ATHABASCA UNIVERSITY

MODERN METHODOLOGY & TOOLS
FOR DATA WAREHOUSE DEVELOPMENT

BY

INTIKHAB KHAN

A project submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE in INFORMATION SYSTEMS

Athabasca, Alberta

December, 2012

© Intikhab Khan, 2012
I would like to dedicate this paper to my late father and mother who always had confidence in me and offered me encouragement and support in all my endeavours;
ABSTRACT

Data Warehousing (DW) methodologies are continuously being developed and are widely different because the field of data warehousing is not well established. Therefore, it is a challenge for organizations to get the right information for measuring their performance in different areas such as customer satisfaction, customer service, company’s standing and accomplishments.

An enormous number of data warehousing methodologies and tools are available in the market to help different types of organizations to develop decision support systems and achieve their strategic goals. The question always asked is which methodology is trustworthy and which tools can help them to develop the system they are looking for. A huge amount of money is spent by organizations when determining methodologies and tools that best suit them and benefits them the most.

This paper includes an overview of Data Warehousing technologies and modern tools available in the market. In addition, we perform a comparison between the most prominent and useable advanced methodologies of data warehousing development. We compare Ralph Kimball team’s Dimensional Design Methodology and W.H. Inmon’s Data Driven Methodology and discusses all the steps performed by these methodologies during their DW development lifecycle process.
The paper also covers some basic technological terms used in data warehousing such as extraction, transformation and loading process, online analytical processing and data mining.
ACKNOWLEDGMENTS

I would like to express my special gratitude to my supervisor Dr. Larbi Esmahi for supervising this essay and for providing very valuable insight into the subject matter. His guidance helped me in all the time of research and writing of this paper. I could not have imagined having a better advisor and mentor for this paper.

I would like to thank the faculty and staff of the School of Computing and Information Systems at Athabasca University for their expert guidance throughout this Program.

Last but not the least, without the encouragement and support of my family, specially my wife Shazia Intikhab and three kids (Haroon Khan - 11, Hiba Khan – 9 and Haris Khan - 7); this accomplishment would not have been possible.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................................................. 5

GLOSSARY OF TERMS ............................................................................................ 7

CHAPTER I ................................................................................................................. 9
   INTRODUCTION .................................................................................................... 9
   STATEMENT OF PURPOSE ............................................................................... 12
   RESEARCH PROBLEMS ................................................................................. 13
   SIGNIFICANCE ................................................................................................. 13
   ASSUMPTIONS ................................................................................................. 14
   LIMITATIONS ................................................................................................. 15
   ORGANIZATION OF ESSAY ........................................................................... 15

CHAPTER II ............................................................................................................. 17
   REVIEW OF RELATED LITERATURE ............................................................. 17
      Data Warehouse ........................................................................................... 17
      Data Reconciliation Support Tools ............................................................ 29

CHAPTER III ......................................................................................................... 42
   METHODOLOGY .............................................................................................. 42
      Dimensional Design Methodology .............................................................. 42
      Managing Project / Program ...................................................................... 43
      Business Requirements Definition ........................................................... 45
      Technical Architecture Design ................................................................. 46
      Dimensional Modeling ............................................................................... 48
      Physical Design ........................................................................................... 51
      Business Intelligence Application Design ............................................... 53
      Business Intelligence Application Development ...................................... 55
      Product Selection & Installation ................................................................. 56
      Deployment .................................................................................................. 58
      Maintenance ................................................................................................ 58
      Growth ......................................................................................................... 59
   Data Driven Methodology ............................................................................ 59
   METH 2 – Data Warehouse Development ..................................................... 61
      Data Model Analysis ................................................................................... 61
      Breadbox Analysis ....................................................................................... 66
LIST OF TABLES

Table 1 - Comparison of operational and informational systems (Hoffer et al., 2008:426) ........ 20
Table 2 - Comparison between DW and DM (Hoffer et al., 2008:470) ........................................ 23
Table 3 – A comparison between different DW architectures (Abdolreza et al., 2011) ........... 29
Table 4 - Tools to support data reconciliation (Hoffer et al., 2008:451) ........................................ 33
Table 5 - Typical data mining applications (Hoffer et al., 2008:470) ........................................... 40
Table 6 – Complexity of the two methods .................................................................................. 77
Table 7 - Comparison of established development Methodologies ............................................ 79
Table 8 – Physical Design ............................................................................................................. 80
Table 9 – Data Orientation ........................................................................................................... 80
Table 10 - Tools ........................................................................................................................... 81
Table 11 – End user accessibility ................................................................................................ 81
Table 12 – Primary Audience ................................................................................................... 82
Table 13 – Place in the Organization .......................................................................................... 82
Table 14 - Objectives ................................................................................................................... 82
Table 15 – Characteristics .......................................................................................................... 83
Table 16 – ETL three year total cost of ownership ...................................................................... 92
Table 17 – Ramp up time and efforts for Informatica and SSIS (Philip, Howard, 2010) ........... 94
Table 18 – Tools Comparison as per Survey Result (Philip, Howard. 2010) .............................. 95
Table 19 - Informatica PowerCenter and SSIS comparison ......................................................... 96
Table 20 – Test case study environment ....................................................................................... 99
Table 21 – Case study outcome data after using Informatica PowerCenter ................................... 104
Table 22 – Case Study outcome data after using SSIS ................................................................ 104
Table 23 - Main Advantages and Disadvantages of Kimball & Inmon’s Methodologies .......... 110
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Generic two-level DW architecture (Hoffer et al., 2008:428)</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>Independent DM, DW Architecture (Hoffer et al., 2008:430)</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>Dependent DM, DW architecture (Hoffer et al., 2008:431)</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>Logical DM and real-time DW architecture (Hoffer et al., 2008:433)</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>Steps in data reconciliation (Hoffer et al., 2008:442)</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>High level DW/BI system architecture model (Kimball et al., 2008:114)</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>Single-field transformation (Hoffer et al., 2008:451)</td>
<td>34</td>
</tr>
<tr>
<td>8</td>
<td>Algorithmes transformation (Hoffer et al., 2008:451)</td>
<td>34</td>
</tr>
<tr>
<td>9</td>
<td>Table look-up (Hoffer et al., 2008:451)</td>
<td>34</td>
</tr>
<tr>
<td>10</td>
<td>Dimensional Model Lifecycle (Kimball et al., 2008:3)</td>
<td>43</td>
</tr>
<tr>
<td>11</td>
<td>Components of Star Schema (Hoffer et al., 2008:454)</td>
<td>49</td>
</tr>
<tr>
<td>12</td>
<td>Dimensional Modeling Process Flow Diagram (Kimball et al., 2008:288)</td>
<td>50</td>
</tr>
<tr>
<td>13</td>
<td>High Level Physical Design Process (Kimball et al., 2008:329)</td>
<td>53</td>
</tr>
<tr>
<td>14</td>
<td>Analytic cycle for BI analysis (Kimball et al., 2008:477)</td>
<td>55</td>
</tr>
<tr>
<td>15</td>
<td>System Development lifecycle for DW Environment (Inmon, 2002:34)</td>
<td>60</td>
</tr>
<tr>
<td>16</td>
<td>METH 2 (Inmon, 2002:365)</td>
<td>61</td>
</tr>
<tr>
<td>17</td>
<td>How different Levels of Modeling Relate (Inmon, 2002:90)</td>
<td>62</td>
</tr>
<tr>
<td>18</td>
<td>An Example of Stability Analysis (Inmon, 2002:91)</td>
<td>63</td>
</tr>
<tr>
<td>19</td>
<td>First Level Modeling (ER Diagram) (Inmon, 2002:92)</td>
<td>64</td>
</tr>
<tr>
<td>20</td>
<td>Four Constructs that make up midlevel data model (Inmon, 2002:95)</td>
<td>65</td>
</tr>
<tr>
<td>21</td>
<td>Example of a Physical Data Model (Inmon, 2002:99)</td>
<td>66</td>
</tr>
<tr>
<td>22</td>
<td>3 Dimensional perspectives of the entities (Inmon, 2002:139)</td>
<td>69</td>
</tr>
<tr>
<td>23</td>
<td>A simple Star Join (Inmon, 2002:140)</td>
<td>70</td>
</tr>
</tbody>
</table>
Figure 24 - Feedback loop between DSS analyst and data architect (Inmon, 2002:286) .......... 73
Figure 25 – Inmon’s top-down approach ................................................................. 75
Figure 26 – Kimball’s bottom-up approach ............................................................ 75
Figure 27 – Inmon’s architectural structure ............................................................. 76
Figure 28 – Kimball’s architectural structure ......................................................... 76
Figure 29 – Informatica PowerCenter Architecture ................................................. 90
Figure 30 – SQL Server Integration Services (SSIS) Architecture ......................... 91
Figure 31 – Average number of sources and targets per projects (Philip, Howard. 2010) ........ 94
Figure 32 - Logical diagrams of Student Application and Registration schemas .......... 98
Figure 33 – list of SA & SR tables and data size loaded into schema ....................... 99
Figure 34 – Case study scenarios ............................................................................. 100
Figure 35 – Dimensions and facts tables of students’ Application and Registration Data Marts 101
GLOSSARY OF TERMS

DSS – Decision Support System

DW – Data Warehouse

ETL – Extraction, Transformation and Loading

ERD – Entity Relationship Diagram

OLAP – Online Analytical Processing

OLTP – Online Transaction Processing

EDW – Enterprise Data Warehouse

ODS – Operational Data Store

CIF – Corporate Informative Factory

ODBC – Open Database Connectivity

OLE – Object Linking and Embedding

BI – Business Intelligence

PL – Procedural Language

SQL – Sequential Query Language

MOLAP – Multidimensional Online Analytical Processing

ROLAP – Relational Online Analytical Processing

GUI – Graphical User Interface

KPI – Key Performance Indicator

RDBMS – Relational Database Management System

EBA – Enterprise Bus Architecture

SDLC – System Development Lifecycle

DIS – Data Item Set
DASD – Direct Access Storage Device

SSIS – SQL Server Integration Services
CHAPTER I

INTRODUCTION

A vast number of data warehousing methodologies and tools are currently available in the market to support the growing DW market. There are many methodologies to choose from; therefore, a big challenge for many firms is which one to use for a specific data warehousing project. An organizational decision made by executives requires a complete view of all aspects of the organization (Ramakrishnan and Gehrke, 2003). Despite the vast amount of data collected by an organization these days, majority of the managers have difficulty in identifying and collecting the information they need to help them make a strategic decision to reach their goal. This paper discusses some state of the art technologies and tools used for data warehousing projects.

Modern methodologies and tools are playing an essential role in the development of a data warehousing application for strategic planning and decision support systems; but rarely a methodology has achieved the highest possible quality of a data warehousing methodology. It is strongly believed that successful data warehousing depends on three fundamentals: focus on business model, dimensionally structured data that deliver to the business via ad hoc queries or reports and iteratively develop the overall data warehousing environment in manageable lifecycle increments (Kimball et al., 2008).
Most of the time, it is observed that a data warehouse, which is developed using traditional approaches and tools provides reasonable services without any interruption. However, for most big organizations’ business needs, these approaches hardly suffice, since they require time consuming and costly solutions.

Changes in the business needs require IT staff to work for several months to re-design and implement new solutions. On the other hand, using the latest DW tools and approaches can significantly reduce the time and cost of implementing new business requirements. The sources of existing methodologies for data warehousing can be classified into three categories:

- Core-technology vendors
- Infrastructure vendors
- Information modeling companies

(Arun & Atish, 2005).

In this work, we discuss distinct attributes that these methodologies considered and analysed in the process of data warehousing development, and how these attributes are qualified for data warehousing development process. We discuss how data Integration technologies have experienced tremendous growth in the last decade and how data warehousing tools are playing a significant role in this integration process.
A large variety of data warehousing methodologies and tools are available in the market that organizations can use to develop a data warehouse and business intelligence application to help them support managerial decisions. Whatever methodology and tool they decide to use, two things should always be considered: time variance and non-volatility. Although the methodologies used by different companies differ in details, they all focus on the techniques of capturing and modeling user requirements in a meaningful way” (Arun & Atish, 2005).

One of the most recognised methodologies used in data warehouse development are from Kimball and Inmon paper in 2005 (Sen & Sinha, 2005). Kimball and Inmon proposed two different approaches, which are most commonly used in Data warehousing methodologies. However, delivering dimensional data to Data Warehouse system is not enough we also need to provide proper tools for OLAP (Kimball et al., 2008).

There are four major Data Warehouse development methodologies found in literature as listed below:

- Top-down
- Bottom-up
- Data Vault (Hybrid)
- Federated

There “data vault” and “federated” are more or less derivatives of top-down and bottom-up methodologies.
This essay is to explain two of the most prominent existing methodologies regarding data warehousing and some tools to deliver a powerful business intelligence application. A brief of these methodologies is given below.

**Dimensional Design Methodology**

Ralph Kimball suggested a bottom-up methodology, in which individual data marts are created. These data marts provide microscopic views into the organizational data. Later these data marts are aggregated into larger data warehouses (Kimball et al., 2008).

**Data Driven Methodology**

Data Driven Methodology is in contrast to dimensional design methodology. In 2002, William Inmon suggested that, the data warehouse should be designed using a top-down method, to include all the organization data. In this methodology, data marts will be created after the complete data warehouse is created (Inmon, 2002).

**STATEMENT OF PURPOSE**

A large number of data warehouse methodologies and tools are available in the market to support the growing industry. None of these methodologies have been considered as a standard DW/BI application development yet. The purpose of this essay is to do a comparative assessment of different methodologies and tools that are commonly available in the market for data warehouse development. After a discussion regarding the state of the art methodologies and tools, we will use the proposed methodology and tool to build a sample data warehouse for analytical analysis. We will use transactional data to explain all stages of methodologies, technologies and discuss all the activities and
strategies of building a standard information database for an organization. We will finally
discuss the risks and limitations of the applied methodology and tools.

RESEARCH PROBLEMS

Modern organizations are drowning in the data, but they lack the necessary data
point analysis to make their business decisions. Most organizations lose a significant
portion of their profit every year because of lack of knowledge regarding the customer’s
behaviour and attitude toward their products. Many companies build an analytical
database, but these databases have not met their needs. This is solely for the reason that the
methodology and tools they used for development were not well designed or customised
for their business intelligence needs.

There are number of important research problems that need better resolutions. These
typically include Integration, Warehouse Specification and Optimization. Even
though the theory of data warehousing offers a lot to business intelligent area, better
solutions are being sought for more flexibility, robustness and efficiency for the future DW
systems.(Widom, Jennifer, 2002).

Though there is a lot of information available on business intelligence
development, it is very difficult to find relevant literature which helps us to decide the
right combination of methodology and tools for an efficient data warehouse development.

SIGNIFICANCE

Due to widespread use of Information Technology, organizations are experiencing
data overload. Identifying and analyzing critical data points is a significant step for an
organization. Without this critical data analysis it is difficult for an organization to make the right decision. To aggregate and integrate this information and make available for managers to take the right decisions we need a well-designed DW / BI system. To build such a solid decision support system we need a well-established DW / BI methodology and tools (Kimball et al., 2008).

In today’s highly competitive business world, the role of a well-established methodology, which has the required functionality, is a critical factor both for the company’s profit and growth. There can be several advantages if a standard DW / BI development methodology and quality tools are used in a development team.

Using a well suited methodology, one can develop an optimal strategy and then use the right tools and technology available in the market to build a business intelligence application for management. Once this task is complete, the business users will be self-motivated to collect the right information which is required for strategic decision making to gain the competitive advantage. Using such a methodology along with the product quality, timelines and awareness of the customer’s trend can perform a significant role not only in the organization’s profit earning, but also in the organization’s future growth.

ASSUMPTIONS

It is assumed that the reader of this essay is familiar with the basic concept of data warehousing and also understands the terminology used in data warehousing. The reader
should also be familiar with the Relational Database management fundamentals like Entity Relational (ER) diagrams and normalization. It also assumed that the reader is familiar with Extraction, Transformation and Load (ETL) processes, its tools and terminologies. This essay discusses two of the methodologies that are available in the business intelligence world, and a comparative assessment is performed. This essay also identifies which of the methodology and tools is useful and adoptable for typical enterprises. After that, it explores a case study to support assumptions regarding existing methodologies and tools that are available in the market.

LIMITATIONS

This research discusses existing methodologies and tools available in the market to be used for Data Warehouse development. It also discusses the methodology adapted by different companies for data warehouse design and implementation. The result of our comparative assessment will provide guidelines and recommendations for a methodology and tools, which can be used in developing a robust business intelligence application.

ORGANIZATION OF ESSAY

- This chapter (Chapter I) provided a brief introduction to the essay subject and define the scope of this project.
- Chapter II – Includes the review of literature on the various data warehouse development methodologies and tools available for creating an enterprise data warehouse.
- Chapter III – Provides detailed explanation of the following two most
common data warehouse development methodologies:

- Dimensional Design Methodology
- Data Driven Methodology

o **Chapter IV** – Provides detailed comparison between the two most common data warehouse development methodologies.

o **Chapter V** – Provides introduction about data warehouse tools in general, evaluation parameters and comparison criteria, technology comparison between SSIS and Informatica and conclusion.

o **Chapter VI** – Presents a comparison between the following two most common ETL tools used for data warehouse development:

  ✓ SQL Server Integration Services (SSIS)
  ✓ Informatica PowerCenter

o **Chapter VII** – Presents a Case Study that compares the impact of two well-known DW development approaches.

o **Chapter VII** – Presents the conclusions, recommendations and suggestions for further research.
CHAPTER II

REVIEW OF RELATED LITERATURE

Data Warehouse

Data warehousing is one of the most critical component of an organization to analyse the past progress of the organization and for future business decisions. Future decision making typically requires a comprehensive view of all aspects of the organization. There is plenty of literature available regarding data warehousing development methodologies, techniques and tools. Hoffer et al. defines data warehouse as follows:

“A data warehouse is a subject-oriented, integrated, time-variant, non-updatable collection of data used in support of management decision-making processes and business intelligence” (Hoffer et al., 2008:422).

In DW development, selection of the architecture has considerable importance but unfortunately very little research exists on this topic. Most of the existing literature provides examples of case-studies or provides survey data (Thilini, & Hugh, 2010)

Data has become the new currency since internet, new technologies and free trade agreements have opened the world for business. Today, it becomes a challenge for organizations to effectively find data, use it, and trust it in order to survive in the competitive market (Erickson, 2009).

Data warehousing is a process in which, enterprises create, maintain and analyze their data through some sort of data warehouse. A data warehouse is not just a
Combination of different operational databases in an organization, it contains specific and unique kinds of databases and its focus is on business intelligence. In other words data warehousing is a process whereby organizations develop and maintain data warehouses for retrieving meaningful information to increase the organization’s profile, and performance in the market. Hoffer (2008:423) describes that the main factor, which triggers data warehousing development, is the recognition of fundamental differences between operational and informational data. These days, data warehousing is the hottest topic in the information systems due to its importance in the market for business growth and development.

Following are the main factors, which force organizations to develop a data warehouse:

- Companywide view of high quality information for business development
- Better management and performance of company’s operational and informational data
- No single book of record
- Data Integrity, Consistency, issues between multiple systems
- Analysis of an organization’s activities in a balanced way

There are two renowned DW design models in the literature:

- **Inmon model**
- **Kimball model.**

The Inmon DW design model relies on a 3NF design foundation in which, data is stored at the atomic level in the DW. This data is further analyzed, integrated and aggregated as per business needs and made accessible across the enterprise. This model is built using the spiral development approach.
On the other hand, Kimball’s approach does not require a normalized data structure before dimensional presentation. This approach structures the data before dimensional presentation with respect to the source data realities (Drewek, 2005).

Inmon suggests the hub-and-spoke architecture (e.g., Corporate Information Factory) while Kimball promotes the data mart bus architecture with conformed dimensions. These two architectures are fundamentally different, and each has strong believers (Thilini, & Hugh, 2010).

Recent surveys show that hub-and-spoke is preferred over bus architecture by margin of 13% by data warehouse implementation teams. In this study, four measures were used to find the success of architecture as listed below:

- Information quality
- System Quality
- Individual Impacts
- Organizational Impacts

(Thilini & Hugh 2010).
While developing a DW, it is very important to make a clear distinction between the characteristics listed in the table 1 and use of the information system. For DW purposes these systems can be categorized into two groups:

✓ Operational Systems
✓ Information Systems

Main characteristics of the two are tabled below.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Operational Systems</th>
<th>Informational Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary purpose</td>
<td>Run the business on the current basis</td>
<td>Support managerial decision making</td>
</tr>
<tr>
<td>Type of data</td>
<td>Current representation of state of the business</td>
<td>Historical point-in-time and predictions</td>
</tr>
<tr>
<td>Primary users</td>
<td>Clerks, salespersons, administrators</td>
<td>Managers, business analysts, customers</td>
</tr>
<tr>
<td>Scope of usage</td>
<td>Narrow, planned, and simple updates and queries</td>
<td>Broad, ad hoc, complex queries and analysis</td>
</tr>
<tr>
<td>Design goal</td>
<td>Performance: throughput, availability</td>
<td>Ease of flexible access and use</td>
</tr>
<tr>
<td>Volume</td>
<td>Many constant updates and queries on one or a few table rows</td>
<td>Periodic batch updates and queries requiring many or all rows</td>
</tr>
</tbody>
</table>

Table 1 - Comparison of operational and informational systems (Hoffer et al., 2008:426)

Hoffer et al. (2008:426) define three basic factors for making operational and informational systems separate as listed below:

- Combine operational data from scattered systems to make them useful for decision support applications.
- Properly structured and designed data improve quality and consistency.
- Data warehouse eliminates contention for resources, when information applications confounded with operational processing.
The basic architectures currently adopted by small and large organizations are:

- Two-level physical architecture for entry-level data warehouse
- Three-level architecture that is increasingly used in more complex environments
- Three-level data architecture associated with a three level physical architecture (Hoffer et al., 2008:426).

Specific areas such as data design, technical design, hardware design and software design are included in the architecture blueprint. These documents and drawings open doors for communication, planning, maintenance, learning, and reuse. With the help of architectural design a schema design strategy for OLTP databases is defined. Several approaches for schema design such as top-down, bottom-up and mix of both exists. These DW architecture design approaches can be used as a classified approach for enterprise-wide data warehouse design and data marts (Arun, & Atish, 2005).

Several components come together in the design of data warehouse architecture. Following is a brief introduction of these components (Kimball et al., 2008:113):

- **Operational Source Systems**

  Operational source system houses and maintains current data in order to manage the real time business activities. It is critical that the operational system process is capable of handling a large amount of transactions resulting in an exceptionally fast response. An operational source system is a storage area that keeps the record of each transaction. Online Transaction Processing Systems (OLTP) is designed for supporting decision support queries or business questions that managers typically
need to address (Atish, & Arun, 2005).

- **Data Staging Area**
  
  This is a temporary area where data is stored after extraction and transformation processes have been applied before it is loaded into a Data Warehouse. Data staging area is used to collect information from different operational databases and file systems. This area is used solely for transformational processes before moving the data into DW. This area is not meant for any kind of data analysis or reporting purposes.

- **Data Mart**
  
  A data warehouse that is limited in scope is generally referred to as “data mart”. Data marts are customized for end users’ decision making activities. Independent ETL processes or data warehouses are used to obtain the contents of data marts. An organization can develop data marts for individual departments like marketing data mart, finance data mart or supply chain data mart. Table 2. shows a comparison between data warehouse and data mart.

- **Metadata**
  
  Metadata is in other words data about data, technical data that illustrates the features or properties of other data. There are three different types of metadata:
  
  ✓ Operational metadata
  ✓ Enterprise data warehouse metadata
  ✓ Data mart metadata
### Characteristics

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Data Warehouse</th>
<th>Data Mart</th>
</tr>
</thead>
</table>
| Scope           | - Application independent  
                - Centralized, possibly enterprise-wide  
                - Planned | - Specific DSS application  
                - Decentralized by user area  
                - Organic possibly not planned |
| Data            | - Historic, detailed and summarized  
                - Lightly denormalized | - Some history, detailed, and summarized  
                - Highly denormalized |
| Subjects        | - Multiple subjects | - One central subject or concern to users |
| Sources         | - Many internal and external sources | - Few internal and external sources |
| Other Characteristics | - Flexible, data-oriented, long life, large, single, and complex structure | - Restrictive, project-oriented, short life, start small, becomes large, multi, semi-complex structures, together complex |

Table 2 - Comparison between DW and DM (Hoffer et al., 2008:470)

**Generic Two-Level Data Warehousing Architecture**

The two levels of this architecture are the source data system and the consolidated data and metadata storage area. Using this architecture for data warehouse requires four basic steps:

1. Data is extracted from the various internal and external source system files and databases.
2. Data extracted from the various source systems is transformed and integrated before loading into data warehouse.
3. The data warehouse is a database organized for decision support. It contains both detailed and summary data.
4. Users can access the data warehouse by means of a variety of query languages and analytical tools.
Independent Data Mart Data Warehousing Architecture

In this type of architecture, we create many independent data marts directly from operational databases. Each data mart represents a subject based data warehouse. Data mart is a data warehouse, which is limited in scope and customized for a specific business group for decision making applications. Each data mart obtains its contents from an independent ETL process.

An organization can build data marts for each of their different departments like a data mart for marketing, one for finance, one for supply chain, etc. in order to support
known analytical processing. When the focus of an organization is on short-term business objectives, it is preferable to create independent data marts. However, designing a data warehouse with the help of multiple data marts results is significant less of flexibility for the long term business decision making.

![Diagram of Independent DM, DW Architecture](Hoffer et al., 2008:430)

Independent data marts by nature are individually built data bases from one another by independent teams. These teams usually use different tools, software, hardware and processes which poses its own set of challenges (Abdolreza et al., 2011).
Dependent Data Mart and Operational Data Store

The most popular approach to address the limitations of an independent data mart is the three-level method presented by dependent data mart and operational data source architecture (Hoffer et al., 2008:430). In this new approach, operational data store, metadata and data storage level are re-designed. Dependent data mart loads from EDW and operational data store provides options for obtaining current data. Operational data store provides different options for obtaining current data to make it available for end users for decision support applications. In the diagram in Figure 3. The dependent data marts still have a reason to provide the transformed high performance environment for decision-making needs of user groups. User groups can access the data marts for regular queries, but when other data is needed, the user should access the enterprise data warehouse (Hoffer et al., 2008:432). The dependent data mart and operational data store architecture is also called a Corporate Information Factory (CIF).
Dependent data marts partially depend on the data warehouse. They load data from a specific business area and provide business information for specific clients who are members of that department. These data marts will later become a part of enterprise data warehouse (Abdolreza et al., 2011).

**Logical Data Mart and Real-time Warehouse Architecture**

The logical data mart and real-time data warehouse architecture are only practical for moderate-sized warehouses or when using high-performance data warehousing technology, such as NCR Teradata system (Hoffer et al., 2008:432). Real-time warehouse architecture has the following unique characteristics:

- Logical data marts are not physically separate databases; but they are more like different relational views of one physical slightly denormalised relational data
Data warehouse.

- Data is moved into data warehouse rather than separate staging areas in order to utilise the high-performance computing power.
- New data marts can be created quickly because it involves no physical storage to be created.
- Data marts are always up to date because data is literally created on the fly.

Figure 4 - Logical DM and real-time DW architecture (Hoffer et al., 2008:433)

Data warehouse experts suggest that the adoption of either top-down or bottom up model supports and in line with characteristics of an organization. Some of these characteristics are:
✓ organization’s decision support requirements
✓ staffing
✓ skills requirements
✓ time to delivery
✓ cost to deploy

(Thilini, & Hugh, 2010).

Experts typically use eight parameters to manage and evaluate the performance of any architectural framework. Table 3. shows a comparison between different data warehouse architecture frameworks with respect to these eight main parameters:

<table>
<thead>
<tr>
<th>DW Architecture</th>
<th>Generic two-level data warehousing architecture</th>
<th>Independent data mart, data warehousing architecture</th>
<th>Dependent data mart and operational data store</th>
<th>Logical data mart and real-time warehouse architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>Local independence</td>
<td>Short-term implementation</td>
<td>Early ROI</td>
<td>Low COI</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>Average</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>High-efficiency</td>
<td>Compatibility with optimal plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>29%</td>
<td>85%</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>87%</td>
</tr>
</tbody>
</table>

Table 3 – A comparison between different DW architectures (Abdolreza et al., 2011)

Data Reconciliation Support Tools

ETL tools provide varying degree of flexibility and functionality in relation to backward compatibility for legacy systems some of which are excellent to extract data from old mainframe systems and while others are excellent to extract data from non-relational database structures. ETL tools read the data from different sources such as
multiple operational databases, flat file systems, ODBC, OLE, and DB. These tools are also capable of defining lookups and other kind of join. Transformation requirements can be straightforward if the source system is relational and developers are knowledgeable. On the other hand, handwritten ETL systems are extremely common because it is relatively quicker and easier to write SQL queries, and then attach these with scripts and use excellent relational engines to start ETL process (Kimball et al., 2008:429).

A good ETL tool must have the ability to read the source files, based on summaries in order to perform calculations on fields including look-ups and the ability to reformat the proper fields (McCabe, & Grossman, 1996).

There are some requirements to fulfill before selecting an ETL development tool from the market. We need to consider the project size, organization budget and the risk associated with making any decision. Overall, the right decision depends on the organization needs and ambitions (Carlos, 2011).

Data transformation is one of the main processes of the data reconciliation process. It makes the source data compatible with the enterprise data warehouse. It takes data from different operational databases, transform and load it into its final destination. This data transformation procedure can be as easy as a small change in data design or can be a highly complicated exercise in data integration. Before attempting to load data into the target system, it must first be clean. Failure to do so will result in erroneous inconsistent data and will ultimately be propagated in the data warehouse (McCabe, & Grossman, 1996).
Kimball et al. (2008:429) provides the following reasons for using industry standard ETL tools for DW development:

- excellent documentation to understand the process
- advanced transformation logic
- improved System Performance
Figure 6 - High level DW/ BI system architecture model (Kimball et al., 2008:114)

In order to make data reconciliation process extremely satisfactory, integrated applications are required. There are many tools available in the market, which assist in developing these kinds of well integrated applications. Hoffer et al. (2008:451) categorises these tools in three categories:

✓ **Data Quality Tools**

✓ **Data Conversion Tools**

✓ **Data Cleansing Tools**

These tools are discussed in detailed below.

Data Quality Tools – These tools are used to analyse the quality of the existing data and to verify that the data fulfills the requirements of the data warehouse. These tools focus on features like profiling, automatic quality code generation, converting data into a common format, verifying data against external data sources, and basic matching of
data between different sources. Hoffer et al. mentions WizRule as “one of these rule discovery tools that searches through records in existing tables and discover the rules associated with the data” (Hoffer et al., 2008:451).

Some commonly used data quality tools are listed below.

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Company</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WizRule</td>
<td>WizSoft</td>
<td>Rules discovery</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.wizsoft.com">www.wizsoft.com</a>]</td>
<td></td>
</tr>
<tr>
<td>InfoRefiner</td>
<td>Computer Associates</td>
<td>Extract, transform, load and index</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.ca.com">www.ca.com</a>]</td>
<td></td>
</tr>
<tr>
<td>DataBridger</td>
<td>Taurus Software, Inc.</td>
<td>Extract, transform, enrich, load</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.taurus.com">www.taurus.com</a>]</td>
<td></td>
</tr>
<tr>
<td>Hummingbird Integration</td>
<td>Hummingbird, Ltd.</td>
<td>Extract, transform, load and index</td>
</tr>
<tr>
<td>Power Center</td>
<td>Informatica</td>
<td>Extract, transform, load and index</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.informatica.com">www.informatica.com</a>]</td>
<td></td>
</tr>
<tr>
<td>Trillium</td>
<td>Harte-Hanks</td>
<td>Quality analysis and data cleansing</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.harte-hanks.com">www.harte-hanks.com</a>]</td>
<td></td>
</tr>
<tr>
<td>Information Quality Suite</td>
<td>FirstLogic</td>
<td>Quality analysis and data cleansing</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.firstlogic.com">www.firstlogic.com</a>]</td>
<td></td>
</tr>
<tr>
<td>Quality Stage</td>
<td>Ascential Software</td>
<td>Family of products for quality analysis, data scrubbing</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.ascential.com">www.ascential.com</a>]</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 - Tools to support data reconciliation (Hoffer et al., 2008:451)

Data Conversion Tools – Data conversion can be a difficult and expensive process if data conversion tools are not utilized properly. Data conversion is the process of making data acceptable from one system to another. This process is also known as transformation. Figure 7-9. shows some examples of data transformations.
Figure 7 - Single-field transformation (Hoffer et al., 2008:451)

Figure 8 - Algorithmes transformation (Hoffer et al., 2008:451)

Figure 9 - Table look-up (Hoffer et al., 2008:451)
Data Cleansing Tools – IBM’s QualityStage, Harte-Hanks’s Trillium, and FirstLogic’s Information Quality Suite are some examples of available tools which are exceptionally good for data quality analysis, data cleansing, and data re-engineering.

ETL tools usually package these functionalities as a set of transforms. Each of these transforms implements a precise data manipulation. The inputs and outputs of these transforms are well-matched so that the transformations can easily be looped together. Some of the usual categories of transforms that come in built-in form with lots of examples in each category includes aggregators, filters, joiners, lookups, sequence generators, sorters and general expression (Beate et al., 2002)

According to (Meyers, 2001) the number one success factor for a data warehousing project is the selection of right Extraction, Transformation and Loading (ETL) tool. In chapter VI, a comparative assessment of ETL tool is given with the help of a case study that can help us find the right tool for DW/BI development. Working with the right tool can help us save up to 70% of the time, which is spent on a typical data warehousing project. Selecting the right tool for data warehousing is an immense challenge because large money is involved in this decision. We should be able to choose the right tool for a project if we carefully review our key requirements against the ETL products available in the market.

ETL tools perform three functions, which all focus on the data moving from the source server to the target server. These three functions are as follows: first, read the data; second, modify or enhance the data as per instructions on the process and finally
write the data into the target database. It is important to note that data cleansing tools are not designed to perform ETL functionality and vice versa. Most of The ETL tools only provide limited data cleansing functionality (Myers, 2001).

During the ETL process, time is the primary concern. Generally, this translates into developing processing tasks that eventually enable the fastest loading of data into the presentation tables and then the fastest end user response times from those tables (Michael et al., 2005).

There are more than eighty ETL tools available in the market that claims to have ETL functionality. Myers (2001) classifies these ETL tools into three categories:

1. Functions
   - Etl Tools – These “small t” tools considered more as data migration tools instead of full ETL processors.
   - eTL or ETl Tools – These “small e” or “small l” tools usually accept specific input or output type and offer fairly healthy transformation functions within the processing engine.
   - eTl Tools – These “capital T” tools perform the data transformation steps relatively better. However, they are not highly efficient to connect too many of the common data formats.
   - ETL Tools - These complete ETL tools provide a rich mix of functionality and connectivity, but are considerably more expensive than the tools we defined in other categories.

2. Engine Type – The engine type classification segments the tools by how the developed ETL processes is executed. The tools are mostly classified in one of the following two
categories: server engine and client engine. The server engine tools allow multiple parallel jobs from more than one developer. It takes benefit of multiple CPUs, which are designed to organize and manage the execution of multiple concurrent routines.

3. Development Environment – Tools in this class are divided into two categories: Code-based tools and GUI-based tools. Code-based tools are extremely common but they are not considered as separate tools because any language can be used for development; for instance, Perl language can be used for code-based ETL tools. Also, we have some other examples of embedded transactional code languages within common database platforms like PL/SQL with Oracle or Transact*SQL with Microsoft SQL Server. On the other hand, GUI-based ETL tools are available in the market for the last 10 years. These tools remove the coding layer for development and provide declarative wizards to do the job. In this way, the user does not need to have expertise of a particular coding language. These kinds of tools also produce some self-documentation regarding the process flow.

Myers (2001) in his article emphasized following 5 c’s while evaluating any tool for DW development after providing a background about availability of different types of tools mentions that before we decide to buy any ETL tool, we should focus on the following five C’s:

- Complexity
- Concurrency
- Continuity
- Cost
- Conformity
Before finalizing an ETL product for a DW project, we need to perform an evaluation and check whether they fulfill the following criteria or not:

- **Integration** – ETL tool should easily integrate into the existing system.
- **Interfaces** – ETL tool must support both target and source systems.
- **Graphical editor** – Provides GUI to model ETL processes.
- **Functionality** – Allow third parties’ libraries to plug-in for processing.
- **Support** – Vendor support and active community forums are available.
- **Documentation** – High quality documentation is available.
- **Up-to-date** – ETL tool should have active developer base for timely future development (Tim et al., 2011).

In the current information technology market, the most popular tools, which most of the organizations are using, for data warehouse development are as follows (Parsad, 2005:49)

- **PowerCenter Informatica** - This product provides a complete BI solution. Important components of this tool are repository manager, designer, workflow manager and workflow flow monitor

- **SQL Server Integration Services (SSIS)** – It is tremendously popular and contains a rich set of development tools like Cognos DecisionStream, Cognos Impromptu, Cognos query, Cognos PowerPlay.

- **SAP Business Object Data Services** – Provides functionality to extract, transform, and load the data from source to target database. It also provides a pre-built report. Data integration here has four components which are
graphical designer, data integration server, metadata repository and administrator.

- **IBM Information Server (Datastage)** – It provides a set of powerful tools for DW development. It contains many components like DataStage manager, DataStage designer, DataStage director and DataStage administrator.

- **Oracle Data Integrator** – Provides high performance data movement and transformation among enterprise platforms. It has a built-in connection to all major databases. ODI used to be more flexible in extracting data from different database technologies and also writing to them.

Enterprise data warehouse or data mart cannot be used without a powerful user interface to access and analyze the data. Varieties of tools are available in the market to analyze and query the data warehouse for business decisions (Hoffer et al., 2008:465).

Hoffer et al. (2008:465) classified these tools as follows:

- **Traditional query and reporting tools** – Spreadsheets, personal computer databases, and report writers are the example of this kind of tools.

- **On-line analytical processing (OLAP), MOLAP, and ROLAP tools** – These are graphical tools which allow users to analyze the data using basic windows techniques.
  - OLAP – is a general term for categories of data warehouse and data mart access tools.
  - ROLAP – are OLAP tools that view the database as a traditional relational database.
- MOLAP – are OLAP tools that load data into an intermediate structure into three or more dimensional array.

- Data mining tools – which are applications that use sophisticated data searching capabilities and allow for application of advanced statistical algorithms for data processing (Weldon, 1996). Following are some of the typical data mining applications available in the market:

<table>
<thead>
<tr>
<th>Type of Application</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profiling populations</td>
<td>Developing profiles of high-valued customers, credit risks, and credit-card fraud.</td>
</tr>
<tr>
<td>Analysis of business trends</td>
<td>Identifying markets with above average (or below average) growth.</td>
</tr>
<tr>
<td>Target marketing</td>
<td>Identifying customers (or customer segments) for promotional activities.</td>
</tr>
<tr>
<td>Usage analysis</td>
<td>Identifying usage patterns for products of services.</td>
</tr>
<tr>
<td>Product affinity</td>
<td>Identifying products that are purchased concurrently, or the characteristics of shoppers for certain product groups.</td>
</tr>
<tr>
<td>Customer retention and churn</td>
<td>Examining the behaviour of customers who have left for competitors to prevent remaining customers from leaving.</td>
</tr>
<tr>
<td>Profitability analysis</td>
<td>Determining which customers are profitable given the total set of activities the customer has with the organization.</td>
</tr>
<tr>
<td>Customer value analysis</td>
<td>Determining if valuable customers are at different stages in their life.</td>
</tr>
<tr>
<td>Up-selling</td>
<td>Identifying new products or services to sell to a customer based upon critical events and lifestyle changes.</td>
</tr>
</tbody>
</table>

Table 5 - Typical data mining applications (Hoffer et al., 2008:470)

- Data Visualization tools – There are some tools available which represents data into graphical and multimedia formats for human analysis. For example:
  - R Project – Statistical Computing
  - iCharts – Creating and presenting compelling charts
✓ WolframAlpha – Computational knowledge engine to draw charts

✓ JQuery Visualize – Draw different kinds of Charts

✓ D3.js – Draws amazing diagrams and charts

✓ Google Charts – Seminal charting solution
CHAPTER III

METHODOLOGY

Dimensional Design Methodology

Kimball et al. (2008:10) defines DW as

“providing information to support business decision making”. A typical data warehouse
gathers all its information from operational databases, also known as “staging area” in data
warehousing terminology. In order to understand the “dimensional design methodology”, we
need to understand two main components of the data warehousing. These components
provide feed to a typical Data Warehouse.

- Data Mart

Data Mart contains a process-centric detailed database that represents a component of the
complete enterprise’s data architecture. Kimball et al. (2008:248) states that
“departmental” and “stand-alone solutions” are considered as a Data Mart.

- Data Staging Area

Kimball et al. (2008:330) states “most ETL systems need a set of staging tables to support
the ETL processes”. This is the area that is typically used to store data after passing
through the cleaning and transformation processes. Once the data has moved into the
staging area, it is considered ready to move into facts tables of Data Warehouse.

Kimball et al. (2008:2) defines that the success of data warehousing depends on the
following three factors:

- Focus on the business
- Dimensional Structuring of data that claims business needs through reports and
  ad-hoc queries
o Develop the data warehouse environment in a manageable lifecycle incrementally rather than building everything in one shot.

Kimball’s Dimensional Lifecycle is illustrated in Figure 10 (Kimball et al., 2008:2, 3).

![Dimensional Model Lifecycle](image)

Figure 10 - Dimensional Model Lifecycle (Kimball et al., 2008:3)

**Managing Project / Program**

In the book, “The data warehouse lifecycle toolkit” (Kimball et al.2008:4), a program is defined as an ongoing coordination of resources, infrastructures, timelines and communication across multiple projects; they resemble a program to an umbrella which encompasses more than one project. They also defined a project as a single iteration of the Kimball lifecycle from start through its complete development and they mention that a project must be finite.

Another point that is important to consider before starting data warehousing development project is to review the readiness of your organization for a DW / BI system Kimball et al. (2008:16). To be confident regarding project success, the management need to consider the following points:
• strong sponsorship and support from management
• compelling business motivation
• feasibility
• positive relation between IT & business

On the other hand, to implement a successful project we need to consider the following activities:

• scope definition
• task identification
• scheduling
• resource planning
• workload assignment

The preliminary scope should be defined after the requirement assessment. In accordance with the Kimball approach management also need to monitor the following activities:

• status monitoring
• issue tracking
• communication plan between IT and business

The Program/Project management process must ensure that the activities defined in the Kimball’s lifecycles are properly implemented. Program/Project management activities focus on monitoring project status, issue tracking, and change control to preserve the scope boundaries (Kimball et al.2008). Kimball’s lifecycle also confirms that ongoing management process includes comprehensive communication plan that define both business and information technology limitations.
**Business Requirements Definition**

An effective business requirements definition is critical as it establishes the foundation for all downstream lifecycle activities (Kimball et al., 2008). This methodology’s approach regarding gathering business requirements is different from traditional ones. DW/BI analysts must understand the key factors driving the business in order to successfully translate the business requirements into design considerations (Kimball et al.2008).

The main theme of Kimball’s lifecycle approach is that DW systems must focus on collecting accurate business requirements. Building DW/BI without a complete and accurate user requirement leads to a project failure. The DW/BI development must have clear directions regarding which requirements are critical and which are not. When an organization has a clear picture regarding the business requirements, it can successfully build data warehouse within their budget and time constraint.

Following are the steps suggested by Kimball’s lifecycle regarding collection of business requirements:

- prepare interview
- conduct interview
- wrap-up interview
- review interview results
- prepare requirements deliverables
- prioritize and agree on next step
- finalize project level requirements
risk analysis

Once all of the business requirements are finalized, different groups in the team can start working on different parts; for example, one group can start working on technical architecture design, another group can start working on dimensional modeling and a different group start working on BI application design.

Technical Architecture Design

In this phase of a project, we identify the existing capabilities of the organization. It is a continuous process as the project moves forward and new business requirements are added. The technical architecture design establishes the overall architectural framework and vision. Factors such as business requirements, current technical environment, and planned strategic technical directions must be considered simultaneously to establish the appropriate data warehouse technical architecture design (Kimball et al., 2008).

Some of the benefits of a solid DW / BI technical architecture and design are as follows:

- Satisfy business requirements – Correct implementation of business requirements is a key for system functionality.
- Communication – This is a source to discuss issues about project with management.
- Planning – A crosscheck for project plan.
- Flexibility, Maintenance and Productivity – Anticipating possible issues and
building a system to handle those issues. It increases productivity and re-use.

- Learning – New team members can easily understand the system.

In the process of architecture design, we must use business requirements as a guide. After completing the architecture design phase, a high-level technical architecture model is elaborated (Kimball et al., 2008:112).

Kimball’s model logically divides the system into two parts: data staging services (Back Room) and query services (Front Room). The query service is the area where the process of data acquisition executed. The data staging area is the area where data extraction, transformation, load and job control works.

The Front Room Architecture contains the following parts (Kimball et al., 2008:115):

- Source Systems – Operational and Operational Data Store (ODS), ERP Systems, User desktops, MDM systems, external supplies, RDBMS, and Flat files
- ETL System (Data Staging area) – The area to hold the data and perform data cleaning and transformation before loading data into DW.
- Presentation Server – An Enterprise Bus Architecture (EBA) adapted dimensional and facts table. Direct queries are executed by the end-users or some time some organizations create views on top of these facts to allow users to run the query. This (EBA) architecture also allows for development of business process data mart.

The second part, which is called query services (front room), contains BI applications. The main purpose of this front room is to provide data warehouse access for users to
perform day-to-day activities (Kimball et al., 2008:115).

The front room architecture provides the following services (Kimball et al., 2008:115):

- **DW Queries & Standard Reports** – User can access the information via ordinary queries, production style and fixed format reports from DW.
- **Analytical Applications** – In addition to normal database queries it contains powerful analytics algorithms.
- **Dashboards** – Interfaces used to view key performance indicators (KPI) textually and graphically.
- **Operational BI** – Real time queries of operational status.
- **Data Mining & Models** – Provided powerful application to perform data analysis.
- **BI Management Services** – metadata services, security services, usage monitoring, query management, enterprise reporting and web portal services.
- **BI Data Stores**– It stores reports, application server which cashes local user databases, disposable analytical data stores, analytical application results, downstream systems and data store security.

**Dimensional Modeling**

Most of the time when data is stored in a data mart, a model named “Star Schema” or “Dimensional Model” is used. This model is different from the relational model. A star schema is a basic database model where dimensional data is separated from fact data. The star schema is designed with two types of tables: Dimensional tables and Fact tables. There are many variations of star schema; for example, model with
multiple fact tables and snowflake schemas that arise when one or more dimensions have a hierarchical structure (Hoffer et al., 2008:471).

![Figure 11 - Components of Star Schema (Hoffer et al., 2008:454)]

Dimensional Modeling Process is extremely iterative and a self-motivated process. It repeatedly requires testing the series of designs against the understanding of the business requirements. Typically three to four weeks is required to design a single business process for a dimensional model. Since dimensional modeling is extremely tedious and time consuming, an experienced dimensional modeller can help an organization to avoid fall starts, dead end paths, and spinning on key decision points (Kimball et al.2008:288).
The dimensional modeling begins with an initial graphical model, pulled from the bus matrix and presented at the entity level. This model is created and critically reviewed in an early set of high level designs that also yields to an initial list of attributes. The last phase of the process is validation. The primary goals of this phase are:

1. To create a model that meets the company’s business model.
2. To verify that the data is available to populate the model.
3. To provide the ETL team with a solid starting point and clear direction (Kimball et al., 2008:322).
Kimball et al. (2008:305) strongly recommended that dimensional modeling should be part of the presentation phase development of the data warehousing development process. They presented four different dimensional design processes (Kimball et al., 2008:246):

- Choose business process – DW designer should first implement only single source data marts to reduce lengthy extraction and then apply these data marts in forms of conformed dimensions so we can later plugin these data marts into the data warehouse bus.

- Declare the grain – When we are defining the grain of the fact tables we need to be exceptionally accurate. The grain should be extremely low to accommodate maximum robust design. Choosing a low-level grain such as individual transactions, individual day snapshots and individual documents line items is advantageous.

- Identify dimensions – Often the grain determines a minimal set of dimensions that is required. More than that, the designers should examine all available resources and attach single-valued descriptors as dimensions.

- Identify facts - the Final step is prudently selecting facts or metrics that are appropriate for the business process.

**Physical Design**

Physical design is the creation of a database with SQL statement. We convert the data gathered during the logical design phase into a description of the physical database structure. During the logical design phase, we defined the data warehouse
and its entities, attributes and relationships. Entities are linked together using relationships. Attributes are used to define entities (Kimball et al., 2008:328).

Kimball et al., (2008:353) recommend the following steps for completing the physical design:

- **Developing standards** – In this step we define naming conventions, staging tables, develop file location standards, use synonyms or view, primary keys and foreign keys.
- **Develop physical data model** – Define the physical structure, finalize source-to-target map.
- **Instantiate relational database** – finalize star schema or snowflake schema, initial sizing estimate.
- **Build the development database** – Prepare a development database before ETL work start.
- **Design processing data stores** – Staging tables to support ETL system, auditing tables, access monitoring tables, security tables.
- **Developing initial index plan** – Indexing and query strategy, indexing dimension tables, indexing fact tables.
Business Intelligence Application Design

After collecting business requirements, some team members work on technical architecture designing and dimensional models while other team members work with the business teams to identify the business intelligence requirements. Business Intelligence applications are the tools for delivering business value from the DW / BI solution, rather than just delivering the data. Most of the time business users just need parameter driven ad hoc reports (Kimball et al.2008).
Kimball et al., (2008:474) defined the importance of business intelligence applications as follows:

- **Be Correct** – BI application must provide accurate audit report.
- **Perform Well** – The average query response time must be less than 5 minutes.
- **Be easy to use** – BI application must be user friendly; that is a user with basic knowledge can easily use the application.
- **Look good** – Reports and other utilities must be clear and attractive.
- **A long-term investment** – Application must be properly documented and enhance for future modification.

Common processes to monitor activities in order to identify a problem are finding the issue, finalizing the solution and then monitoring the result of that solution. There are four steps for a complete analytical cycle of a business intelligence system (Kimball et al., 2008:477):

- Monitor activity
- Identify exceptions
- Determine casual factors
- Model alternatives
As per Kimball et al., (2008:479) there can be six different types of business intelligence applications as listed below:

- direct access to dimensional model
- data mining
- standard reports
- analytical applications
- dashboard and scorecards
- operational business intelligence applications

**Business Intelligence Application Development**

In Kimball’s lifecycle, after completing the “requirements definition” step, senior management would approve to start designing and developing the process of a business intelligence application. For the development tasks, the very first step is the configuration of business metadata and tool infrastructure; followed by constructing
and validating the specified analytical and operational BI applications, finally a navigational portal is designed and constructed (Kimball et al., 2008:505).

Resource planning for a business intelligence application is extremely important because a significant set of resources is required to develop and maintain BI applications (Kimball et al., 2008:506). The business intelligence application development steps are (Kimball et al., 2008:522):

- prepare for application development
- build the application
- test and verify the applications and data
- complete documentation
- plan for deployment

**Product Selection & Installation**

After finalizing two key components of data warehousing i.e. business requirements and architectural design, we can start a general product evaluation process. Timing is extremely crucial in the product evaluation process. If we do not have a clear picture of business requirements we will end up without proper comparison analysis of the product (Kimball et al., 2008:191).

As per Kimball et al., (2008:192) during a product’s evaluation process four cardinal DW/BI areas must be evaluated:
Kimball also simplifies that before taking any decision regarding product, the following evaluations options must be evaluated (Kimball et al., 2008:115):

- Understand the purchasing process – to know how influential ticket items purchased in the organization.
- Develop product evaluation matrix – list of all the required functions and evaluated tools and vendors.
- Conduct market research – search the internet, network and read trade publications; Seek the opinion from industry analysts.
- Narrow down options list – take a look at the scores of each product and narrow down the list.
- Evaluate candidates – listen to their presentation; ask pointed questions; ask each competitor about product’s functionality.
- Recommend and trial product – include anyone who is significantly affected by a product in the decision-making process.

Once the evaluation matrix is completed, the list can be narrowed down to three products. It will be extremely helpful if you can get a trial copy of these products and install and review end-to-end functionality of the product.
Deployment

Deployment characterises the margin of technology, data and end user applications which are accessible from the business users desktop (Kimball et al.2008:541).

As per Kimball there are two steps for DW / BI system deployment:

Pre-Deployment Testing – System test procedures, data quality tests, operations process testing, live testing, performance testing, usability testing, desktop readiness and configuration are some of the phases of the process.

Deployment – Relational DB deployment, ETL deployment, OLAP DB deployment, and report deployment are the second part of this phase.

Maintenance

Once the DW / BI system is in production, technical and operational tasks are necessary to keep the system performing optimally these tasks include usage monitoring, performance tuning, index maintenance, and system backup. Management also needs to focus on the business users with ongoing support, education and communication (Kimball et al., 2008:563).

Regarding maintenance and support processes, Kimball et al., (2008:563) suggested the following steps to manage the front room and back room:

Manage front room - provides user support, maintain BI portal, manage security, monitor usage and report on usage

Manage back room – support data reconciliation, execute and monitor ETL system, monitor resources, manage disk space, tune for performance, backup and recovery and long term archiving
Growth

In Kimball lifecycle, if the team delivered quality product, then the DW / BI system should expand and evolve to deliver more value to the business. In this method unlike the traditional system development initiatives, change should be viewed as a sign of success. Also, priorities must be established to deal with the ongoing business demands. The team can then go to the beginning of the lifecycle to leverage and build upon the foundation that has already been established, while attending to the new requirements (Kimball et al., 2008:579). In this method, a successful DW/BI will grow extremely fast in all dimensions such as volume, structure, depth and utilization of transformed data and system functionality. This will prepare the system for blitz of requests for more data or reports.

Data Driven Methodology

It does not take an application-to-application approach to develop a system. Instead of starting from scratch, the legacy database and processes previously developed form a good base for a newer and better system. To build on previous efforts, the cohesion of data and processing must be recognized. Once recognized, data is built on as if it already exists; if no data exists, then data is constructed so that future development can be built on it. (Juan, Palacio. 2010)

List of processes used for Software and Data Warehouse Development life-cycle:

<table>
<thead>
<tr>
<th>Classical SDLC</th>
<th>Data Warehouse SDLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Requirements gathering</td>
<td>• Implement warehouse</td>
</tr>
<tr>
<td>• Analysis</td>
<td>• Integrated data</td>
</tr>
<tr>
<td>• Design</td>
<td>• Test of bias</td>
</tr>
<tr>
<td>• Programming</td>
<td>• Program against data</td>
</tr>
<tr>
<td>• Testing</td>
<td>• Design DSS system</td>
</tr>
</tbody>
</table>
For data warehouse development, Inmon (2002:34) uses a data-driven methodology and defines a data warehouse as:

- **Subject-oriented** – The data elements from the same real world events or objects are linked together in the database.

- **Integrated** – Data from all the organization’s operational applications are combined in one database in an organized and consistent manner.

- **Time-variant** – Each transaction that is saved into the database is tracked and recorded so the changes can be tracked and reported overtime.

- **Non-volatile** – Once data is loaded into the data warehouse there are no data manipulation performed.

Data-Driven Methodology is presented in three parts. The second part named METH 2 represents DW development steps.
METH 2 – Data Warehouse Development

Data Warehousing includes Business Intelligence (BI) tools to extract transform and load data into the repository. Business Intelligence aims to support better business decision making. Therefore a BI system can be called a Decision Support System (DSS).

Inmon’s Decision Support System development process requires the following procedures to finalize a system:

![Diagram of METH 2](Image)

**Figure 16 - METH 2 (Inmon, 2002:365)**

**Data Model Analysis**

The first task in the development of Decision Support System (DSS) is to perform Data Model analysis. As per Inmon (2002:90) corporate data model can be built without considering the difference between existing operating system and the data warehouse, and this model contains only primitive data. Performance factors are added into the corporate
model, to be transferred into the operational model.

A fair number of changes have been applied to the corporate model to build data warehousing model. As per Inmon, (2002:89) following are the changes applied to the data warehouse model:

- Operational data is removed entirely from data warehouse model.
- The key structure of corporate data model is enhanced with the element of time.
- Derived data is publically used and calculated only one time.
- Artifacts of relationships are created.
The question is how to identify the similarities between the characteristics of attributes. Inmon explains this procedure through a process called stability analysis. Figure 18, shows how grouping of attributes of data collectively performs on the bases of their tendency for change.

![Stability Analysis Diagram](image)

Figure 18 - An Example of Stability Analysis (Inmon, 2002:91)

Data modeling is divided into three levels, first the Entity Relationship Diagram (ERD) which is a higher level of modeling, second the Data Item Set (DIS) which is the
mid-level modeling and the last one is Physical modeling (Inmon, 2002:92).

First hand of modeling is shown with an example below.

The following are four parts that make up the mid-level data model:

- A primary grouping of data
- A secondary grouping of data
- A connector, signifying the relationship of data between major subject areas.
- Types of data
Primary grouping holds attributes that exist only once for each subject area and exists only once for each key area and contains keys. The secondary grouping holds data attributes that can exist more than once, for each main subject area. The connectors relate the data from one grouping to another.

The lowest level model, called physical data model is created from the mid-level data model. This model looks like relational database tables (Inmon, 2002:96).

Figure 21, shows an example of physical model.
As per Inmon (2002: 358) a data model needs to have done the following:

- major subject areas identification
- clearly defined model’s boundaries
- separated primitive data from derived data
- Identification of key attributes and data types

**Breadbox Analysis**

Breadbox analysis is a process used for performing gross estimates of Decision Support System (DSS). If volume of data poses a problem, it is important to know that at the beginning. Breadbox analysis simply is a process to estimate data volume (Inmon, 2002:359).
Technical Assessment

The technical requirements and consideration for managing data and processing in the operational environment is totally different from the technical requirement to manage the data warehousing (Inmon, 2002:360).

For a successful data warehouse, the functionality of a data warehouse must be fulfilled with the following standards:

- Data Warehouse must be able to handle huge volume of data.
- Allow flexibility to access the data.
- Organize the data according to a data model.
- Send and receive data to a wide variety of technologies.
- Ability to have data periodically loaded in bulk.
- Access data set at a time or record at a time.

Technical Environment Preparation

Once architectural configuration for a DW has been established, we technically identify how the configuration can be accommodated (Inmon, 2002:361).

Here is the list of some typical issues, which must be addressed:

- the amount of capacity of Direct access storage device required
- the kind of network connectivity which is required
- the expected volume of processing
- concurrency issues with respect to DB access
- data warehouse can handle the data query traffic volume
Inmon (2002:360) lists the following IT aspects as critical for a successful DW implementation.

- the network
- direct access storage device
- operating system managing DASD
- interfaces to and from data warehouse
- data warehouse management software
- data warehouse

Subject Area Analysis

In this phase, we decide about subject area that we are going to build in the first phase. The subject area must be large enough to be meaningful and small enough to be applicable. If, for any reason, the subject area is too large and complicated, we need to break it into small subsets and select a subset of it to implement. The result of this implementation is a scope of effort in terms of a subject (Inmon, 2002:361).

Data Warehouse Design

The data warehouse design is the main phase of data warehouse development. DW is stored into multidimensional perspective. The physical database of data warehouse is designed based on the data model (Inmon, 2002:362).
Figure 22 - 3 Dimensional perspectives of the entities (Inmon, 2002:139)

In Figure 22 entities representing vendor, shipment, product and customer will be scarcely data populated and entities for orders will be densely populated. The “star join” is a design structure that manages a large amount of data residing in an entity of a data warehouse (Inmon, 2002:139).

In Figure 23, “order” is at the center of the star join. It is the entity of a data warehouse that is heavily populated. Surrounding the star join are the entities of data warehouse such as part, date, supplier and shipment. These entities contain modest number of occurrences of data. The middle entity “order” is called “fact table” and surrounding tables are called “dimension tables”. These are all the components of the physical database design of a data warehouse.
Figure 23 - A simple Star Join (Inmon, 2002:140)

Inmon (2002:361) defines the characteristics of a data warehouse design as follows:

- contains accommodation of different levels of granularity
- placement of data to the major subjects of the corporation
- only allowed primitive data and publically derived data
- have the ability to load mass data periodically
- provide the ability to access a set of data or a record at a time

**Source System Analysis**

After a subject, which is going to be populated, is identified, we identify the source data for that subject in the existing systems. It is absolutely normal to have a variety of data for decision support system (Inmon, 2002:362)

As per Inmon, following criteria can determine the best source of data in an existing environment:
• Which data is the most complete?
• Which data is the timeliest?
• Which data is the most accurate?
• Which data follows the structure of data model?
• Which data is the closest to source of entry?

Inmon (2002:362) suggested that in the source system analysis we must address the following integration issues:
  o How to choose from multiple sources.
  o What to do when there are no sources.
  o What transformations are needed for data that will be transported to the DSS system?

If we follow the above steps properly, then the source system that will feed the data to a data warehouse will be complete, timely, accurate, and easy to access. It will also follow the structure of the data warehouse (Inmon, 2002:363).

Specifications

After successfully completing source system analysis, we will formalize those system analyses in terms of programming specification.

Inmon (2002:363) defined some key issues regarding specifications as listed below:
• Determine which operational data should be scanned
  o Is the operational data time stamped?
o Is there a delta file?

o Are there any systems or log files that can be used?

o Is it possible to change current source code and data structure to create a delta file?

o Are after and before image files rubbed together?

• Once the output is scanned, determine how can we store it by investigating the following issues

  o Is the DSS data pre-formatted and pre-allocated?

  o Is data appended?

  o Is data replaced?

  o Are updates made in the DSS environment?

After completing this step, we will have the actual program specification that will be used to load data from the operational environment to the data warehouse (Inmon, 2002:363).

**Programming**

This step includes all the standard activities of programming such as:

  o development of pseudo code

  o coding

  o compilation

  o walkthrough

  o testing

If this step is done properly, the code completed from this step is efficient,
Population

In this step we populate only a fraction of the data, which is needed in a data warehouse at this point. There may be a need to make some changes to the data. Populating only a small amount of data means that changes can be made easily and quickly. On the other hand, populating a large amount of data greatly diminishes the flexibility of the data warehouse. Once the end user had a chance to look at the data and give feedback to the architect then it is safe to populate large amount of data (Inmon, 2002:286).

Figure 24 shows a feedback loop between DSS analyst and data architect. This feedback loop is vital for the success of data warehouse environment and due to its significance following considerations must be observed.

Figure 24 - Feedback loop between DSS analyst and data architect (Inmon, 2002:286)
• DSS analysts operate quite legitimately.

• The shorter the cycle of feedback loop, the better is for the data warehouse development effort.

• Delay in the feedback loop will increase the size of data to analyze.
CHAPTER IV

DATA WAREHOUSE METHODOLOGIES – A COMPARISON

Methodology Architecture

Making a good choice for data warehouse development methodologies requires thorough understanding of two main data warehousing methodologies, namely bottom-up and top-down approach. Understanding of similarities and differences provides solid foundation knowledge for an organization before applying it.

- Overall Approach

Figure 25 – Inmon’s top-down approach
Inmon’s top-down architectural approach includes information systems and their databases from all departments of an organization. He named this monstrous size of database as Corporate Information Factory (CIF). This approach insures that complete information is consistent because all departmental information originates from a single Atomic DW. (Mary Breslin, 2004:8) (Inmon, 2002:3).

Figure 26 – Kimball’s bottom-up approach
On the other hand, Kimball’s bottom-up approach builds the data marts independently at different times while the business requirements become available from each department. These data marts are later combined and merged into a corporate Data Warehouse. (Mary Breslin, 2004:12) (Kimball et al., 2008:13).

- **Architectural Structure**

Inmon’s architectural structure is organized into four levels:

- operational
- atomic DW
- departmental
- individual

The first level contains daily transactional processing data, and the last three levels become part of a data warehouse, which provides a logical framework for business intelligence and business management capabilities. Inmon believes that the initial efforts to construct atomic data warehouse later helps in the creation of any number of departmental data warehouses without risking data incompatibility between them (Mary
Kimball’s architecture starts from individual department’s data and builds data marts. Then, it uses these individual data marts to build the enterprise data warehouse. In this architecture all data marts are modeled within reliable data standards called adapted dimensions (Mary Breslin, 2004:11).

- **Complexity of the Two Methods**

  Inmon’s top-down method is quite complex in compared to Kimball’s bottom-up approach.

<table>
<thead>
<tr>
<th><strong>Inmon Method</strong></th>
<th><strong>Kimball Method</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>It requires more time and initial investment.</td>
<td>It takes less time and initial investment.</td>
</tr>
<tr>
<td>The design becomes more complicated as soon as the process starts to increase the level of details.</td>
<td>The design becomes simpler as soon as the process starts to increase the level of details.</td>
</tr>
<tr>
<td>More data storage space is required when method store data in a detailed level and due to this, cost also increases.</td>
<td>Less data storage space is required because consistent data standards called conformed dimensions are well defined.</td>
</tr>
</tbody>
</table>

(Mary Breslin, 2004) (Thilini, & Hugh., 2010) (Naveen et al., 2007) (Inmon, 2002:3)

Table 6 – Complexity of the two methods

- **Comparison with established development methodologies**

  The Inmon’s top-down approach uses spiral development methodology, which he calls METH2. This development methodology contains ten decision support systems.
Kimball on the other hands recommends a four-step dimensional design process development methodology.

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSS1 – Modelling steps, completed three-level data models as first input.</td>
<td>Select the business process for the data warehouse project.</td>
</tr>
<tr>
<td>DSS2 – Size / Granularity analysis, measures the detail of the data. If volume is massive, then need to consider multiple levels of granularity for the data. After fixing granularity issues, the first subject area (DSS5) is selected. Decision support 5 to 10 impacts atomic DW and repeats for each subject.</td>
<td>Declare the grain – This process decides what level of detailed data will be contained by data warehouse. The lowest level of granularity is called “atomic”, which means it cannot be further subdivided.</td>
</tr>
<tr>
<td>DSS3 – The team conducts a technical assessment</td>
<td>Choose the dimension – Text-like attributes which are highly correlated with each other. In a retail database we have a product dimension, a store dimension, a customer dimension, a time dimension, etc.</td>
</tr>
<tr>
<td>DSS4 – Preparation of technical environment.</td>
<td>Identify the facts – Determine which facts are to be included in fact tables. Kimball chooses to include some computed values as well as some truly atomic values. This method helps end user in making queries simple and improves the performance to an acceptable level.</td>
</tr>
<tr>
<td>DSS5 – This becomes the first departmental database.</td>
<td>Kimball’s four-step dimensional approach is very simple and more accessible for the end user than spiral development methodology.</td>
</tr>
<tr>
<td>DSS6 – Atomic DW database design begins concurrently.</td>
<td>The bottom-up approach involves a few data elements instead of a data-driven development.</td>
</tr>
<tr>
<td>DSS7 – Analyse the source system of the first subject.</td>
<td></td>
</tr>
<tr>
<td>DSS8 – Write specification.</td>
<td></td>
</tr>
<tr>
<td>DSS9 – Development of pseudo code, coding, compilation, implementation and testing.</td>
<td></td>
</tr>
<tr>
<td>DSS10 – Execution of DSS programs which are previously developed.</td>
<td></td>
</tr>
</tbody>
</table>

Table 7 - Comparison of established development Methodologies

- **Physical Design**

  Top-down approach concentrates more on physical design. It has three levels of data modeling. Bottom-up approach on the other hand, discusses very little regarding physical model. It proposes dimensional data modeling which is different from entity relationship approach.

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>- First is EDR (Entity Relationship Diagrams) as in the development of operational databases. ER Diagrams are used to explore and refine entities, their attributes, and the relationship between entities.</td>
<td>- Kimball’s dimensional data modeling approach is unique in data warehousing.</td>
</tr>
<tr>
<td>- Mid-level of data model establishes DIS (Data Item Set) for each department. It includes three parts:</td>
<td>- The overall architecture features multiple databases that are expected to be highly interoperable.</td>
</tr>
<tr>
<td>- Primary data</td>
<td></td>
</tr>
<tr>
<td>- Secondary Data</td>
<td></td>
</tr>
<tr>
<td>- Connector, signifying relationship of data between major subject areas.</td>
<td></td>
</tr>
</tbody>
</table>

Mid-level data model is designed such that primary grouping exists only once for each major subject. This mean that the ERD created in the first level is the basis for a DIS in the mid-level data model.

- The third-level of data model is the physical data model. The physical model is basically created by
extending the mid-level data model to include keys and physical characteristics of the model.


Table 8 – Physical Design

○ Data Modeling

The two main parts where Inmon’s and Kimball’s data models differ are as follows:

- Data Orientation
- Modeling rules and techniques

Inmon’s logical architecture is an architecture that extracts detailed and time-stamped data from different operational databases and transforms and stores it in a single database called Data Warehouse. The data marts for each department are created from the massive enterprise data warehouse (Mary Breslin, 2004:7). On the other hand, Kimball recommends designing and building a data mart for each department and then transforming these data marts into enterprise data warehouse (Mary, Breslin, 2004:7).

○ Data Orientation

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Inmon’s data modeling approach is data-driven or subject-driven which means the nature of the data directs the data modeling process.</td>
<td>- Kimball’s data modeling approach is process oriented which means data modeling becomes an attempt to define the collaboration of data across the business.</td>
</tr>
<tr>
<td>- Each data mart is based on multiple decision support procedures.</td>
<td>- Each data mart is based on a single business process.</td>
</tr>
</tbody>
</table>


Table 9 – Data Orientation
o **Tools**

Inmon methodology uses traditional tools like High-level Model (ERD), Middle-level Model (DIS) and Low-level Model. The ERD represents the known requirements of the Decision Support System as created by Joint Application Design (JAD). Kimball uses dimensional modeling moving from traditional relational modeling.

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Inmon’s traditional data modeling tools like ERD and DIS are more suitable and applicable for the subject-driven data modeling approach. ERD is used to define highest level of Entity relationship diagrams. DIS is used to further define each entity in the ERD diagram.</td>
<td>• Kimball’s data modeling is different from the traditional approach. It is a new approach in which the process determines which facts and dimensions are important enough to claim a place in the data warehouse.</td>
</tr>
</tbody>
</table>


| Table 10 - Tools |

o **End-user Accessibility**

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>• End users can attend review presentation but few could review ERD and DIS without any assistance unless they are well trained and familiar with technicalities.</td>
<td>• Dimensional Modelling tools allow end users to take an active role in the data modeling process.</td>
</tr>
<tr>
<td></td>
<td>• Kimball’s data mart is highly accessible to the end-user and provides a reasonable query response time.</td>
</tr>
</tbody>
</table>


| Table 11 – End user accessibility |
o Philosophy

Inmon (2002:10) describes a logical architecture as an architecture that extracts time-stamped and detailed data from operational databases and then transforms it and stores it in a single database called data warehouse.

o Primary Audience

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Data-driven approach views IT professionals as the primary developer and provider of the data warehouse.</td>
<td>- Kimball assumed that end-users and IT professionals share roughly equal burden of responsibilities.</td>
</tr>
<tr>
<td>- Inmon believes that the performance of completed data warehouse will be optimized by ensuring a technically oriented development process.</td>
<td>- Bottom-up approach ensures active participation of end-users throughout development process.</td>
</tr>
</tbody>
</table>

(Mary Breslin, 2004) (Inmon, 2002)

Table 12 – Primary Audience

o Place in the Organization

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Integral part of the Corporate Information Factory (CIF).</td>
<td>- Transformer and retainer of operational data.</td>
</tr>
</tbody>
</table>

(Mary Breslin, 2004) (Inmon, 2002)

Table 13 – Place in the Organization

o Objective

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Deliver a sound technical solution based on proven database methods and technologies.</td>
<td>- Deliver a solution that makes it easy for end users to directly query data and still get reasonable response time.</td>
</tr>
</tbody>
</table>

(Mary Breslin, 2004) (Inmon, 2002)

Table 14 - Objectives
o Characteristics

<table>
<thead>
<tr>
<th>Inmon Method</th>
<th>Kimball Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>• This approach helps organizations with strategic decision support.</td>
<td>• This approach helps organizations with tactical decision support.</td>
</tr>
<tr>
<td>• This approach is appropriate if data integration requirements are enterprise wide.</td>
<td>• The approach is good if data integration requirements are department wide.</td>
</tr>
<tr>
<td>• Non-metric data and for data that will be applied to meet multiple and varied information needs.</td>
<td>• Data is structured into business metrics, performance measures and scorecards.</td>
</tr>
<tr>
<td>• If growing scope and changing requirements are critical, then this approach is suitable to implement.</td>
<td>• If DW needs to adapt to highly volatile needs within a limited scope, top-down approach is a better option.</td>
</tr>
<tr>
<td>• If source systems are not stable and there is a high rate of change, then Inmon’s approach is more favourable.</td>
<td>• If source systems are relatively stable, then bottom-up approach is more appropriate.</td>
</tr>
<tr>
<td>• If an organization has a large team of IT Professionals and no issue of staffing, then a top-down approach is preferable.</td>
<td>• Bottom-up approach makes better use of skill shortage, if any, and scarce resources.</td>
</tr>
<tr>
<td>• This approach is good to apply if the organization’s requirements allow for a longer start-up time.</td>
<td>• This approach is suitable, if data warehouse application is urgent.</td>
</tr>
<tr>
<td>• Higher start-up cost with lower subsequent project development costs.</td>
<td>• Lower start-up costs while each subsequent projects cost about the same.</td>
</tr>
</tbody>
</table>


Table 15 – Characteristics

Inmon’s development methodology ensures that end users have a more passive role in the development process, mostly reviewing the results generated by IT professionals; this methodology also results in a sound technical solution which provides very good query response time to end users. The Kimball’s development methodology ensures that
end users are actively involved in the development process and ensures that final product (i.e. data marts) have reasonable query response time. (Beate et al., 2002:213)

Kimball’s approach does not discuss the presence of a physically implemented DW; it does not in fact recognize a need for a central data warehouse. As per Kimball’s approach, raw data is transformed into useful information in staging area (Naveen et al., 2007).

Inmon’s approach considers that the dimensional modeling is for data marts design only and not for the entire DW. As per Inmon, a bottom-up approach is fragile; it only focuses on the end-user requirements and cannot replicate the data which can be more useful on enterprise level. Inmon’s approach focuses on the integration of data from various systems to have a centralized repository (Naveen et al., 2007).

Both methodologies believe that successful performance depends on effective collection of the business requirements. These requirements play an important role in the design of the data marts. Both approaches have a theory of staging; Kimball calls it backroom and Inmon calls it warehouse (Naveen et al., 2007).
CHAPTER V
ETL TOOLS – A COMPARISON

BI Tools - Introduction

Data warehousing development tools are designed to eliminate the need for hand coding. When an organization plans to build a new data warehouse and select the right tools for development, it can save time and money and achieve better results. Building a data warehouse is a complex development effort that requires the right software tools. In the data warehouse development process, we pass through different steps and each step requires different kind of tools to achieve its goal (McCabe, & Grossman, 1996).

Business Intelligence (BI) is considered to have a high impact on businesses and research activities; this has been realized profoundly in the last couple of years. The most important and complicated part of a business intelligence system is to implement extraction, transformation and loading mechanisms. This process takes up to 80% of the effort in a project (Majchrzak et al., 2011).

Gartner (2012) recommended that before deciding to build a DW, an organization needs to find the answer to a few questions like:

- What is the organization’s budget for hardware, software and services from the vendor?
- When would an organization need a DW and how long would the project take?
- Who will use it, build it and maintain it?
• Which ETL tools will be used in order to build the DW?

Business intelligence platform as a software platform deliver different capabilities. These capabilities are organized into three categories of functionalities namely integration, information delivery and analysis (Hagerty et al., 2012).

Following are the categories and their functionalities:

Integration
- BI infrastructure
- Metadata management
- Development tools
- Collaboration

Information Delivery
- Reporting
- Dashboard
- Ad hoc query
- Microsoft office integration
- Search-based BI
- Mobile BI

Analysis
- Online analytical processing (OLAP)
- Interactive visualization
- Predictive modeling and data mining
- Scorecards

The role that an ETL process and tools play in an organization has changed during the last few years. In reaction, vendors are looking to increase the flexibility and functionality
offered by their platforms. Meanwhile, organizations are looking for slots of functionalities by responding to the shift in focus of the market. This has resulted in industry consolidation, both horizontally across data integration and vertically over the data supply chain (Michael & Jamie 2005).

As a business faces collective challenges to incorporate data from a continuously growing number of diverse systems, the software engaged to enable that integration also needs to change to meet those challenges. The following are the business and technical challenges, which organizations are facing (Michael, & Jamie, 2005):

- requiring uninterrupted and real time information
- more regulation
- more sophisticated users
- operational and strategic information requirement
- diversity in sources and targets of data integration
- higher volumes of data
- cost control

**ETL Tools – Evaluation Parameters**

When choosing software, the main focus is to find a user-friendly tool with a solid technology. Because, a user-friendly interface reduces the development time and cost as well as being easy to learn. In our case study, we find that SQL Server Integration Services has a very light and simplified user interface, which makes the learning process less time consuming. On the other hand Informatica has a more complicated interface and will take more time to learn. One thing we realized is that Informatica has comparatively a very
solid technology and ability to address real-time data integration schemes. SSIS was relatively low-cost and has an excellent support and distribution model. On the other hand, Informatica was more expensive but completely focused on B2B data exchange.

Before choosing ETL tool for DW development project, some important points need to be considered such as complexity of the data warehouse infrastructure, functionality, ease of use, performance and price. Building a DW is a complex development process, which requires numerous software tools. BI platform is considered to be a suitable platform if it provides some programming tools, visual development environment, and integrated software development kit, capability to integrate into business process or embedding into another application. The development environment should provide web services while executing common tasks such as scheduling, delivering, administering and managing (Hagerty et al., 2012).

When searching for the right ETL tool for DW projects, it is necessary to adopt a solid judgement regarding the software to buy. It would be helpful if we compare two or three different tools before deciding about which tool is more appropriate for our project (Carlos, 2011:29).

The evaluation parameters we used for ETL tools selection and comparison are as follows:

- **ETL total cost**
- **ease of use**
- **deployment**
After reviewing different products of ETL tools we decided to perform a comparison between the following two ETL tools: Informatica PowerCenter and SQL Server Integration Services (SSIS).

A survey done by (Philip, & Howard 2010) shows that the majority of companies using informatica were using a single product whereas companies using Microsoft SSIS were using multiple products of that same vendor.

ETL simply is a component of business intelligence platform. ETL for data warehousing is simply a technology that is now being used to encompass data integration across the whole organization and to support the following areas (Michael, & Jamie, 2005):

- scalability
- integration of bundled applications
- database integration
- integration support for inheritance systems, relational and non-relational data sources
- support of web services and service alignment
- integration with messaging middleware
- development cooperation
ETL Tools Selection

After going through the above defined selection criteria, Informatica and SSIS were selected as an ETL tool for comparative study assessment. Now we discuss structure and advantages of each tool.

Informatica PowerCenter

The followings are some of the advantages and structure of Informatica PowerCenter ETL development tool (Gartner, 2012):

- Considerable size and resources on the market of data integration tools vendors.
- Solid technology, consistent track record, simple learning curve, ability to address real-time data integration schemes.
- Specifically focuses on ETL and Data Integration topics not only for BI but as a whole.

![Informatica PowerCenter Architecture](image)

Figure 29 – Informatica PowerCenter Architecture

*Informatica* provides the following tools to extract the data from source tables transform it...
and load it into target tables.

**Microsoft SQL Server Integration Services (SSIS)**

The followings are some of the advantages and structure of SSIS an ETL development tool according to Gartner (2012):

- Well organized support, documentation and best performance for DW
- Simple and fast implementation
- Standardized data integration
- Real-time message based capabilities
- Lower cost and outstanding support

- SSIS clients contain the following tools:

![SSIS Architecture Diagram](image)

*Figure 30 – SQL Server Integration Services (SSIS) Architecture*
ETL Tools – Final Conclusion

Initial & Annual Cost

Licensing structure and pricing opinions for software development and maintenance can fluctuate considerably. Initial costs include licensing, implementation, and maintenance. Gartner’s Magic Quadrant Survey which provides a graphical competitive positioning of four types of technology providers in fast-rowing market (Philip, & Howard, 2010). According to Gartner, a three-year total cost of ownership of ETL Informatica is approximately $173K. This number for Microsoft SSIS is approximately $114K (Philip, & Howard, 2010).

Table 15, shows the detailed costs of these two software for licensing, implementation, and maintenance.

<table>
<thead>
<tr>
<th></th>
<th>Informatica</th>
<th>SSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Cost</td>
<td>$41,089</td>
<td>$36,828</td>
</tr>
<tr>
<td>Implementation Cost</td>
<td>$88,679</td>
<td>$62,599</td>
</tr>
<tr>
<td>Maintenance Cost</td>
<td>$44,895</td>
<td>$16,343</td>
</tr>
</tbody>
</table>

Table 16 – ETL three year total cost of ownership

While the cost per project is a useful measure when comparing products, there are other dimensions that can help to define or confirm value and can provide a basis for planning integration in an organization. According to Philip and Howard, the average cost per project for Informatica is $2080.0 and for Microsoft SSIS is $1585.0 (Philip, & Howard, 2010). Therefore, most of the companies buy Microsoft SSIS because of its low licence cost and overall low cost of the ownership.
Ease of Use

Through the case study we implemented, we found that creating a simple workflow in Informatica is more time consuming than in SSIS. This is due to the fact that Informatica stores sources, target and mappings in separate locations and bringing them back in workflow takes more time. Informatica has too many tabs to design a workflow like session, execute workflow, and monitor the process. Scripting in Informatica is less powerful compared with SSIS because it is in Java, while SSIS script editor is the shell of Visual Studio, which provides breakpoints, watch, call stack and step by step debugging.

SSIS has the advantage of having several built-in tasks, which is not available in informatica. However, a couple of functionalities are very impressive in Informatica like adapters, pipeline, XML destination, XML transformation, upsert capability, and visibility of metadata.

Informatica stores everything in a centralized repository; this means that informatica must be connected all the time to the server if a developer needs to perform some development, which is very frustrating. On the other hand, a developer can develop SSIS packages on a local machine in completely disconnected mode. When the code is ready for production, it can be uploaded in the repository or file system.

One of the leading Europe’s independent Information technology research analysis and consultancy organization called Bloor Research conducted a survey on both tools. According to Bloor’s research and survey results 2010, the learning time of the first project in Microsoft SSIS is comparatively less than Informatica PowerCenter. This survey does
not mention the size of the project but they focus on the average number of sources and targets used in each project (Philip, & Howard, 2010).

<table>
<thead>
<tr>
<th>Time to Learn</th>
<th>Resources required to build first solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Weeks</td>
</tr>
<tr>
<td>Informatica</td>
<td>4.2</td>
</tr>
<tr>
<td>SSIS</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Table 17 – Ramp up time and efforts for Informatica and SSIS (Philip, Howard, 2010)

As Figure 31, shows when comparing the average number of sources and targets used in complex projects, Informatics has a higher rank than SSIS (Philip, & Howard, 2010).

![Informatica and SSIS comparison](image)

Figure 31 – Average number of sources and targets per projects (Philip, Howard, 2010).

As per Bloor research survey, informatica requires more resources per week than SQL Server Integration Services (Philip, & Howard, 2010).

<table>
<thead>
<tr>
<th>Component</th>
<th>Conversion</th>
<th>ETL/ADM</th>
<th>B2B</th>
<th>SOA</th>
<th>SaaS</th>
<th>ERP/CRM</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informatica</td>
<td>11.6</td>
<td>12.7</td>
<td>11.9</td>
<td>14.2</td>
<td>16.5</td>
<td>13.4</td>
<td>13.4</td>
</tr>
<tr>
<td>SSIS</td>
<td>7.8</td>
<td>8.2</td>
<td>7.1</td>
<td>12.1</td>
<td>10.0</td>
<td>6.2</td>
<td>8.5</td>
</tr>
</tbody>
</table>

Figure 32 – Average resources spent per project (Philip, Howard, 2010).
SSIS has the best community support in comparison with any other ETL product in the market. There is not a single scenario, which does not have some kind of suggestion or solution available on the network.

**Deployment**

SQL Server Integration Services (SSIS) is a Microsoft product and can be installed and run only on Windows operating system and can run on windows machine only. On the other hand, SSIS can be setup to run by deploying in SQL Server Data Engine or schedule task. Informatica PowerCenter can perform using a server with platforms like Windows, Solaris, HP-UX, IBM-UX, Redhat and SUSE Linux.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>SSIS (SQL Server)</th>
<th>Informatica PowerCenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETL Tool Ownership Cost</td>
<td>LOW</td>
<td>HIGH</td>
</tr>
<tr>
<td>Risk</td>
<td>MEDIUM</td>
<td>LOW</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Heavy UI</td>
<td>Intuitive</td>
</tr>
<tr>
<td>Deployment</td>
<td>Windows Only</td>
<td>Multiple Operating System</td>
</tr>
<tr>
<td>Maintenance</td>
<td>LOW</td>
<td>HIGH</td>
</tr>
</tbody>
</table>

Table 18 – Tools Comparison as per Survey Result (Philip, Howard. 2010)

<table>
<thead>
<tr>
<th>Informatica PowerCenter</th>
<th>SQL Server Integration Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Informatica has excellent commercial data integration suite.</td>
<td>• SSIS also has excellent commercial data integration suite.</td>
</tr>
<tr>
<td>• It was founded in 1993.</td>
<td>• SSIS is part of SQL Server since 1990.</td>
</tr>
<tr>
<td>• As per Gartner Dataquest, It is the market share leader in data integration.</td>
<td>• Microsoft’s BI market share has grown steadily to take the No.3 spot in 2010.</td>
</tr>
<tr>
<td>• It has 2600 customers. Of those, there are fortune 100 companies those are listed in the Dow Jones and governmental organizations.</td>
<td>• Microsoft customers rate its BI platform infrastructure among the highest compared to most other vendors.</td>
</tr>
<tr>
<td>• The company’s sole focus is data integration.</td>
<td>• The company’s main focus is Database visual studio and data integration</td>
</tr>
</tbody>
</table>
• It’s a complete package for enterprises to integrate their systems, cleanse data and connect to a vast number of current and legacy systems.

• Microsoft’s market success is also driven by its IT-oriented, BI authoring tools within SQL Server.

(Gartner 2012), (Carlos, 2011)

Table 19 - Informatica PowerCenter and SSIS comparison
CHAPTER VI

CASE STUDY

CASE STUDY SETUP

In this chapter we compare the impact of two well-known DW development approaches top-down and bottom-up of physical changes in operational database structure and also evaluate the performance of two commonly used ETL tools, Informatica and SQL Server Integration Services. This case study answers the following two questions:

- What is the impact on each model if we modify a single domain?
- What is the impact on ETL development tools due to that modification, in terms of:
  - development time
  - performance
  - maintenance

Virtual University Inc. is a medium size university offering online courses around the globe. University uses multiple systems for student application and registration processes. As per case study requirements we designed the schemas for both systems and populated fictitious data into the tables of both schemas. We built data marts from these two databases and then used these data marts to build Enterprise Data Warehouse.

We designed and built two fictitious transactional databases called Student Application (SA) and Student Registration (SR) to build an Enterprise DW called University Admission Process (UAP). Student Application (SA) database consists of 9 tables to keep track of Student Application and Student Registration (SR) database consists of 12 tables to keep track of Student Registrations data. Microsoft SQL Server 2012 database engine, Microsoft SQL Visual Studio 2010, SQL Server Integration Server, Oracle Database 11g Express Edition,
Informatica 9.1.0 and Oracle SQL Developer 3.2 have been used for the design, extraction, transformation and data loading for Enterprise Data Warehouse.

The following diagrams show the OLTP schemas for storing Student Application and Student Registration data at transaction level of the applications.

Figure 32 - Logical diagrams of Student Application and Registration schemas
The following diagrams show the names of tables and the size of data created in the SA and SR databases. Almost 10M rows are populated in each SA and SR schemas to test ETL tools extraction and loading performance.

Figure 33 – list of SA & SR tables and data size loaded into schema

Test Machine – Basic Information

Windows Edition

Windows 7 Professional Service Pack 1

System

Manufacturer: Fujitsu America, Inc.
Model: FAI Image Version 2.4 for lifeBook T5010
Processor: Intel® Core™2 Duo CPU P8600 @ 2.40 GHz
RAM: 4.00 GB
System Type: 64-bit Operating System

Table 20 – Test case study environment
Case Study Scenario

The following ER diagrams show the OLAP schemas for storing transactional data (OLTP) for SA and SR Data Marts in form of dimensions and facts.

Figure 34 – Case study scenarios
The measurement parameters we considered for methodologies and tools comparison for Data Warehouse design and development were as follow:

- Development Time (per person hours)
- Performance Time
- Maintenance Cost (per person hours)
Development time is the overall time a person takes to design a process using any one of the discussed methodology and tool. Both tools have the option of declarative development including wizards and also drag and drop implementation so very less coding was required. Performance time means which methodology can be implemented in shorter time and which tool can transfer the same amount of data in minimum period of time. In the last metric parameter we tried to see which methodology use less per person hours to add business requirement in the enterprise data warehouse and which tool uses less hour per person to enhance and maintains requirement.

First Kimball’s methodology was used to design different data marts like SA and SR. After that, as per Kimball’s guidelines, Enterprise Data Warehouse was designed using SA and SR data marts. We used Informatica tool to develop a workflow called ETL1 to move data from transactional databases to into SA and SR data marts and then from these data marts into EDW. We also used SSIS to develop a process called ETL3 to implement same functionality we implemented through Informatica to build EDW. We recorded total development hours taken by one person to develop this complete process. In the first ETL (ETL 1) which was developed and run through Informatica to build-up of a new EDW at time t1 seconds is done by transport of operational data. In the second ETL (ETL 3) which was developed and ran through SSIS to build a new EDW at time t3 seconds is done by transportation of operational data.
Next, Inmon’s methodology was used to design EDW using transactional databases of SA and SR. After that, as per Inmon’s guidelines, we designed SA and SR data marts using EDW. We used Informatica tool to develop a workflow called ETL2 to move data from transactional databases to EDW and then built SA and SR data marts from EDW. We also used SSIS to build a process called ETL4 to implement same functionality we implemented through Informatica to build SA and SR data marts. We recorded complete development hours taken by one person to develop this complete process. In the third ETL (ETL 2) which was developed and run through Informatica to build a new EDW at time t2 seconds is done by transport of operational data. In the fourth ETL (ETL 4) which was developed and run through SSIS to build a new EDW at time t4 seconds is done by transport of operational data.

Next step in the process was to modify transactional databases to add couple of attributes in the SA and SR schemas to fulfill the new business requirements. We used Kimball’s and Inmon’s methodologies individually to see how much per person hours is required to adjust these two new attributes in SA and SR data marts and in EDW. We also used Informatica and SSIS to modify existing ETLs to create ETL 5, 6, 7 and 8 to build the data marts and EDW. We recorded complete development hours taken by one person to modify these processes. In the fifth and sixth ETLs (ETL 5, ETL 6) which was developed and run through Informatica to build of a new EDW at time t5 & t6 seconds is done by transport of operational data. In the seventh and eight ETLs (ETL 7 & ETL 8) which was developed and run through SSIS to build of a new EDW at time t7 & t8 seconds is done by
transport of operational data. Complete development, execution and maintenance time is shown in the following tables.

Test Results

**Informatica PowerCenter**

<table>
<thead>
<tr>
<th>Phase1</th>
<th>KIMBALL (BOTTOM-UP)</th>
<th>INMON (TOP-DOWN)</th>
<th>INFORMATICA</th>
<th>Development hours/person</th>
<th>Performance 10 million rows</th>
<th>Maintenance hours/person</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETL1</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>2</td>
<td>3 (1777 Sec)</td>
<td>2</td>
</tr>
<tr>
<td>ETL2</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>2</td>
<td>3 (1923 Sec)</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase2</th>
<th>KIMBALL (BOTTOM-UP)</th>
<th>INMON (TOP-DOWN)</th>
<th>SSIS</th>
<th>Development hours/person</th>
<th>Performance 10 million rows</th>
<th>Maintenance hours/person</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETL5</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>2</td>
<td>3 (2048 Sec)</td>
<td>2</td>
</tr>
<tr>
<td>ETL6</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>2</td>
<td>3 (1987 Sec)</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 21 – Case study outcome data after using Informatica PowerCenter

**SSIS (SQL Server Integration Services)**

<table>
<thead>
<tr>
<th>Phase1</th>
<th>KIMBALL (BOTTOM-UP)</th>
<th>INMON (TOP-DOWN)</th>
<th>SSIS</th>
<th>Development hours/person</th>
<th>Performance 10 million rows</th>
<th>Maintenance hours/person</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETL3</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>3</td>
<td>1 (36545 Sec)</td>
<td>2</td>
</tr>
<tr>
<td>ETL4</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>3</td>
<td>1 (34132 Sec)</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase2</th>
<th>KIMBALL (BOTTOM-UP)</th>
<th>INMON (TOP-DOWN)</th>
<th>SSIS</th>
<th>Development hours/person</th>
<th>Performance 10 million rows</th>
<th>Maintenance hours/person</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETL7</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>3</td>
<td>1 (31046 Sec)</td>
<td>2</td>
</tr>
<tr>
<td>ETL8</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>3</td>
<td>1 (8203 Sec)</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 22 – Case Study outcome data after using SSIS

Development ➔ 1 = Simple <= (4 hours / person), 2 = Medium <= (8 hours / person), 3 = Difficult >= (12 hours / person)

Performance ➔ 1 = Poor, 2 = Good, 3 = Excellent

Maintenance ➔ 1 = Low <= (2 hours / person), 2 = Medium <= (4 hours / person), 3 = Hard >= (6 hours / person)
Conclusion

After successfully completing the case study, answers to the two questions posed earlier are as follows:

• What is the impact on each model if we modify a single domain?

  Both methodologies have negative and positive points but development time to developing EDW both methodologies almost took the same time. Bottom-up methodology took less time to design data marts but more time to design EDW using existed data marts. Top-down methodology took more time to developing EDW but took less time to design data marts from EDW.

• What is the impact on ETL development tools due to that modification, in terms of:

  o development time
  o performance
  o maintenance

  Informatica PowerCenter and MS SQL Server Integration Services (SSIS) are good tools and are heavily used in the DW industry. Development time through informatica is much less than SSIS. Informatica contains many built in functions to use for transformation but SSIS on the other hand requires custom logic, compile and call through visual builder. For example if we need to assign a sequence id to records in the dataset, in Informatica we have a function sequence available that can be called for doing that while for SSIS we have to develop our own algorithms and methods.

  From a time performance point of view, as reported above in table 2, Informatica PowerCenter is much faster than SSIS. In this case study, we made sure to have a test environment that is reasonably fast and compatible with both SSIS and Informatica. However, the results show big difference in performance as Informatica surpassed SSIS in data loading process and processing time.
Throughout this case study, we found also that maintenance is much easier with SSIS than Informatica DataCenter. For example, SSIS provides an integrated environment that consists of one window with multiple tabs to design Control flow, Data Flow, Parameters etc., while Informatica DataCenter provides only stand-alone applications like designer Mapping Architecture, Repository Manager, Workflow Manager, Workflow Monitor etc. In fact, the stand-alone applications take more time to navigate between multiple components.

We presented an evaluation of two well-known DW development approaches with the help of two most popular commercial tools used for Extraction, Transformation and Load processes. After outlining the background of BI and particularly ETL, two tools namely Informatica PowerCenter and SQL Server Integration Services (SSIS) were selected for performance comparison. They are very stable and well-designed commercial tools available for DW development on commercial level. The evaluation involved a variety of experimental runs while monitoring different parameters and collecting data about diverse criteria such as CPU utilization and memory usage. The case study results showed that Informatica PowerCenter is a better designed for big projects with large distributed data sets the requires a significant amount of ETL processing, while SSIS works better for small project with a well-defined or centralized dataset.

Despite all the observations reported here, we still believe that the selection of the relevant product suite is highly dependent on the particular nuances of individual project requirements.
CHAPTER VII

ISSUES, CHALLENGES, AND TRENDS

A complete literature collection regarding modern technologies and tools for data warehouse is the main issue preparing this paper. There are literature available which discusses data warehouse techniques, technologies and methodologies used for development, but there are very few papers available which really discuss different data warehousing development methodologies and tools available in the market, and the comparison between them.

Finding the list of modern methodologies for DW development was a big challenge. As we discussed earlier, there are only two schools of thoughts in this area, which are dimensional design methodology and data drive methodology. These two methodologies exist in the market but with different names, so finding the correct name and entrepreneur of those methodologies is a big challenge. Finding the books which discusses about the modern tools available in the market regarding DW development was another big challenge. I located a couple of books regarding DW/BI development tools, but they discussed only three or four popular products in the market. Therefore, having a compact survey and analysis of different DW techniques and methodologies and their comparison is needed.
CHAPTER VIII

CONCLUSION AND FUTURE WORK

In this paper, our goal is to analyse the modern data warehouse development methodologies and tools, to see how these methodologies works, and what procedures they are best suited for in the data warehouse development lifecycle. We also try to analyse the benefits of existing methodologies and tools, mainly in terms of output. The question, which is the focus of this analysis, is that whether these methods provide benefits to the management of the organization to get the system they need in terms of getting the accurate information to review company performance and make accurate business decisions or not.

Nowadays, data warehousing methodologies are rapidly developing but fluctuate extensively because the field of data warehousing is not truly recognized yet. At the other end, organizations are struggling to get the precise information to measure their performance in different fields of the organization like customer satisfaction, customer service, company’s standing and accomplishments.

After reviewing dimensional design methodology and data driven methodology, I believe that these methodologies have some common procedures ranging from collecting requirements to deploying data warehouse in production. Users from both schools of thoughts praise and recognise each of these methodologies, but organizations are not getting the complete benefits from either of these methodologies. Both methodologies have
some advantages and disadvantages, however due to lack of information; organizations are not getting the exact product they are looking for.

As per my analysis, both methodologies use almost the same methods for DW development lifecycle such as for collecting user requirements. As an instance, Kimball suggests joint application design that is the same in Inmon’s lifecycle. Inmon’s lifecycle also suggests JSD as one of the options to collect the user requirements.

There are some main differences regarding enterprise data in these two methods. Kimball believes in creating small data marts to achieve department level analysis. On the other hand, Inmon believes to get overall business intelligence system from building a single-enterprise-wide data warehouse. However, the procedure to achieve these goals is somehow similar. In conclusion we can say that they have different points of view regarding enterprises-data.

As we see in these methodologies, both agree on providing accurate, timely, and basic data access on enterprise data as a key success factor. In these methodologies, they also suggest that a mart can resolve only some specific needs and therefore, it requires more efforts when we try to convert it to an enterprise-wide solution.

Both approaches agree that the data accuracy is the key issue to handle in the data warehousing development. They each try to explain that in DW development lifecycle in what granularity level we can fix data accuracy issues easier. As per my analysis, through reading different papers, I find that they suggest both methodologies have their own merits.
and demerits. Table 19, shows the main advantage and disadvantage of these two methodologies. In conclusion they suggest defining a hybrid methodology from these two, in order to build a better decision support system.

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Main Advantage</th>
<th>Main Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kimball’s Dimensional Design Methodology</td>
<td>Scaled down project, which means that the ability to demonstrate is more achievable within an allotted time period and budget.</td>
<td>Data marts can potentially be developed by different development teams using different systems; Therefore, the ultimate goal of providing a consistent and comprehensive view of corporate data may never be easily achieved.</td>
</tr>
<tr>
<td>Inmon’s Data Driven Methodology</td>
<td>Potential to provide consistent and comprehensive view of enterprise data.</td>
<td>Larger and complex projects that may fail to deliver value within an allotted time period or budget.</td>
</tr>
</tbody>
</table>

Table 23 - Main Advantages and Disadvantages of Kimball & Inmon's Methodologies

According to this case study, both methodologies are almost the same in terms of the time needed for Data Warehouse development. However, Kimball’s approach has a little edge on Inmon’s approach because of its Data Mart approach. These data marts contain each departmental data, which can be viewed as DW. Therefore, users can start getting benefit faster than Inmon’s top-down approach.

According to my comparative assessment, both approaches produce a useful DW design but Kimball’s approach is faster and easier for creating and implementing a DW. That is because there is no need to design a data warehouse in 3NF. It also provides more balance in terms of centralized and localized flexibility.
As we discussed above building a data warehouse is a complicated and time consuming process. This tedious process requires professional tools, which can fulfill all the required functionalities we need for a data warehouse development. Finding the right tool for data warehouse development is also a challenging process. There are lots of tools available in the data warehousing market. Almost all of the database vendors offer some tools which contains ETL functionality for an extra cost. There are also some other companies apart from DB vendors, which offer excellent ETL tools; however, these tools are comparatively expensive.

Inmon does not discuss any process for finding the right DW development tools. On the other hand, Kimball’s methodology explains in detail the process of finding the right tool with the right price. The process is very self-descriptive and in-detailed, but also very time consuming. Although there are is a large amount of development tools available in the market, it is hard to identify the suitable tools for the DW project. Most of the tools do not provide a complete functionality that is needed for a specific DW development. In this case, organizations try to use different products from different vendors to achieve their goals. Therefore, organizations spend more time and money to finalize the tools.

At the end, the selection of the right ETL tool depends on the project dimension, budget cost and the associated risk. All commercial ETL tools are useful but when we compare these tools we find some qualities and deficiencies in each tool. Finally, the decision about which tool is the most suitable in my opinion depends entirely on the project type and goals. In general, if an organization needs to develop data transformation
and database processes for a small to medium level DW, then SSIS is a good choice, but if the amount of data and information is large and the BI project is more ambitious, then Informatica PowerCenter is a better choice.

An interesting area of research can be formed based on existing methodologies and to review both methodologies and find the best DW development process. It might be better to design a new methodology for DW Development lifecycle too. In this way, organizations can save time and money and will be able to develop a helpful decision support system, which can help them to make the right decisions at the right time for organization’s business development.
REFERENCES


Magazine Communications of the ACM - The disappearing computer CACM Homepage Archive Volume 48 Issues 3, March 2005, Pages 79 - 84 ACM New York, NY, USA


