel Usage	Customer's Total Cost to the Branch	BRANCH_COST	ABM_KAROZET_ABMTA	Customer's Total Cost to the Branch	
Channel Usage	Customer's Total Cost to the Callcenter	CALLCENTER_COST	ABM_KAROZET_ABMTA	Customer's Total Cost to the Callcenter	CMR_MALIYETI
Channel Usage	Customer's Total Cost to the Internet	INTERNET_COST	ABM_KAROZET_ABMTA	Customer's Total Cost to the Internet	INT_MALIYETI
Channel Usage	Customer's Total Cost to the ATM	ATM_COST	ABM_KAROZET_ABMTA	Customer's Total Cost to the Atm	ATM_MALIYETI
Channel Usage	Customer's Total Cost to the Batch	BATCH_COST	ABM_KAROZET_ABMTA	Customer's Total Cost to the Batch	BATCH_MALIYETI
Channel Usage	Customer's Total Cost to the Central Operations Center	MOP_COST	ABM_KAROZET_ABMTA	Customer's Total Cost to the Central Operations Center	MOP_MALIYETI
Channel Usage	Customer's Total Amount of Money used over Branch	BRANCH_CAPACITY	ABM_KAROZET_ABMTA	Customer's Total Amount of Money used over Branch	SUBE_HACIM
Channel Usage	Customer's Total Amount of Money used over Callcenter	CALLCENTER_CAPACITY	ABM_KAROZET_ABMTA	Customer's Total Amount of Money used over Callcenter	CMR_HACIM
Channel Usage	Customer's Total Amount of Money used over Internet	INTERNET_CAPACITY	ABM_KAROZET_ABMTA	Customer's Total Amount of Money used over Internet	INT_HACIM
Channel Usage	Customer's Total Amount of Money used over Atm	ATM_CAPACITY	ABM_KAROZET_ABMTA	Customer's Total Amount of Money used over Atm	ATM_HACIM
Channel Usage	Customer's Total Amount of Money used over Batch	BATCH_CAPACITY	ABM_KAROZET_ABMTA	Customer's Total Amount of Money used over Batch	BATCH_HACIM
Channel Usage	Customer's Total Amount of Money used over Central Operations Center	MOP_CAPACITY	ABM_KAROZET_ABMTA	Customer's Total Amount of Money used over Central Operations Center	MOP_HACIM
Channel Usage	Branch Transaction Count	BRANCH_COUNT	ABM_KAROZET_ABMTA	Branch Transaction Count	SUBE_ADET
Channel Usage	Callcenter Transaction Count	CALLCENTER_COUNT	ABM_KAROZET_ABMTA	Callcenter Transaction Count	CMR_ADET
Channel Usage	Internet Transaction Count	INTERNET_COUNT	ABM_KAROZET_ABMTA	Internet Transaction Count	INT_ADET
Channel Usage	Atm Transaction Count	ATM_COUNT	ABM_KAROZET_ABMTA	Atm Transaction Count	ATM_ADET
Channel Usage	Batch Transaction Count	BATCH_COUNT	ABM_KAROZET_ABMTA	Batch Transaction Count	BATCH_ADET
Channel Usage	Central Operations Center Transaction Count	MOP_COUNT	ABM_KAROZET_ABMTA	Central Operations Center Transaction Count	MOP_ADET
Channel Usage	Product Count	MOP_COUNT	ABM_KAROZET_ABMTA	Product Count	URUN_SAYISI

Target Table	Target Field Name	Target Physical Field Name	Source Table (DWH)	Source Field Name (English)	Source Physical Field Name
Description: Fact table keeping periodical information about incomes, expences, avarage balance and profits of customers.					
Profit	Year	YEAR	ABM_KAROZET_ABMTA	Year	YIL
Profit	Month	MONTH	ABM_KAROZET_ABMTA	Month	AY
Profit	Customer Number	CUST_NO	ABM_KAROZET_ABMTA	Customer Number	MUSTERI_NO
Profit	Branch Code	BRANCH_CODE	ABM_KAROZET_ABMTA	Branch Code	SUBE_KODU
Profit	Account Number	ACCOUNT_NUMBER	ABM_KAROZET_ABMTA	Account Number	HESAP_NUMARASI
Profit	Account Type	ACCOUNT_TYPE	ABM_KAROZET_ABMTA	Account Type	HESAP_TURU
Profit	Sub Product Code	SUB_PRODUCT_CODE	ABM_KAROZET_ABMTA	Sub Product Code	URUN_KD_OR
Profit	Status Code	STATUS_CODE	ABM_KAROZET_ABMTA	Status Code	ACIKKAPALI
Profit	Interest Income	INTEREST_INCOME	ABM_KAROZET_ABMTA	Interest Income	FAIZ_GELIRI
Profit	Interest Expense	INTEREST_EXPENSE	ABM_KAROZET_ABMTA	Interest Expense	FAIZ_GIDERI
Profit	Charge Commission Income	CHARGE_COMMISSION_INCOME	ABM_KAROZET_ABMTA	Charge Commission Income	MASRAF_KOMISYON_GELIRI
Profit	Charge Commission Expense	CHARGE_COMMISSION_EXPENSE	ABM_KAROZET_ABMTA	Charge Commission Expense	MASRAF_KOMISYON_GIDERI
Profit	Purchase Sale Income	PURCHASE_SALE_INCOME	ABM_KAROZET_ABMTA	Purchase Sale Income	ALIM_SATIM_GELIRI
Profit	Purchase Sale Expense	PURCHASE_SALE_EXPENSE	ABM_KAROZET_ABMTA	Purchase Sale Expense	ALIM_SATIM_GIDERI
Profit	Average Balance	AVERAGE_BALANCE	ABM_KAROZET_ABMTA	Average Balance	VASATI
Profit	Profit Amount	PROFIT_AMOUNT	ABM_KAROZET_ABMTA	Profit Amount	KAR
Description: Fact table keeping periodical information about balances of several products of the customers.					
Balance	Date ID	DATE_ID	ACH_KARTON_AKATA	Date ID	YUKLEME_TARIHI
Balance	Customer Number	CUST_NO	VDL_KARTON_VKATA	Customer Number	MUSTERI_NO
Balance	Account Branch Code	ACC_BRANCH_CODE	DVS_KARTON_SKATA	Account Branch Code	SUBE_KODU
Balance	Account Number	ACCOUNT_NUMBER	DVL_KARTON_LKATA	Account Number	HESAP_NO
Balance	Account Type	TYPE	PRT_DURUM_PDRTA	Account Type	HESAP_TIPI
Balance	Currency Code	FC_CODE	EKH_DURUM_EDRTA	Currency Code	DOVIZ_KODU
Balance	Original Balance	BALANCE_FC	TKR_DURUM_TDRTA	Original Balance	DOVIZ_BAKIYE
Balance	TL Balance	BALANCE	KRK_DURUM_KDRTA	TL Balance	TL_BAKIYE
			YDS_GUNLUKKARTON_DONEM_YGKTA		
			BCH_FIRMABCH_DONEM_BCHTA		
			DOS_DOSYADVZ_DONEM_DDVTA		
			TEM_TLTEMINAT_TTMTA		
			TEM_YPTEMINAT_YTMTA		

Appendix F: Detailed Design of the Cube



Server : localhost
Database : MI Cube

Status:

Deployment Completed Successfully

MI Cube DataSourceView

Data Source	MI Cube
Description	
Name	MI Cube
NameMatchingCriteria	(none)
RetrieveRelationships	True
SchemaRestriction	

Description
Specifies the description of the object.

Appendix G: The Questionnaire

Questionnaire for
Investigation of the Opportunities And Limitations of using OLAP Cube
for Managerial Reporting in Banking

This survey is a part of the master’s thesis that is being conducted by Onur Can Ulas, in the field of Management Information Systems at Bogaziçi University.

The questionnaire presented below has been designed to gather information from the IT experts and users regarding their opinions and perceptions about opportunities and limitations of using OLAP Cubes for managerial reporting in banking, in comparison with relational reporting.

An opportunity was assumed in this study as a pattern promoting the improvement of BI reporting (design, development, maintenance, speed etc.) and a limitation was considered as a constraint for use of BI technology.

The questionnaire consists of four sections

1. Personal information
2. Questions about performance issues of relational and/or OLAP Cube reporting
3. Questions about additional requirements to have an OLAP Cube
4. Questions about general features and/or functional properties of relational and/or OLAP Cube reporting

The questions are structured on a five-point scale (from -2 to +2) and ask you to indicate the extent of your agreement or disagreement with given statement. At the end of each section, you are asked to submit Additional Comments, which may clarify your responses.

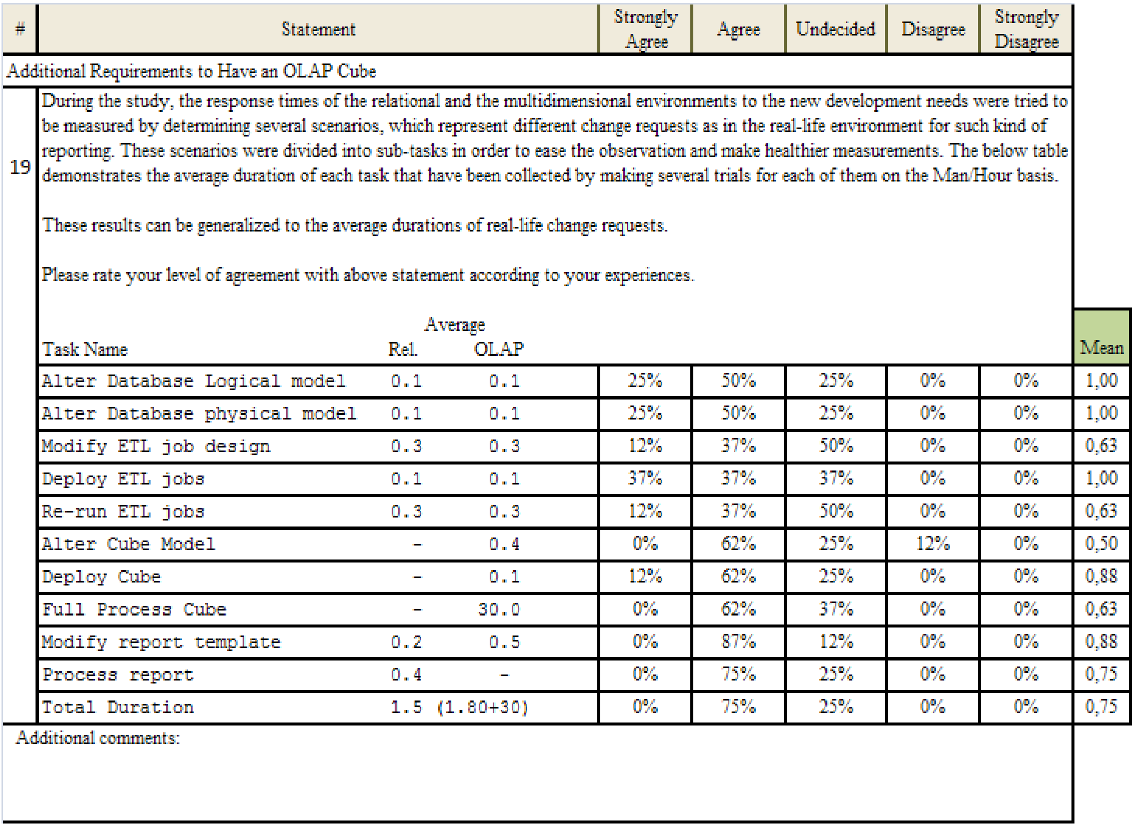
It will be highly appreciated if you complete the questionnaire. All of the information you provide will remain confidential.

Thank you very much for your cooperation.

Profession	ETL Developer, BI Developer x5, BI Director x2, Reporting Specialist, DBA, BI Project Manager
Number of years experience with BI	4,73
Number of BI projects participated in	9,00
Number of OLAP projects participated in	1,82

#	Statement	Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree	Mean
Performance Comparison: OLAP vs. Relational							
1	Execution cost of a query generally is a good estimate, which describes the performance and complexity level of a query.	36%	64%	0%	0%	0%	1,36
2	Number of queried rows, number of rows at the result set and the number of tables joined within the query are some other important features that determine the complexity of the queries.	45%	55%	0%	0%	0%	1,45
3	For the relational reporting, grouping (aggregation) operations can be more costly than reading the data from the database.	45%	36%	18%	0%	0%	1,27
4	Operations such as complex data aggregation, comparison and consolidation are more difficult in relational method than OLAP.	36%	45%	18%	0%	0%	1,18
5	OLAP provides very high query performance when it is compared with relational model.	64%	27%	9%	0%	0%	1,55
6	A good opportunity for using an OLAP cube in comparison with relational reporting is time intelligence (availability to make trend analysis).	55%	45%	0%	0%	0%	1,55
7	In relational reporting, even the most powerful hardware and the most optimized database cannot provide the performance of OLAP.	9%	9%	18%	64%	0%	-0,36
8	Running the SQL and MDX queries rather than the formatted reports provides better results on performance since data presentation duration would be eliminated.	9%	82%	9%	0%	0%	1,00
Additional comments:							

#	Statement	Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree	Mean
Additional Requirements to Have an OLAP Cube							
9	Building a data mart is necessary for relational systems in order to efficiently report such kind of information since there are complex aggregations and queries need to run on optimized data models	36%	55%	9%	0%	0%	1,27
10	Building an external data mart is necessary for OLAP systems since the intensive use of the database during the Cube processing would negatively effect other usages of the system.	9%	9%	45%	36%	0%	-0,09
11	Maintenance of OLAP systems is generally harder than the relational model.	27%	36%	27%	9%	0%	0,82
12	The necessary effort for daily and monthly update of OLAP cube is greater than the relational system because of OLAP's processing needs.	36%	55%	9%	0%	0%	1,27
13	An OLAP Cube of this size necessitates manual interference for monthly and full processing	18%	9%	45%	27%	0%	0,18
14	Any change in OLAP model or the major changes at the source data necessitates full processing of the Cube	55%	45%	0%	0%	0%	1,55
15	OLAP Cube generally requires longer duration than relational system to respond the new development needs.	27%	55%	18%	0%	0%	1,09
16	If full processing the Cube was not necessary, a change can be implemented approximately within the same duration with the relational system.	0%	55%	27%	18%	0%	0,36
17	Due to long process durations of OLAP Cubes, up to dateness with the DW may not be maintained.	27%	36%	27%	9%	0%	0,82
18	The necessary disc capacity to have an OLAP Cube is almost equal to the data size of the relational database.	0%	18%	55%	27%	0%	-0,09



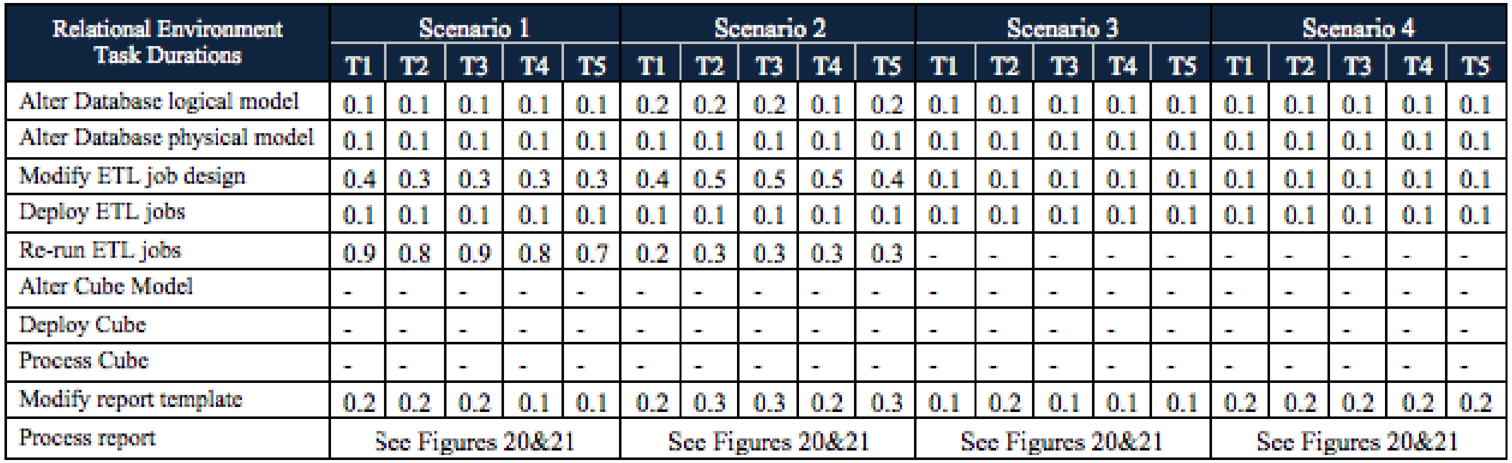
#	Statement			Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree	
Additional Requirements to Have an OLAP Cube									
19	During the study, the response times of the relational and the multidimensional environments to the new development needs were tried to be measured by determining several scenarios, which represent different change requests as in the real-life environment for such kind of reporting. These scenarios were divided into sub-tasks in order to ease the observation and make healthier measurements. The below table demonstrates the average duration of each task that have been collected by making several trials for each of them on the Man/Hour basis. These results can be generalized to the average durations of real-life change requests. Please rate your level of agreement with above statement according to your experiences.								
		Average							
	Task Name	Rel.	OLAP						Mean
	Alter Database Logical model	0.1	0.1	25%	50%	25%	0%	0%	1,00
	Alter Database physical model	0.1	0.1	25%	50%	25%	0%	0%	1,00
	Modify ETL job design	0.3	0.3	12%	37%	50%	0%	0%	0,63
	Deploy ETL jobs	0.1	0.1	37%	37%	37%	0%	0%	1,00
	Re-run ETL jobs	0.3	0.3	12%	37%	50%	0%	0%	0,63
	Alter Cube Model	-	0.4	0%	62%	25%	12%	0%	0,50
	Deploy Cube	-	0.1	12%	62%	25%	0%	0%	0,88
	Full Process Cube	-	30.0	0%	62%	37%	0%	0%	0,63
	Modify report template	0.2	0.5	0%	87%	12%	0%	0%	0,88
	Process report	0.4	-	0%	75%	25%	0%	0%	0,75
	Total Duration	1.5	(1.80+30)	0%	75%	25%	0%	0%	0,75
Additional comments:									

#	Statement	Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree	Mean
General Features of OLAP and Relational Reporting							
20	Ad hoc query creation is easier and more flexible with OLAP tools when compared to the relational reporting.	18%	64%	0%	18%	0%	0,82
21	Short query response times and capabilities like drilling-down, slicing & dicing of OLAP provide interactivity with the data for users while it is limited for the relational system.	73%	27%	0%	0%	0%	1,73
22	Relational reporting tools are better than OLAP reporting tools at developing formatted static reports.	18%	64%	18%	0%	0%	1,00
23	A good opportunity for using an OLAP Cube in comparison with relational reporting is simplicity to model complex environments.	9%	55%	36%	0%	0%	0,73
24	OLAP is more appropriate than relational model for managerial reporting in banking.	36%	55%	9%	0%	0%	1,27
25	The granular data in the relational model is also used to service unknown future needs for information while OLAP cube only meet existing needs of existing user groups.	45%	36%	9%	9%	0%	1,18
26	Flexibility of OLAP systems can be considered as a limitation when compared to the relational model.	36%	45%	18%	0%	0%	1,18
27	OLAP cube and data marts are not scalable; those are departmental and not enterprise solutions and only appropriate when there is a predictable usage.	9%	82%	9%	0%	0%	1,00
28	OLAP Cubes are not proper tools for rapidly changing environments.	9%	82%	0%	9%	0%	0,91
29	Total cost of ownership of OLAP for MI reporting is more than the relational system	0%	36%	45%	18%	0%	0,18
30	Querying with MDX language is more difficult than querying with SQL because of the technical complexity of the MDX language.	36%	36%	18%	0%	9%	0,91
Additional comments:							

Appendix H: Durations of Trials – Report Processing

Query Duration (Seconds)		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Relational (Aggregate)	Trial 1	478.399	506.273	507.551	535.448	679.457	663.965	-	9,406.247
	Trial 2	487.626	484.864	423.853	444.31	681.78	669.263	-	7,626.527
	Trial 3	392.384	488.473	456.211	469.921	667.314	672.512	-	7,770.452
	Trial 4	552.481	500.201	628.638	641.173	620.267	632.196	-	8,662.342
	Trial 5	529.56	507.469	510.927	531.295	672.451	634.215	-	7,389.141
	Avg.	488.09	497.456	505.436	524.429	664.253	654.430	-	8,170.941
Relational (Non Aggregate)	Trial 1	182.629	207.231	317.912	318.395	444.743	506.426	964.956	4,642.233
	Trial 2	185.404	180.451	226.524	224.293	476.543	573.583	733.611	4,154.363
	Trial 3	210.583	182.596	260.847	393.065	381.334	379.241	631.054	2,946.58
	Trial 4	251.981	211.845	303.014	302.138	370.233	534.743	624.34	2,995.435
	Trial 5	235.419	263.3	235.437	233.974	336.419	390.097	635.694	3,006.231
	Avg.	213.203	209.084	268.746	294.373	401.854	476.818	717.931	3,548.968
OLAP	Trial 1	0.115	0.096	0.205	10.079	1.016	3.782	6.738	130.226
	Trial 2	0.065	0.07	0.063	9.094	0.953	3.095	6.46	137.785
	Trial 3	0.072	0.067	0.079	9.172	0.984	3.126	6.469	133.471
	Trial 4	0.078	0.073	0.08	9.25	0.968	3.188	6.363	134.32
	Trial 5	0.06	0.062	0.081	9.39	0.969	3.142	6.147	136.218
	Avg.	0.078	0.0736	0.1016	9.397	0.978	3.2666	6.4354	134.404

Appendix I: Durations of Trials – Additional Requirements (Relational Environment)



Relational Environment Task Durations	Scenario 1					Scenario 2					Scenario 3					Scenario 4				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
Alter Database logical model	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Alter Database physical model	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Modify ETL job design	0.4	0.3	0.3	0.3	0.3	0.4	0.5	0.5	0.5	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Deploy ETL jobs	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Re-run ETL jobs	0.9	0.8	0.9	0.8	0.7	0.2	0.3	0.3	0.3	0.3	-	-	-	-	-	-	-	-	-	-
Alter Cube Model	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Deploy Cube	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Process Cube	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Modify report template	0.2	0.2	0.2	0.1	0.1	0.2	0.3	0.3	0.2	0.3	0.1	0.2	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.2
Process report	See Figures 20&21					See Figures 20&21					See Figures 20&21					See Figures 20&21				

Appendix J: Durations of Trials – Additional Requirements (OLAP Environment)

OLAP Environment Task Durations	Scenario 1					Scenario 2					Scenario 3					Scenario 4				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
Alter Database logical model	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Alter Database physical model	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Modify ETL job design	0.4	0.3	0.3	0.3	0.3	0.4	0.5	0.5	0.5	0.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Deploy ETL jobs	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Re-run ETL jobs	0.9	0.8	0.9	0.8	0.7	0.2	0.3	0.3	0.3	0.3	-	-	-	-	-	-	-	-	-	-
Alter Cube Model	0.3	0.3	0.2	0.2	0.2	1	0.8	0.8	0.8	0.8	0.2	0.1	0.1	0.2	0.1	0.5	0.5	0.4	0.5	0.4
Deploy Cube	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Process Cube	See Table 13					See Table 13					See Table 13					See Table 13				
Modify report template	0.7	0.5	0.4	0.5	0.4	0.9	0.9	0.8	0.8	0.8	0.4	0.3	0.4	0.3	0.3	0.5	0.6	0.5	0.5	0.5
Process report	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Appendix K: Break Even Analysis

Scenario parameters:
One field needs to be added to the Segment dimension
15 reports have to be modified
Relational reporting system can process 2 reports at the same time
Average run duration of the reports is 24 minutes (0.4 MH)
Duration necessary to process 1 term is: 24 min x 15 reports / 2 parallel=180 min
Number of terms to process: 36 months
One man works 10 hours a day
One Cube partition is processed within 120 minutes
Altering the relational model takes 60 minutes
Altering the relational reports takes 15 x 12 min. (0.2 MH)= 180 min.
Altering Relational + Cube Model: 60=60+120 min.
Altering Cube reports: 15 x 32 (0.5 MH) = 480 min.

Day	1	2	3	4	5	6	7	8	9	10	11	12	Total
OLAP													
MODEL CHANGE	600	0	0	0	0	0	0	0	0	0	0	0	600
PROCESS 3 year (120 min/partition)	0	600	600	600	600	600	600	600	120	0	0	0	4,32
RELATIONAL													
MODEL CHANGE	240	0	0	0	0	0	0	0	0	0	0	0	240
REPORT PROCESS 3 years (180 min/term)	360	600	600	600	600	600	600	600	600	600	600	120	6,480
Number of Reports updated - OLAP	0	75	75	75	75	75	75	75	15	0	0	0	540
Number of Reports updated - Relational	30	50	50	50	50	50	50	50	50	50	50	10	540
Cumulative Difference (Relational - OLAP)	30	5	-20	-45	-70	-95	-120	-145	-110	-60	-10	0	-
Break Even Point	*2.20 days*												

REFERENCES

Bach, M. P., Strugar, I. & Jaković, B. (2007). Influence of business intelligence implementation to the banks' performance: Case study of Croatian banking sector. *Proceedings from Conference IIS 2007: Information and Intelligent Systems*. Zagreb: Croatia.

Banking business intelligence. (n.d.). In *Academic Dictionaries and Encyclopedias.* Retrieved February 22, 2011, from http://en.academic.ru/dic.nsf/enwiki/10622710.

Boselli, R., Cesarini, M. & Mezzanzanica, M. (2010, October). Business Intelligence Exploitation for Investigating Territorial Systems, Methodological Overviews and Empirical Considerations. *Paper presented at European Association for Research on Services Conference, Gothenburg, Sweden.*

Brayman, A. & Bell, E. (2003). *Business Research Methods*. Oxford: Oxford, University Press.

Business Process Modeling Tools. (n.d.). *Startup BizHub.* Retrieved February 26, 2011, from http://www.startupbizhub.com/business-process-modeling-tools.htm.

Carpi, A. & Egger, A. E. (2008). Research Methods: The Practice of Science. *Visionlearning*. Vol. POS-2 (1), 2008.

Chaudhuri, S. & Dayal, U. (1997, March). An Overview of Data Warehousing and OLAP Technology. *ACM SIGMOD Newsletter,* 26(1). Retrieved March 9, 2011, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.6667.

Codd, E. F. (1970, June) "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM*: Vol. 13, 6, p. 377-387. Retrieved February 28, 2011, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.5286&rep=rep1&type=pdf.

Codd E. F., Codd S. B., & Salley C. T. (1993). *Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate* (Technical Report). Codd & Associates. San Jose, California.

Cooper, D. R. & Schindler, P. S. (2003). *Business Research Methods* (Eighth ed.). Boston:Erwin, McGraw-Hill.

Coveney, M. (2003). Corporation Performance Management (CPM): What is it and how It differs from traditional approaches ?.*The Business Forum Online.* Retrieved March 14, 2011, from http://www.businessforum.com/Comshare01.html.

Cubes (n. d.).*On-Line Analytical Processing (OLAP) Tutorial*. Retrieved March 5, 2011 from http://training.inet.com/OLAP/Cubes.htm.

Drewek, K. (2005a, March). Data Warehousing: Similarities and Differences of Inmon and Kimball. *Beye Network.* Retrieved February 16, 2011, from http://www.b-eye-network.com/view/743.

Drewek, K. (2005b, April). Data Warehousing: Our Great Debate Wraps Up. *Beye Network.* Retrieved March 8, 2011, from http://www.b-eye-network.com/view/766.

Eckerson, W. & Hammond, M. (2011). *Visual reporting and analysis: Seeing is knowing* (TDWI Best Practice Report). Retrieved May 24, 2011 from http://www.datavisualization.fr/files/tdwi_bpreport_q111_vra_web.pdf.

Ghosh, S. & Mukherjee, S. (2006, March). Measurement of corporate performance through balanced scorecard: An overview. *Vidyasagar University Journal of Commerce*, 11, 60-70.

Grekov, B. (2009). *Analysis and comparison of available solutions for OLAP (Online Analytical Processing) of data warehouses.* (Bachelor thesis, Czech Technical University, Czech Republic).

Greener, S. (2008). *Business Research Methods*. Vantes Publishing ApS. Retrieved March 26, 2011, from http://educationcity.weebly.com/uploads/4/4/5/1/4451099/introduction-to-research-methods1.pdf.

Guo, Z. (2009). *Partial aggregation and query processing of OLAP cubes* (Unpublished master's thesis, Simon Fraser University).

Hindriks, C. (2007). *Towards chain wide business intelligence.* (Master's thesis, University of Twente, The Netherlands).

Inmon, W. H. (2005). *Building the Data Warehouse* (4th ed.). New York: Wiley Publishing.

Jensen, N. A. (2006). *Applied data mining for business intelligence* (Unpublished master's thesis, Technical University of Denmark).

Karayannidis, N. N. (2003). *Storage structures, query processing and implementation of on-line analytical processing systems* (Doctoral dissertation, National Technical University of Athens).

Kimball, R. & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (2nd ed.). New York: John Wiley & Sons.

Kimball, R., Ross, M., Thornthwaite, W., Mundy, J. & Becker, B. (2008). *The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems* (2nd ed.). New York: John Wiley

& Sons.

Knapik, J. (2007, December). Business intelligence in retail banking. *Datamonitor.* Retrieved March 23, 2009 from http://www.researchandmarkets.com/reports/586519.

Lim, A. (n.d.). *The Importance of Activity-Based Costing/Management to the Balanced Scorecard.* Retrieved February 27, 2011, from http://www.apmss.com.sg/pdf/importanceof-abcin-bsc.pdf.

Linders, S. (2008). *Opportunities and limitations of using SOA concepts and technologies for building BI applications: A Delphi study* (Unpublished master's thesis, University of Twente).

Luhn, H. P. (1958). Business intelligence system. *IBM Journal of Research and Development.* 2 (4). 314-319.

Misra, H. (2007, November). Operational Business Intelligence in Banking. *Indian Banks' Association Special Report by Finsight Media.* Retrieved January 12, 2011, from http://www.maiaintelligence.com/pdf/IBA%20Research%20Report%20on%20Operational%20BI.pdf.

Nadeem, M. & Jaffri, S. A. H. (2004, January). Application of business intelligence in banks. *Journal of Independent Studies and Research* (JISR) 2 (1).

Negash, S. (2004) "Business Intelligence" *Communications of the Association for Information Systems*: Vol. 13, Article 15.

Neuman, W. L. (2007). *Basics of Social Research: Qualitative and Quantitative Approaches* (2nd edition). London: Pearson Education-Allyn and Bacon.

OLAP Analysis vs Relational Reporting. (n.d.). *OLAP Business Solutions.* Retrieved March 9, 2011, from http://www.obs3.com/olap_vs_relational.shtml.

OLAP Functionality (n.d.). In *OLAP Software and Education Wiki.* Retrieved March12, 2011, from http://www.olap.com/w/index.php/OLAP_Functionality.

OLAP: On-Line Analytical Processing (n.d.). *OLAP Council.* Retrieved March 8, 2011, from http://www.olapcouncil.org/research/glossaryly.htm#Defined.

O'Neil, P. & Quass, D. (1997) *Improved Query Performance with Variant Indexes.* In: ACM International Conference on Management of Data (SIGMOD 1997), May 13-15, 1997, Tucson, Arizona. Retreived May 20, 2010 from http://ilpubs.stanford.edu:8090/253/.

Pendse, N. (2002). The origins of today's OLAP products. *DSSResources.COM*, Retrieved May, 24, 2011 from http://dssresources.com/papers/features/pendse10062002.html.

Pendse, N. (2005). What is OLAP?: An analysis of what the often misused OLAP term is supposed to mean. *The BI Verdict*. Retrieved March 8, 2011, from Business Application Research Center website: http://www.bi-verdict.com/fileadmin/dl_temp/5ec91a514212a02e810d44616b382e14/fasmi.htm.

Pirttimaki, V. (2007). *Business intelligence as a managerial tool in large Finnish companies* (Doctoral dissertation, Tampere University of Technology).

Power, D. J. (2007). A brief history of decision support systems. *DSSResources.COM*. Retrieved May, 24, 2011 from http://DSSResources.COM/history/dsshistory.html.

Qihai, Z., Tao, H.. & Tao, W. (2008). Analysis of business intelligence and its derivative- Financial intelligence. *Proceedings from 2008 International Symposium on Electronic Commerce and Security*.

Ritacco, M. & Carver, A. (2007). The business value of business intelligence. *Business Trends Quarterly*. Retreived March 21, 2009 from http://www.btquarterly.com/?mc=business-value-bi&page=bi-viewmarketplace.

Retail banking MIS (n.d.). Retrieved February 21, 2009 from http://www.exodussa.com/Default.aspx?id=1530&nt=18.

Rittman, M. (2005). What is sparsity? Retrieved May, 24, 2011 from http://www.dba-oracle.com/oracle_news/2005_6_27_What_Is_Sparsity.htm.

Robinson, M. (2008, May). Business Intelligence Infrastructure. *Information Management Online*. Retrieved February 16, 2011 from http://www.information-management.com/specialreports/20020521/5211-1.html.

Tabatabaei, S. H. (2009). *Evaluation of business intelligence maturity level in Iranian banking industry*. (Master’s thesis, Lulea University of Technology, Sweden).

Tam, Y. J. (1998). *Datacube: Its implementation and ipplication in OLAP mining*. (Master’s thesis, Simon Fraser University, Canada).

Thomsen, E. (2002). *OLAP Solutions: Building Multidimensional Information Systems* (Second ed.). New York: John Wiley & Sons.

Thornthwaite, W. & Mundy, J. (2006, February). Standard Reports: Basic for Business Users. *Information Week*. Retrieved February 26, 2011 from http://www.informationweek.com/news/software/bi/showArticle.jhtml?articleID=177103011.

Trepte, K. (1997, November). Business Intelligence Tools. *Information Management Online*. Retreived March 12, 2011, from http://www.information-management.com/issues/19971101/964-1.html.

Trochim, M. K. (2006). *Research methods knowledge base*. Retrieved March 28, 2011, from http://www.socialresearchmethods.net/kb/qual.php.

Types of OLAP Systems (n.d.). In *OLAP Software and Education Wiki.* Retrieved March 12, 2011, from http://www.olap.com/w/index.php/Types_of_OLAP_Systems.

Westerlund, P. (2008). *Business intelligence: Multidimensional data analysis.* (Master's thesis, Umea University, Sweden).

Wu, J. (2000, February). What is business intelligence? *Information Management Online*. Retreived March 22, 2009 from http://www.information-management.com/news/1924-1.html.

Yin, R. K. (2003). *Case Study Research: Design and Methods* (Third ed.). Thousand Oaks:CA, Sage Publications.

Zavareh, J. T. (2007). *The role of analytical CRM in maximizing customer profitability in private banking.* (Master's thesis, Lulea University of Technology, Sweden).