DATA WAREHOUSING WITH AN OLAP SYSTEM:
A CASE STUDY FOR THESIS OPERATIONS OF THE
SCHOOL OF NURSING, RAMATHIBODI HOSPITAL,
MAHIDOL UNIVERSITY

SRITHONG POLVISES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2005

ISBN 974-04-5975-7
COPYRIGHT OF MAHIDOL UNIVERSITY
Thesis
Entitled
DATA WAREHOUSE WITH OLAP SYSTEM: A CASE STUDY
FOR THESIS OPERATION OF SCHOOL OF NURSING
RAMATHIBODI HOSPITAL, MAHIDOL UNIVERSITY

Miss Sritihong Pelvises
Candidate

Lect. Songpol Ongwattanakul, Ph.D.
Major-Advisor

Asst. Prof. Warakorn Charoensuk, Ph.D.
Co-Advisor

Co-Advisor

Assoc. Prof. Rassmidara Hoonsawat, Ph.D.
Dean
Faculty of Graduate Studies

Asst. Prof. Piya Ratanasuwn, M.Eng
Chair
Master of Science Program in Technology
Of Information System Management
Faculty of Engineering

Copyright by Mahidol University
Thesis
Entitled
DATA WAREHOUSE WITH OLAP SYSTEM: A CASE STUDY
FOR THESIS OPERATION OF SCHOOL OF NURSING
RAMATHIBODI HOSPITAL, MAHIDOL UNIVERSITY

Was submitted to the Faculty of Graduate Studies, Mahidol University
For the degree of Master of Science
(Technology of Information System Management)
on
April 11, 2005

Miss Srithong Polvises
Candidate

Member

Lect. Songpol Ongwattanakul, Ph.D.
Chair

Lect. Supanat Kitcharoensakkul, Ph.D.
Member

Asst. Prof. Warakorn Charoensuk, Ph.D.
Member

Assoc. Prof. Rasmitara Hoonsawat, Ph.D.
Dean
Faculty of Graduate Studies
Mahidol University

Asst. Prof. Piya Ratanzuwan, M.Eng.
Dean
Faculty of Engineering
Mahidol University

Copyright by Mahidol University
ACKNOWLEDGEMENT

This thesis could not have been successfully completed without advisors. They helped me focus on the right track during the development and led me to finish the project on time. My major advisor, Dr. Songpol Ongwattanakul provided me an excellent guidance throughout the project development, and closely review at my thesis documentation and offer many useful suggestions for the improvement. I am deeply grateful for his valuable support.

I would like to extend my sincere gratitude to my co-advisors, Dr. Warakorn Charoensuk and Lect. Kanat Poolsawasd, who provided constructive comments during my thesis time.

In the system evaluation and the system demonstration phase of my thesis, I received a lot of valuable comments from several people: Assoc.Prof. Dr. Yuwadee Luecha, Assoc.Prof. Dr. Panwadee Putawatana, Assoc.Prof. Dr. Orsa Panpakdee, Dr. Wonnapha Prapaipanich, Assoc.Prof. Artit Ungkanont, Asst. Prof. Piya Ratanasuwun, Asst.Prof. Sompong Chunaim, Asst.Prof. Dr.Nuttawan Yoswathana, and Dr. Waressara Weerawat. I am deeply grateful for their invaluable comments and suggestions.

Finally, I would like to express my gratitude to Dr.Supanat Kitcharoensakkul who served as a thesis examination committee and gave me valuable comments to complete my thesis.

Srithong Polvises
DATA WAREHOUSING WITH AN OLAP SYSTEM: A CASE STUDY FOR THESIS OPERATIONS OF THE SCHOOL OF NURSING, RAMATHIBODI HOSPITAL, MAHIDOL UNIVERSITY

SRITHONG POLVISES 4637212 EGTI/M

M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)

THESIS ADVISORS: SONGPOL ONGWATTANAKUL, Ph.D., ASST. PROF. WARAKORN CHAROENSUK, Ph.D., KANAT POOLSAWASD, M.Sc.

ABSTRACT

Information has become a key factor to the success of modern businesses, as a famous quote says, “Information is power”. To gain such power for competitive advantage typically involves a massive amount of data. Data becomes worthless if it is unstructured for modeling, unavailable for accessing, hard for using and difficult for retrieving. Furthermore, the costs of maintaining may grow wastefully. All decisions are made upon information. Therefore, the accuracy of data and the analysis methods become a critical factor to the quality of a decision. At present, Data warehousing (DW) is among the best solutions for gathering and maintaining information. One of the main objectives of DW is to support decision-making. Some aspects of DW which must be satisfied are information that is, subject-oriented, integrated, non-volatile, and time-variant, and finished application that is easily accessed and gives a minimal response time.

The proposed system is a prototype of a data warehouse for the Thesis Operations with a web-based OLAP system: a case study for the School of Nursing Ramathibodi Hospital, Mahidol University. The source data, which relies on multiple platforms and heterogeneous sources, are fetched to a DW database for analysis and summarization, and then presented to the user via internet. The report result leads to better decisions and a better quality of academic Thesis Operations. The main purpose of this system focuses on the methodologies of establishing the DW: Star schema design, ETL process design, OLAP cube creation, web-based OLAP applications with .NET framework, etc. The system was evaluated by high level academic executives. The result revealed that the most satisfaction came from the ease of use and the usefulness (100%). The lowest scores were the accuracy and integrity of the reports, and completeness of functionality (91.2% and 90% respectively). The overall satisfaction of the entire system was 95.1%.

Full implementation of the DW project is a large-scale and highly complex project, which combines various disciplines and techniques in computer science and engineering such as system analysis, software engineering, and database administration. However, the project cannot be completed without good planning, hard working, and self-discipline.

KEY WORDS: DATA WAREHOUSE / OLAP / ETL / STAR SCHEMA / MULTIDIMENSIONAL REPORT

103 P. ISBN 974-04-5975-7
ระบบคลังข้อมูลวิทยานิพนธ์ ด้วยโอแล็บ: การศึกษา สำหรับโรงเรียนพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล (DATA WAREHOUSING WITH AN OLAP SYSTEM: A CASE STUDY FOR THESIS OPERATIONS OF THE SCHOOL OF NURSING, RAMATHIBODI HOSPITAL, MAHIDOL UNIVERSITY)

ศรีทอง พลวิเศษ 4637212  EGTI/M

บทคัดย่อ

มีต่ำกว่าไร้ว่า การมีข้อมูลมาก ทำให้ได้โอกาสในการแข่งขันสูง เป็นพลังสมองและอีกมาก แต่ถ้ามองในมุมกลับ การอีที่มีข้อมูลจำนวนมาก จะทำให้การจัดเก็บข้อมูลที่เป็นระบบที่มีความยากในการจัดเก็บข้อมูลสูงไม่ได้จำเป็น ใช้ประโยชน์ไม่เหนือ พร้อมทั้งมีการนำข้อมูลที่มีความสำคัญมากขึ้นมาแสดงผล ถ้าทำให้เกิดความเสียหายต่อการตัดสินใจได้ระบบคลังข้อมูล คือ ระบบที่เหมาะสมที่สุด ขณะนี้ ที่จะสามารถตอบโจทย์และสนองตอบความต้องการดังกล่าวได้ เพราะวัตถุประสงค์หลักของระบบนี้คือ สนับสนุนการตัดสินใจของผู้บริหาร โดยมีลักษณะพิเศษคือ เป็นการรวมเรียบร้อยข้อมูลที่เกิดขึ้นในหลายๆที่ในหลายๆเรื่องที่สนใจในช่วงเวลาใดเวลาหนึ่ง สามารถค้นหาข้อมูลได้ง่าย เข้าถึงเร็วและทันกาล ระบบที่น่าสนใจนี้ จะเป็นโปรแกรมต้นแบบของระบบคลังข้อมูลและเว็บเบสโอแล็บ โดยใช้ข้อมูลจากกระบวนการที่ขึ้นจากองค์กรที่มีการมาบัณฑิต โรงเรียนพยาบาลรามาธิบดี มหาวิทยาลัยมหิดล ซึ่งเดิมถูกจัดเก็บอยู่ในหลากหลายรูปแบบ มากกว่า สรุปรวม เพื่อให้ได้ผลลัพธ์ที่ดี นำไปใช้ในการวางแผนบริหารจัดการ เพื่อป้องรุนระบาดที่มีระบาดต่อการเจริญเติบโตขึ้น โดยนวัตกรรมระบบจะมุ่งเน้นไปที่ขั้นตอนและกระบวนการในการพัฒนาระบบคลังข้อมูล เช่น วิธีการออกแบบโครงสร้างฐานข้อมูลที่สามารถเพื่อเตรียมความพร้อมในการเจริญเติบโตขึ้นซึ่งเรียกว่า star schema กระบวนการแปลงข้อมูลตั้งจากหลายๆแหล่ง แถลง ออกโอแล็บข้อมูลตั้ง จากข้อมูลที่ได้เรียกว่า ETL การออกแบบและสร้างโอแล็บ (OLAP cube) และสุดท้ายจะทบทวนวิธีการนำเสนอข้อมูลจากโอแล็บด้วยภาษาใน.NET Framework® รวมทั้งแผนการดำเนินงานในการติดตั้งใช้งานจริง ผลการประเมินจากการทดสอบใช้งานระบบของผู้บริหารระดับสูง พบว่า ค่าเฉลี่ยความพอใจโดยรวมมีเท่ากับ 95.1% โดยมีความพึงพอใจสูงสุด (100%) ในเรื่องความสามารถในการใช้งานและเป็นประโยชน์ ซึ่งมีความถูกต้องของข้อมูลและความสมบูรณ์ของระบบมีระดับคะแนนน้อยสุดที่ 91.2% และ 90% ตามลำดับ การทำระบบคลังข้อมูล ทำที่อย่างเป็นรูปธรรมแล้ว จะเป็นงานที่เพิ่มขึ้นขึ้นชัดและใช้ความสามารถศาสตร์ทางคอมพิวเตอร์ทุกขณะจะทำให้สามารถประยุกต์ใช้ได้อย่างง่าย สภาพที่ดีที่สุดถือที่จะไม่แล้วเสร็จ ถ้าขาดการวางแผนกระบวนการการทำงานทุกขั้นตอนอย่างเป็นระบบ ระเบียบ
CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT (ENGLISH) iv
ABSTRACT (THAI) v
LIST OF FIGURES viii

CHAPTER

I INTRODUCTION
   Introduction and background 1
   Problem statement of existing system 3
   Objective 4
   Scope of study 4
   Expected results 5

II LITERATURE REVIEW
   Definition of Data Warehouse 6
   Data Warehouse architecture and components 8
      Overall architecture 9
      Source system 10
      Staging area 11
      ETL 12
      Data Warehouse database 13
      Database model 13
      Surrogate key 16
      Information delivery 18
      Data mart 19
   Methodology of Data Warehouse 20
CONTENTS (cont.)

Database model, management and implementation 23
OLAP 24
Data Warehouse and the web 29
Related research 31

III METHODOLOGY
Project planning and control 33
Gather requirement and data analysis 33
Design data warehouse architecture (Logical) 34
Develop data warehouse (Physical) 34
Testing 35
Implementation and Testing 35
Tools of studying 35
Research schedule 36

IV EXPERIMENT AND RESULT
Analyzing the requirement and sources data 38
Logical and physical dimensional database design 41
Design data transfer structure 48
ETL program 49
Building OLAP cube 56
Building web-base OLAP application 62

V DISCUSSION 68
VI CONCLUSION 76
REFERENCE 81
APPENDIX 84
BIOGRAPHY 103
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 DW architectural and components</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Star schema</td>
<td>14</td>
</tr>
<tr>
<td>2.3 Snowflake schema</td>
<td>15</td>
</tr>
<tr>
<td>4.1 6 Steps of experiment</td>
<td>37</td>
</tr>
<tr>
<td>4.2 Example of Multidimensional report requirement for Statistics of student who are in Research process</td>
<td>39</td>
</tr>
<tr>
<td>4.3 Example of Multidimensional report requirement of Statistics of advisor and ratio of student they are holding in the past (cover 5 years) and present</td>
<td>40</td>
</tr>
<tr>
<td>4.4 A set of entity relationship before combining into a single ERD of DW database</td>
<td>42</td>
</tr>
<tr>
<td>4.5 The entity relationship diagram of DW database</td>
<td>43</td>
</tr>
<tr>
<td>4.6 Physical design of actual database</td>
<td>44</td>
</tr>
<tr>
<td>4.7 The User Interface (UI) of table creation in SQL Server 2000</td>
<td>45</td>
</tr>
<tr>
<td>4.8 Example of Star schema</td>
<td>46</td>
</tr>
<tr>
<td>4.9 Example code for generation the date of the time dimension table in DW database</td>
<td>47</td>
</tr>
<tr>
<td>4.10 The architecture of preparing plan for ETL process.</td>
<td>48</td>
</tr>
<tr>
<td>4.11 Example of SQL statement for data cleansing</td>
<td>50</td>
</tr>
<tr>
<td>4.12 An ERD for code preparing of the fact transferring</td>
<td>51</td>
</tr>
<tr>
<td>4.13 Example of pseudo-code for SQL statement of the fact data transferring</td>
<td>51</td>
</tr>
<tr>
<td>4.14 DTS package for table creation and loading data to Staging area</td>
<td>52</td>
</tr>
<tr>
<td>4.15 DTS package for loading data to DW and OLAP database</td>
<td>53</td>
</tr>
<tr>
<td>4.16 Architecture of the first data transfer to DW database and use during experiment</td>
<td>55</td>
</tr>
<tr>
<td>4.17 Architecture of update transferring in periodic run</td>
<td>55</td>
</tr>
</tbody>
</table>
**LIST OF FIGURES (cont.)**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.18</td>
<td>Example SQL statement for a single view of data source for a cube creation of advisee in process report</td>
<td>57</td>
</tr>
<tr>
<td>4.19</td>
<td>Graphic User Interface (GUI) of OLAP tool</td>
<td>58</td>
</tr>
<tr>
<td>4.20</td>
<td>User Interface of cube creation in design mode</td>
<td>59</td>
</tr>
<tr>
<td>4.21</td>
<td>Design a storage type of a cube</td>
<td>60</td>
</tr>
<tr>
<td>4.22</td>
<td>User Interface of browsing data in OLAP engine</td>
<td>61</td>
</tr>
<tr>
<td>4.23</td>
<td>Client/Server architecture for web-base OLAP Implementation</td>
<td>63</td>
</tr>
<tr>
<td>4.24</td>
<td>The structure and components of web-base OLAP</td>
<td>64</td>
</tr>
<tr>
<td>4.25</td>
<td>The final web-base OLAP report that provide on WWW and using browser to view as a dynamic web page</td>
<td>65</td>
</tr>
<tr>
<td>4.26</td>
<td>Wide area Network diagram for DW and web-base OLAP implementation</td>
<td>66</td>
</tr>
<tr>
<td>6.1</td>
<td>6 steps of experiment</td>
<td>78</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

Introduction and Background

In modern organizations, especially the IT department, there exists database and an On-Line Transaction Processing (OLTP) system. The OLTP system supports day-to-day business operation and accumulates a huge amount of data. Building an OLTP system depends on the nature of the business process and the IT staff’s knowledge and skills. The database applications become a crucial tool that facilitates the entire business. Without them, the modern business may not survive. They are functionally effective from the design. They gather, store, and process all the business data required to successfully perform the daily operations. They provide online information and produce a variety of reports to monitor and run the business. However, they do not fully support ad-hoc query, analyze and summarize of strategic information, which are useful to executive and the decision-makers.

One of the drawbacks of OLTP system is that it can only provide a basic day-to-day business process. Many transactions are produced everyday and stored in multiple physical locations. The amount of transaction grows rapidly day by day and the business transactions may be stored in different platforms, locations, database models and formats. All data is a very important asset of the organization, as a famous quote, “Information is power”. Million of records of data are worthless if they are stored in an unstructured format which may be unavailable to access, hard to use, and difficult to retrieve. Furthermore the cost of maintenance will grow rapidly and wastefully. Moreover, if the data analysis is inaccurate, it will lead to a poor decision making. At this moment, our assumption is that all organization needs to organize their data for efficiency of retrieval and accessibility. The use of historical data for statistical analysis will improve decision making of modern executives and enable the organization to make decision based on the understanding of the entire system rather than using rough estimation based on personal opinion.
Data Warehouse (DW) is the one of emerging technology that helps to manage data effectively and can deliver information to the decision-maker under reasonable amount of time. The DW concepts are created for the purpose of integrating and organize large amounts of data for the ease of retrieval and analysis. Historical and current data from various sources must be cleansed and verified before it is deposited into the data warehouse. This mechanism is called Extract, Transform and Loading (ETL). The ETL process is critical. It provides rules, constrains, and data calculation before storing to DW database. The DW select only interested and related data to store, analyze, summarize, and deliver to end-user in an attractive way and easy to access with minimum disturbance to normal OLTP traffic.

The characteristic of DW database is difference from OLTP system. Inmon, *a.k.a.* the father of Data Warehousing, describes the warehouse as a subject oriented, integrated, nonvolatile, and time-variant collection of data in support of management decision. Using multidimensional data model, star and snowflake schema is almost always different and much simpler than the schema of an OLTP system. It contains a derived data from OLTP system and other external data. Accessing data in DW database can be done in various ways such as OLAP, data mining, and query reporter tools. Information delivery also provides in various ways such a web-based and host based by using a benefit of client/server architecture.

The *Thesis Operations* is archived data, which is stored on an existing OLTP system. This system contains data about *Thesis Operations* of Master degree student at School of Nursing, Ramathibodi Hospital, Mahidol University since 1992. This database system is considered effective in terms of OLTP. It produces specific reports for monitoring and governing the academic processes, but cannot support ad-hoc summary reports and real-time analysis of data. It also takes much time to generate statistic or summary report and has limited access. If it can provide the executive with easily access and monitor their business. They will improve performance of the thesis progress and quality.
The DW project can overcome all the above problems without interfering the existing system or disturbing the IT staffs. The new separate DW system will gather data from the existing OLTP system. Building DW system is a challenging task. Various computing knowledge and skills will be applied to the implementation of this DW framework. In an enterprise DW project, the process may take a very long time, huge expenses, and several man-months labor. There are two approaches to build a DW system: top-down approach and bottom-up approach.

This project focuses on the bottom-up approach to build the individual warehouse which known as Data Mart because it is;
- faster and easier to implement of manageable pieces, and save cost,
- less risk failure,
- suitable for the mission of pilot project and create prototype in a single business process.

This project contains the description of the process development of the DW project for Thesis Operations. Star and Snowflake schema is used to build DW database model. Data source from School of Nursing, Ramathibodi Hospital, Mahidol University, which stored on RDBMS system and external flat files that covering 5-10 years will be extracted to DW database. OLAP is used as a middle-ware to connect to the DW database by using OLE DB connectivity (a universal data access). Then store data into Multidimensional Online Analysis Processing model (MOLAP). The MOLAP provide a data to web-based client applications in both static and dynamic pages. Users can access and view data through HTTP protocol via internet in visualized (graphic) and textual format. All reports are presented in multidimensional and multilevel view (drill-down, roll-up, slice, and dice).

Problem Statement of existing system

Problems can be summarized as follows,
- Problems with the current statistical report architecture are,
  - accessibility,
  - timeliness,
  - format,
  - integrity.
- Take long time to generate report and limit accessed to IT staff.
- Not ready/easy to use for data analyze and summarization.
- Data store in a single standalone PC.
- Not accessible via internet.
- Existing historical and current data was difficult to understand and hard to use for monitoring business process.

**Objective**

The objective of this project is to build the DW with web-base OLAP system which,

- Start from a single subject of thesis operations data from School of Nursing, Ramathibodi Hospital, Mahidol University.
- Solve the problem individually as they happen and to provide an attractive way for decision-maker to access data and reports.

**Scope of study**

The scope of the project includes,

- Developing a prototype of Data Warehouse with web-base OLAP system composes of:
  - DW database, using Star and Snowflake schema,
  - ETL program process,
  - OLAP server, focus on MOLAP model,
  - WEB application for information delivery, connect to OLAP server and provide data to user on static and dynamic web pages with .NET framework,
  - Report result can be easily accessed via WWW in visualized (graphic) and textual format, including drill-down, roll-up, slice and dice,
  - Implement under LAN, WAN of Mahidol University Computing Network (MUC-net) as client/server architecture.
- Using *Thesis Operations* as a subject for implementation.
- Using data source from OLTP system of School of nursing, Ramathobodi Hospital, Mahidol University to study, which are stored on RDBMS (Microsoft access and Oracle) and flat files.

- Following summarize reports is delivered to user via internet that can drill down, roll-up, slice and dice
  - Statistics of graduated student in each academic year
  - Statistics of student who are in Research process
  - Statistics of advisor and ratio of student they are holding in the past (cover 5 years) and present.

**Expected results**

The results are,

- A prototype of Data Warehouse with web-base OLAP system, which implement on the subject of “Thesis Operations”, running on windows platform at School of Nursing, Ramathibodi Hospital, Mahidol University under LAN, WAN of Mahidol University Computing Network (MUC-net).

- Summarized the multidimensional reports that can be accessed on a web page via internet using HTTP protocol. The report should be simple and is in visualize and textual format, including drill down, roll up, slice and dice.

- Blueprint of the system, system specification, and user manual
CHAPTER II
LITERATURE REVIEWS

This chapter provides the following topics:
1. Definition of Data warehouse,
2. Data warehouse architecture and components,
3. Development methodology of data warehouse,
4. Database model, Database management, and implementation,
5. OLAP definition and its architecture,
6. Data warehouse and the WEB.

1. Definition of Data warehouse

Data Warehouse (DW) is not a software product but it is instead the architecture used for enterprise level of business intelligence information system. DW database is a central of heterogeneous database. A full function of DW solutions consists of complementary products and technology [5, 7, 9].

DW technology supports information management for decision making by integrating data from operational systems and external sources in a single database warehouse. It emerges as an increasingly popular and powerful concept for applying the technology to turn a huge island of data into meaningful information for better business decision [1].

Inmon, known as the father of Data Warehousing, describes the warehouse as a subject oriented, integrated, nonvolatile, time-variant collection of data to support management decision. DW is a database used solely for making meaningful report. This is opposed to traditional data capture or online transaction processing system (OLTP). The DW concept covers different aspects such as architecture, model, and tools. DW contains a collection of data tools daily transaction [10].
- **Subject-oriented**: means that all relevant data about a subject is gathered and stored as a single useful format.

- **Integrated**: Integration is closely related to subject orientation. Data warehouse may have the data from different sources put into a single consistency format. This implies that naming conflicts and problems, such as different units of measure, must be resolved.

- **Non-volatile**: means the data in DW is permanent. When data is loaded into the data warehouse, changes, inserts, or deletes are rarely performed. The loaded data is a transformation that originates in the operational databases. The data warehouse is subsequently reloaded or, more likely, appended on a periodic basis (usually nightly, weekly, or monthly) with new, transformed, or summarized data from the operational databases. Apart from this loading process, the information contained in the data warehouse generally remains static. The property of non-volatility permits a data warehouse to be heavily optimized for query processing.

- **Time-variant data**: means that they maintain both historical and present data. Operational databases, in contrast, contain only the most current, up-to-date data. By comparison, data warehouses contain data that is generally loaded from operational databases periodically and usually maintains for a period of three to five years. This aspect marks a major difference between the two types of environments. Historical information is very importance to decision makers, who often want to understand trends and relationships.

**Various definitions and benefits of data warehouse**

DW is not the same as DSS. DW is a platform which integrated data of improved quality to support many DSS and EIS application. DW improves the productivity of corporate decision making through consolidation, conversion, transformation, and integration of operational data to provide consistent view of an enterprise. The ability of DW [2] is as follows;

- Focus on business process and perform a complete analysis of the process.

- Enable organization to make decision base on understanding of entire system rather than using rough estimate base on incomplete data.
DW is not just a component for enterprise reporting that focuses on delivering aggregated data from database center. DW integrates data from heterogeneous data sources. All data must be cleaned and checked against a certain set of rules before fetching into DW database. After that, OLAP, Data mining and report writer tools are used to retrieve data and generate report according to business requirement. It supports the management decision making [1].

DW is a tool to support data analysis that helps the management staffs to evaluate the business by using the historical and current data from IT department. The heart of DW is the business integration. Well-designed business integration leads to the optimal design structure of DW database. Well-established DW must be based on skillful developers who are capable of design and analyze database with clearly understanding of business process [8, 9, 18].

The DW can also provide data in various formats, such as strategic report, ad-hoc report, GIS, MAP, data mining, for decision making [2].

2. Data warehousing architecture and components

The structure that brings all the components of a data warehouse together is known as the architecture. It contains several factors. Primarily, it includes the integrated data that is the centerpiece. The architecture includes everything for delivering information from data warehouse and is further composed of the rules, procedures, and functions that enable data warehouse to work and fulfill the business requirements. Finally, the architecture is made up of the technology that empowers data warehouse. The architecture provides the overall framework for developing and deploying data warehouse. Therefore, it is a comprehensive blueprint [1, 2].

**DW components provide:**

- Overall Technical architecture
- Source, acquisition, cleanup, and transformation tools
- Data warehouse database
- Metadata
- Access tools
- Query and report tools (OLAP, data mining)
The architecture is not just a set of tools that are required to perform the functions and provide the services. When refer to the data extraction function within one of the architectural components, it is simple to mention the function itself and the tasks associated with that function. Tools are the means to implement the architecture. Therefore, the architecture is chosen first and the tools are second. Selection the tools must be appropriated for the data warehouse architecture. Whenever the architecture is established, the architectural plan should include all related components. The plan also states in detail of all related functions, processes, procedures, and data storages of each architectural component. The architectural plans serve as a blueprint for the design and development. It will also serve as a master checklist for tools selection [1, 2, 18].
Source system

Source data [1] in the data warehouse may be grouped into four categories:

1. Production data

This category of data comes from various existing operational systems of the enterprise based on the information requirements in the data warehouse. While dealing with this data, many variations in the data formats and data reside on different hardware platforms may be found. Consequently, the DW must support all different models of database.

2. Internal data

In many organizations, user may keep their “private” spreadsheets, document, customer profiles, and sometimes even departmental database. This is the internal data, parts of that could be useful in the data warehouse. If an organization does business with customers on a one-to-one basis and the contribution of each customer to the bottom line is significant, then detailed customer profiles with sample demographics are important in a data warehouse.

3. Archived Data

Operational systems are primarily intended to run the current business. In every operational system, the old data is stored in archived files. The circumstances in organization dictate how often and which portions of the operational databases are archived for storage. Some data is archived after a year. Sometime data is left in the operational system databases for as long as five years.

Many different methods of archiving exist. There are staged archival methods. At the first stage, recent data is archived to a separate archival database that may still be online. At the second stage, the older data is archived to a flat file on disk storage. In the next stage, the oldest data is archived to tape cartridges or microfilm and even be kept off the site.
4. **External data**

Most executives depend on data from external sources for a high percentage of the information they use. They use statistics relating to their industry produced by external agencies. They may use market share data from competitors and standard values from financial indicators, for their business to check on their performance.

The purposes of such external data sources cannot be fulfilled by the data available within organization itself. The insights gleaned from production data and archived data are somewhat limited. They give only a picture based on what we are doing or have done in the past. In order to spot industry trends and compare performance against other organizations, we need data from external sources.

**Staging area**

This is the place where all extracted data is put together and prepared for loading into the data warehouse. The staging area [2] is similar to an assembly plant or a construction area. In this area, we examine each extracted file, review the business rules, perform various data transformation functions, sort and merge data, resolve inconsistencies, and clean the data. When data is finally prepared either for an enterprise-wide data warehouse or one of the conformed data marts, the data temporarily resides in the staging area waits to be loaded into the data warehouse repository.

In a large data warehouses, data in the staging area is kept in sequential or flat files. These flat files, however, contain fully integrated and cleaned data in appropriate formats ready for loading. Typically, these files are in the formats that could be loaded by the utility tools of the data warehouse’s RDBMS. The data in the staging area are retained for longer periods. Although the data extraction for loading may be easily obtained form relational database with proper indexes, creating and maintaining these relational databases involves overhead for index creation and data migration from the sources system.

The staging area may contain data at the lowest grain that populates tables containing business measurements. It is also common for the aggregated data to be kept in the staging area for loading. The other types of data kept in the staging area
relate to business database dimensions such as product, time, sales region, customer, and promotional schemes.

**ETL (Data extraction/Transformation/Loading)**

The activities that relate to ETL [1, 10, 12, 24] in a data warehouse are by far the most time-consuming and human-intensive because all activities should be effectively performed to increase data quality and system integrity. Therefore, the entire activities must be performed with great care.

**Data Extraction**

Clearly identify the entire internal data source. Specify all the computing platforms and source files from which data is to be extracted. If we are going to include external data sources, determine the compatibility of our data structures with those of the outside sources. Also indicate the methods for data extraction.

**Data transformation**

Many types of transformation function are needed before data can be mapped and prepared for loading into the data warehouse repository. These functions include input selection, separation of input structures, normalization and denormalization of source structures, aggregation, conversion, and resolving of mission values and conversions of names and address. Examine each data element planned to be stored in the data warehouse against the source data elements and ascertain the mappings and transformations.

**Data loading**

Define the initial load. Determine how often each major group of data must be kept up-to-date in the data warehouse. How much of the updates will be nightly updates? Does the environment warrant more than one update cycle in a day? How are the changes going to be captured in the source systems? Define how the daily, weekly, and monthly updates will be initiated and carried out.
Data warehouse database

The central data warehouse database [2, 4, 33, 34] is a cornerstone of the data warehousing environment. This database is almost always implemented on the relational database management system (RDBMS) technology. However, a warehouse implementation based on traditional RDBMS technology is often constrained by the fact that traditional RDBMS implementations are optimized for transactional database processing. Certain data warehouse attributes such as very large database size, ad hoc query processing, and the need for flexible user view creation including aggregated, multi-link joins, and drill-downs, have become drivers for different technological approaches to the data warehouse database. These approaches include:

- Parallel relational database designs that require a parallel-computing platform, such as symmetric multiprocessors (SMPs), massively parallel processor (MPPs), and/or clusters of uni or multiprocessors.
- An innovative approach to speed up a traditional RDBMS by using new index structure to bypass relational table scans.
- Multidimensional database (MDDBs) that are based on proprietary database technology or implemented using already familiar RDBMS.
- Multidimensional databases are designed to overcome any limitations placed on the warehouse by the nature of the relational data model. This approach is tightly coupled with the on-line analytical processing tools that act as clients to the multidimensional data stores. These tools architecturally belong to a group of data warehouse components jointly categorized as the data query, reporting, analysis, and mining tools.

Database model

Star schema [16]

The star schema is the simplest data warehouse schema. It is called a star schema because the diagram of a star schema resembles a star, with points radiating from a center. The center of the star consists of one or more fact tables and the points of the star are the dimension tables.

A star schema is characterized by one or more very large fact tables that contain the primary information in the data warehouse and a number of much smaller
dimension tables (or lookup tables), each of which contains information about the entries for a particular attribute in the fact table.

A star query is a join between a fact table and a number of lookup tables. Each lookup table is joined to the fact table using a primary-key to foreign-key join, but the lookup tables are not joined to each other. A typical fact table contains keys and measures. A measure is typically a numeric or character column, and can be taken from one column in one table or derived from two columns in one table or two columns in more than one table.

A star join is a primary-key to foreign-key join of the dimension tables to a fact table. The fact table normally has a concatenated index on the key columns to facilitate this type of join.

The main advantages of star schemas are that they:
- Provide a direct and intuitive mapping between the business entities being analyzed by end users and the schema design.
- Provides highly optimized performance for typical data warehouse queries.

![Star Schema Diagram](image)

**Figure 2.2** Star schema[16]

**Snowflake schema**[16]

The snowflake schema is a more complex data warehouse model than a star schema, and is a type of star schema. It is called a snowflake schema because the diagram of the schema resembles a snowflake.
Snowflake schemas normalize dimensions to eliminate redundancy. That is, the dimension data has been grouped into multiple tables instead of one large table. For example, a product dimension table in a star schema might be normalized into a Product table, a Product_Category table, and a Product_Manufacturer table in a snowflake schema. While this saves space, it increases the number of dimension tables and requires more foreign key joins. The result is more complex queries and reduced query performance.

![Snowflake schema](image)

**Figure 2.3** Snowflake schema[16]

**Fact and Dimension table**

**Dimension tables**

Dimension tables [10] represent the business dimensions along which the metrics are analyzed. It contains attributes that describe fact record in the fact table. Some of these attributes provide descriptive information; others are used to specify how fact table data should be summarized to provide useful information to the analyst. Dimension tables contain hierarchies of attributes that aid in summarization. Dimensional modeling produces dimension tables in which each table contains fact attributes that are independent of those in other dimensions. The dimension tables are used to specify how a measure in the fact table is to be summarized.
Fact tables

Each data warehouse or data mart includes one or more fact tables [7, 10, 17, 22]. Central to a star or snowflake schema, a fact table captures the data that measures the organization’s business operations. Fact tables usually contain large numbers of rows, sometimes in the hundreds of millions of records when they contain one or more years of history for a large organization. A key characteristic of a fact table is that it contains numerical data (facts) that can be summarized to provide information about the history of the operation of the organization. Each fact table also includes a multi-part index that contains as foreign keys the primary keys of related dimension tables, which contain the attributes of the fact records. Fact tables should not contain descriptive information or any data other than the numerical measurement fields and the index fields that relate the facts to corresponding entries in the dimension tables.

Surrogate Keys

It is important that primary keys of dimension tables remain stable. It is strongly recommended that surrogate keys [18] be created and used for primary keys for all dimension tables. Surrogate keys are keys that are maintained within the data warehouse instead of keys taken from source data systems. There are several reasons for the use of surrogate keys:

- Data tables in various source systems may use different keys for the same entity. Legacy systems that provide historical data might have used a different numbering system than a current online transaction processing system. A surrogate key uniquely identifies each entity in the dimension table regardless of its source key. A separate field can be used to contain the key used in the source system. Systems developed independently in company divisions may not use the same keys, or they may use keys that conflict with data in the systems of other divisions. This situation may not cause problems when each division independently reports summary data, but it cannot be permitted in the data warehouse where data is consolidated.

- Keys may change or be reused in the source data systems. This situation is usually less likely than others, but some systems have been known to reuse keys belonging to obsolete data. However, the key may still be in use in historical data in the data warehouse, and the same key cannot be used to identify different entities.
Changes in organizational structures may move keys in the hierarchy. This can be a common situation. For example, if a salesperson is transferred from one region to another, the company may prefer to track two things: sales data for the salesperson with the person's original region for data prior to the transfer date, and sales data for the salesperson in the person's new region after the transfer date. To represent this organization of data, the salesperson's record must exist in two places in the sales force dimension table, which is not possible if the salesperson's company employee identification number is used as the primary key for the dimension table. A surrogate key allows the same salesperson to participate in different locations in the dimension hierarchy.

In this case, the salesperson will be represented twice in the dimension table with two different surrogate keys. These surrogate keys are used to join the salesperson's records to the sets of facts appropriate to the various locations in the hierarchy occupied by the salesperson.

The employee's identification number should be carried in a separate column in the table so information about the employee can be reviewed or summarized regardless of the number of times the employee's record appears in the dimension table.

The implementation and management of surrogate keys is the responsibility of the data warehouse. OLTP systems are rarely affected by these situations, and the purpose of these keys is to accurately track history in the data warehouse. Surrogate keys are maintained in the data preparation area during the data transformation process.

**Metadata**

Metadata [1] (Information Catalogue) describes the data that is contained in the data warehouse (e.g. Data elements and business-oriented description) as well as the source of that data and the transformations or derivations that may have been performed to create the data element.
Information delivery

The information deliver [1, 9] component makes it easy for the users to access the information either directly from the enterprise-wide DW, from the dependent data marts, or from the set of conformed data marts. Most of the information access in a DW is through online queries and interactive analysis sessions. Nevertheless, your DW will also be producing regular and ad hoc reports.

In many DW, data also flows into specialized downstream decision support applications such as executive information system (EIS) and data mining. The other more common flow of information is to proprietary multidimensional database for OLAP. In component of information deliver, we may provide query services form the user desktop, from and application server, or from the database itself. This will be one of the critical decisions for architecture design.

Functions and service in information delivery area

• provide security to control information access
• monitor user access to improve service and for future enhancements
• allow users to browse data warehouse content
• simplify access by hiding internal complexities of data warehouse from users
• automatically reformat queries for optimal execution
• enable queries to be aware of aggregate tables for faster results
• govern queries and control runaway queries
• provide self-service report generation for user, consisting of a variety of flexible options to create, schedule, and run reports
• store result sets of queries and reports for future use
• provide multiple levels of data granularity
• provide event triggers to monitor data loading
• make provision for the users to perform complex analysis through online analytical processing (OLAP)
• enable data feeds to downstream, specialized decisions support systems such as EIS and data mining
Access tools

The principal purpose of data warehousing is to provide information to business users for strategic decision making. These users interact with the data warehouse using front-end tools. Many of these tools require an information specialist, although many and users develop expertise in the tools. Access tools divide into five main groups:

- data query and reporting tools
- application development tools
- executive information system (EIS) tools
- OLAP: on-line analytical processing tools
- data mining tools

Data mart

The concepts of data marts are presented as an inexpensive alternative to a data warehouse that takes significantly less time and money to build. However, the term data mart means different things to different people. A rigorous definition of this term is a data store that is subsidiary to a data warehouse of integrated data. The data mart is directed at a partition of data (often called a subject area) that is created for the use of a dedicated group of users. A data mart might, in fact, be a set of de-normalized, summarized, or aggregated data. Sometime, such a set could be placed on the data warehouse database rather than a physically separate store of data. In most instances, however, a data mart is physically separated data storage and is normally reside on a separate database server, often on the local area network serving a dedicated user group. Sometimes the data mart simply comprises relational OLAP technology, which creates highly de-normalized star schema relational designs or hyper-cubes of data for analysis by groups of users with a common interest in a limited portion of the database. In other cases, the data warehouse architecture may incorporate data mining tools that extract sets of data for a particular type of analysis. All these types of data mart sets called dependent data marts. Because their data contents is source from the data warehouse, have a high value because no matter how many are deployed and no matter now many different enabling technologies are used,
The different users are all accessing the information views derived from the same single integrated version of the data. [1]

The point-solution-independent data mart is achieving the goal of rapid delivery of enhanced decision support functionality to end-users. The business drivers underlying such developments include:

- extremely urgent user requirements
- the absence of a budget for a full data warehouse strategy
- the absence to a sponsor for an enterprise-wide decision support strategy
- the decentralization of business units
- the attraction of easy-to-use tools and a mind-sized project

Kimbal, described when designing data marts, the organizations should pay close attention to system scalability, data consistency, and manageability issues. The key to a successful data mart strategy is the development of overall scalable data warehouse architecture; and the key step in that architecture is identifying and implementing the common dimensions.

3. Methodology of data warehouse

Designing a Data Warehouse

Building a data warehouse is a very challenging issue because compared to software engineering it is quite a young discipline and does not yet offer well-established strategies and techniques for the development process [9]. Current data warehouse development methods can fall within three basic groups: data-driven, goal-driven and user-driven. All three development approaches have been applied to the Process Warehouse that is used as the foundation of a process-oriented decision support system, which aims to analyses and improve business processes continuously. The aim is to stabilize a link between the methodology and the requirement domain.

9 steps method in the design of DW [2]
1. choosing the subject matter
2. deciding what a fact table represents
3. identifying and conforming the dimensions
4. choosing the facts
5. storing pre-calculations in the fact table
6. rounding out the dimension tables
7. choosing the duration of the database
8. the need to track slowly enhancing dimensions
9. deciding the query priorities and the query modes

In any event, armed with the follow these steps [2]:

**Step 1 choosing the subject matter of a particular data mart.**

The first data mart build should be the one with the most bangs for the buck. It should simultaneously answer the most important business questions and be the most accessible in terms of data extraction. According to Kimball, a great place to start in most enterprises is to build a data mart that consists of customer invoices of monthly statements. This data source is probably fairly accessible and of fairly high quality. In one of Kimball’s laws is that the best data source in any enterprise is the record of “how much money they owe us.” Unless costs and profitability are easily available before the data mart is even designed, it’s best to avoid adding these items to this first data mart. Nothing drags down a data mart implementation faster than a heroic or impossible mission to provide activity-based costing as part of the first deliverable.

**Step 2 deciding exactly what a fact table record represents.**

This step, according to R. Kimball, seems like a technical detail at this early point, but it is actually the secret to making progress on the design. The fact table is the large central table in the dimensional design that has a multiparty key. Each component of the multiparty key is foreign key to and individual dimension table. In the example of customer invoices, the “grain” of the fact table is the individual line item on the customer invoice. In other words a line item on an invoice is a single fact table record, and vice versa. Once the fact table representation is decided, a coherent discussion of what the dimension of the data mart’s fact table are can take place.
Step 3 identifying and conforming the dimensions.

The dimensions are the drivers of the data mart. The dimensions are the platforms for browsing the allowable constraint values and launching these constraints. The dimensions are the source of row headers in the user’s final reports; they carry the enterprise’s vocabulary to the users. A well architects set of dimensions should be chosen with the long-range data warehouse in mind. If any dimension occurs in two data marts, they must be exactly the same dimensions, or one must be a mathematical subset of the other. Only in this way can two data marts share one or more dimensions in the same application. When a dimension is used in two data marts, this dimension is said to be conforming.

Implementation considerations [2]

To illustrate the complexity of the data warehouse implementation, logical steps needed to build a data warehouse:
- Collect and analyze business requirements.
- Define data sources
- Choose the database technology and platform for the warehouse
- Created a data model and a physical design for the data warehouse
- Extract the data from the operational databases, transform it, clean it up, and load it into the database
- Choose database access and reporting tools
- Choose database connectivity software.
- Choose data analysis and presentation software
- Update the data warehouse
4. Database model, management and implementation

**Multidimensional database model** [2]

A multidimensional database (MDB) is a type of database that is optimized for data warehouse and online analytical processing (OLAP) applications. Multidimensional databases are frequently created using input from existing relational databases. Whereas a relational database is typically accessed using a Structured Query Language (SQL) query, a multidimensional database allows a user to ask questions like "How many product have been sold in each department so far this year?" and similar questions related to summarizing business operations and trends.

A multidimensional database - or a multidimensional database management system (MDDBMS) - implies the ability to rapidly process the data in the database so that answers can be generated quickly. A number of vendors provide products that use multidimensional databases. Conceptually, a multidimensional database uses the idea of a data cube to represent the dimensions of data available to a user. For example, "sales" could be viewed in the dimensions of product model, geography, time, or some additional dimension. In this case, "sales" is known as the *measure attribute* of the data cube and the other dimensions are seen as *feature attributes*. Additionally, a database creator can define hierarchies and levels within a dimension (for example, state and city levels within a regional hierarchy). MDDBMS sometime call Star and Snowflake schema.

**DBMS selection** [4]

- Level of user experience. If the users are totally inexperienced with database system, the DBMS must have features to monitor and control runaway queries. On the other hand, if many of our users are power users, then they will be formulating their own queries. In this case, the DBMS must support an easy SQL-type language interface.

- Types of queries. The DBMS must have a powerful optimizer if most of the queries are complex and produce large result sets. Alternatively, if there is and even mix of simple and complex queries, there must be some sort of query management in the database software to balance the query execution.
- Need of openness. The degree of openness depends on the back-end and front-end architectural components and those, in turn, depend on the business requirements.

- Data loads. The data volumes and load frequencies determine the strengths in the areas of data loading, recovery, and restart.

- Metadata management. If our metadata component does not have to be elaborate, then a DBMS with an active data dictionary may be sufficient. Let your requirement definition reflect the type and extent of the metadata framework.

- Data repository locations. Is our data warehouse going to reside in one central location, or is it going to be distributed? The answer to this question will establish whether the selected DBMS must support distributed databases.

- Data warehouse growth. Business requirement definition must contain information on the estimated growth in the number of users, and in the number and complexity of queries. The growth estimates will have a direct relation to how the select DBMS supports scalability.

5. OLAP (Online analytical processing)

OLAP

Online analytical processing tools. These tools are based on the concepts of multidimensional database and allow a sophisticated user to analyze the data using elaborate, multidimensional, complex views. Typical business applications of these tools include product performance and profitability, effectiveness of a sales program or a marketing campaign, sales forecasting, and capacity planning. These tools assume that the data is organized in a multidimensional model, which is supported by a special multidimensional database (MDDB) or by a relational database designed to enable multidimensional properties. [2]

OLAP has emerged recently as an important decision support system technology. It supports queries and data analysis on aggregated databases built from data warehouses. It is a system for collecting managing processing and presenting multidimensional data for analysis and management purposes.
OLAP is application architecture, not a data warehouse or a database management system (DBMS). Whether or not it utilizes data warehouse, the OLAP becomes architecture that more enterprises are implementing to support analytical applications. Looking at OLAP trend, we can see that the architecture has clearly defined layers and that delineation exists between the application and the DBMS. This delineation has given rise to the next generation of OLAP tools, which provide capabilities that utilize RDBMS technology heretofore found only with specialized MDDBMS technology.[17]

E.F. Codd, the father of the relational model, has formulated a list of 12 guidelines and requirements as the basis for selecting OLAP systems. User should prioritize this suggested list to reflect their business requirement and consider that best match those needs [2];

1. Multidimensional conceptual view. A tool should provide users with a multidimensional model that corresponds to the business problem and is intuitively analytical and easy to use.

2. Transparency. The OLAP system’s technology, the underlying database and computing architecture and the heterogeneity of input data sources should be transparent to users to preserve their productivity and proficiency with familiar front-end environment and tools.

3. Accessibility. The OLAP system should access only the data actually required performing the analysis. Additionally, the system should be able to access data from all heterogeneous enterprise data sources required for the analysis.

4. Consistent reporting performance. As the number of dimensions and the size of the database increase, users should not perceive any significant degradation in preference.

5. Client/server architecture. The OLAP system had to conform to client/server architectural principles for maximum price and performance, flexibility, adaptively, and interoperability.

6. Generic dimensionality. Every data dimension must be equivalent in both structure and interoperability.

7. Dynamic sparse matrix handling. As previously mentioned, the OLAP system has to be able to adapt its physical schema to the specific analytical model that
optimizes sparse matrix handling to achieve and maintain the required level of performance.

8. Multi-user support. The OLAP system must be able to support a work group of users working concurrently on a specific model.

9. Unrestricted cross-dimensional operations. The OLAP system must be able to recognize dimensional hierarchies and automatically perform associated roll-up calculations within and across dimensions.

10. Intuitive data manipulation. Consolidation path reorientation (pivoting), drill-down and roll-up, and other manipulations should be accomplished via direct point-and-click, drag-and-drop actions on the cells of the cube.

11. Flexible reporting. The ability to arrange rows, columns, and cells in a fashion that facilitates analysis by intuitive visual presentation of analytical reports must exist.

12. Unlimited dimensions and aggregation levels. Depending on business requirements, an analytical model may have a dozen or more dimension, each having multiple hierarchy. The OLAP system should not impose any artificial restriction on the number of dimension or aggregation levels.

In addition to these 12 rules, robust production-quality OLAP system should also support: [2]

- Comprehensive database management tools. These tools should function as an integrated centralized tool and allow for database management for the distributed enterprise.

- The ability to drill down to detail (source record) level. This means that the tool should allow for a smooth transition from the multidimensional database to the detail record level of the source relational database.

- Incremental database refreshes. Many OLAP database support only full refresh, and this presents an operations and usability problem as the size of the database increases.

- Structured query language (SQL) interface. An important requirement for the OLAP system to be seamlessly integrated into the existing enterprise environment.
In the OLAP world, there are mainly two different types: Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP). Hybrid OLAP (HOLAP) refers to technologies that combine MOLAP and ROLAP. [12]

**MOLAP**

This is the more traditional way of OLAP analysis. In MOLAP [8, 12, 16, 17, 19], data is stored in a multidimensional cube. The storage is not in the relational database, but in proprietary formats.

**Advantages:**
- Excellent performance: MOLAP cubes are built for fast data retrieval, and is optimal for slicing and dicing operations.
- Can perform complex calculations: All calculations have been pre-generated when the cube is created. Hence, complex calculations are not only doable, but they return quickly.

**Disadvantages:**
- Limited in the amount of data it can handle: Because all calculations are performed when the cube is built, it is not possible to include a large amount of data in the cube itself. This is not to say that the data in the cube cannot be derived from a large amount of data. Indeed, this is possible. But in this case, only summary-level information will be included in the cube itself.
- Requires additional investment: Cube technology are often proprietary and do not already exist in the organization. Therefore, to adopt MOLAP technology, chances are additional investments in human and capital resources are needed.

**ROLAP**

This methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP’s slicing and dicing functionality. In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.
Advantages:
- Can handle large amounts of data: The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on data amount.
- Can leverage functionality inherent in the relational database: Often, relational database already comes with a host of functionality. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionality.

Disadvantages:
- Performance can be slow: Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, the query time can be long if the underlying data size is large.
- Limited by SQL functionality: Because ROLAP technology mainly relies on generating SQL statements to query the relational database, and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL), ROLAP technologies are therefore traditionally limited by what SQL can do. ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions.

HOLAP
HOLAP [8, 12, 16, 17, 19] technologies attempt to combine the advantages of MOLAP and ROLAP. For summary-type information, HOLAP leverages cube technology for faster performance. When detail information is needed, HOLAP can "drill through" from the cube into the underlying relational data.
6. Data warehouse and the web

The web has sharply diminished the cost of delivering information. It is the information delivery and makes cost-effective.

Web-enabled data warehouse

A web-enabled data warehouse uses the web for information delivery and collaboration among users. It’s increase in the knowledge level of the enterprise to more users under the communications infrastructure is already there. Almost of user have web browser. No additional client software is required. We can leverage the webs that already exist. The web-enabled data warehouse takes center stage in the web revolution. The web is everywhere. A large number of software vendors have already made their products web-ready. It’s convergence technologies.

- fast response, although some web pages are comparatively slower
- extremely easy and intuitive to use
- up 24 hours a day, 7 days a week
- more up-to-date content
- graphical, dynamic, and flexible user interfaces
- almost personalized display
- expectation connect to anywhere and drill across

Building a web-enabled data warehouse

The primary purpose of the data warehouse is to provide information, and the internet makes the providing of information easy and wider array of users. Web browser is the key to information delivery. The web interface consists of a browser, search engine, GroupWare, push technologies, home pages, hypertext links, and downloaded Java and ActiveX applets. Tools supporting HTML can be universally deployed. However, for complex analysis, HTML is cumbersome. Use HTML as much as possible and reserve Java or plug-ins for complex ad hoc analysis.
OLAP and the WEB

The next generation of DSS replaced complex mainframe computer with easy-to-use GUIs and point-and-click interface. OLAP capabilities were still limited to a small number of users and OLAP services extend to more than a select group of analysts.

Web-OLAP approach

1. Browser plug-ins; this is just a slightly modified version of fat-client in windows implementation except that the client configuration is more towards that of a thin client. Support issues creep in and this approach has scalability problems.

2. Pre-created HTML documents; providing pre-created HTML documents along with the navigation tools to find these. The documents are result sets of analytical operations. This approach takes advantage of web technology and thin-client economy, but users are confined to using predefined reports. The approach is devoid of demand analysis users cannot do typical online analytical processing.

3. OLAP in the server; the best approach is to use the server to do all online analytical processing and present the results on a true thin-client information interface. This approach realizes the economic benefits of the web and thin-client architecture, at the same time, it provides an integrated server environment irrespective of the client machines. Maintenance is minimized because applications and logic are centralized on the server. Everyone shares the same components; server, metadata, and reports. This approach works well in production environments.
Related research

Data Warehouse and World Wide Web

Suvakhon Sindhuvanich, graduate student of Faculty of Engineering, Mahidol University, successfully built a DW for a case study of agricultural census statistic by using Online Analysis Process (OLAP) on the World Wide Web (WWW). Vast volumes of data gathered from Agricultural Censuses and Intercensus surveys of agriculture, which is normally stored in tapes causes a problem of information retrieval. The current data warehouse technique and multidimensional analysis concept were applied to the design. It encompassed fact and dimension tables. The fact tables contained keys for the dimension table and numerical data, and the dimension tables contained attribute data which was used for the analysis part. For the database model, the simple star and snowflake schemes were utilized. MS SQL Server 2000 software was used to create the data warehouse, and Excel PivotTable was used to link with the OLAP cube in order to present dynamic reports on the WWW. FrontPage 2000 was used as originating software platform for developing web pages. The result of this study is that the data warehouse model is useful for the statistical agricultural data, especially with the staple product: rice. It can generate various reports to users. This data warehouse model is flexible for analyzing purposes, for instance allowing change of dimensions, displaying both drill-up and drill-down types of reports. [13]

Data Warehouse in a statistical environment

Kuntaro Chumnumgih, Faculty of Engineering, Mahidol University developed a data warehouse in a statistical environment: labor force survey case study. This case study is concerned with the creation of a new conceptual database; termed data warehouse to provide information accessibility across different database platforms, using Labor Force Survey data. The building of the data warehouse aims to prepare the most required data into storage for delivery to the end user. The data warehouse that stores the statistical data is the same as that developed for business use. The application tool that allows user access data in the warehouse is the On-Line Processing (OLAP) tool. The OLAP tool was designed to support data warehouse
structure and access information across different dimensions or different levels of data in each dimension. The data warehouse provides easy and convenient of statistical data when compared to the old traditional method of data access. Furthermore, user can apply the data warehouse results to a decision support system. [14]

Data Warehouse framework

Somjintana Srettha (Faculty of Science Mahidol University) developed a three-dimension data warehouse framework (3-D DWFW). The main purpose of this research is to design a new framework for data warehousing project management. This framework results from a combination of three basic components of data warehouse, a system development life cycle, and a project management. Such a framework is expected to be used as a project management guideline for a data warehouse system development. Another purpose of this research is to design and develop a tool that supports a management of a data warehousing project according to the proposed framework’s concept. Her 3-D DWFW is against the Capability Maturity Model (CMM) standard. This research contributes the framework and its methodology which can nicely demonstrate the complexity of the data warehouse system in order it is appropriate for planning, monitoring and controlling the complex data warehouse project with effectiveness and efficiency. [15]
CHAPTER III
METHODOLOGY

System Development Life Cycle (SDLC) methodology, which is the most widely used in software engineering and data warehouse methodology, is used in this project to gain effective control over the entire process of implementation. The SDLC and every step inside are described as follows;

**Project planning and control**
- Defining the DW project
- Project scope
- Estimate techniques
- Project schedule
- Project plan activities

**Gather requirement and data analysis**

Study the existing system and determine the feasibility of DW environment. Identify data elements, and sources by:
- Interviewing executive and data owner to obtain requirements, activities and summary reports.
- Analyze the structure of existing OLPT system and its source.
- Select data from various sources required in DW project.
- Define specification of data model in DW database, OLAP server, and web application architecture.
Design Data Warehouse architecture (Logical)

- Design DW database which compose of;
  1. Fact table, which contains measures and numeric keys.
  2. Dimension tables, which contain detailed data and provide to each numeric key in fact table.
- Design the ETL activities and the process flow.
- Place constraints on the data in ETL process.
- Design OLAP cubes and define data storage type.
- Design web application architecture, which will interface with OLAP server.
- Design the end user interface (web page) in both static and dynamic including roll-up, drill-down, slice and dice.

Develop Data Warehouse (Physical)

- Convert the logical data model from design phase to physical database model of DW and OLAP cube.
- Prepare source data for transformation and loading.
- Extract, Transform and Loading by creation process to mapping all data elements from the source systems to the data warehouse tables.
- Load the data to DW database at once and periodic.
- Create OLAP cube by deriving data from DW database and restoring in MOLAP mode.
- Develop static and dynamic web page for delivery report result to user by using ASP.net programming language with ActiveX component to access OLAP server.
- Set Internet Information Services (IIS) web server to run a web application.
- Implement system under LAN, WAN of Mahidol University Computing Network.
Testing

Testing system consist of;

- **Unit test**: each step of system process such as ETL process, DW database design, OLAP cube creation, and web interface will be stress tested and debugged for error.

- **Integrate test**: all components are pieced together and the functional test such as report result and network connection is started. This test involves user integrity and completeness of functionality.

Implementation and Testing

This step is a test for a full deployment of the system. The testing process covers functionality, communication, integration, and interfacing tests to find out any errors in all parts of the system. An evaluation form should be established in this phase to measure the user satisfaction.

Tools of studying

**Hardware**

- CPU: Pentium4 1.8GHz
- Ram: 256MB
- Hard disk: 80 GB
- Monitor: Super VGA (AGP 32MB)
- Peripheral: Keyboard, mouse USB

**Software**

- Operating system: Microsoft windows 2000 advance server
- Database engine: Microsoft SQL server 2000
- ETL tools: DTS
- OLAP tools: Microsoft SQL server analysis service 2000
- Web interface tools: Microsoft office web component, notepad, and Visual studio .NET
- Programming language: HTML, Scripting Language, and ASP.net
- Web browser: Internet explorer and Netscape navigator
- Document generator: Microsoft office, Note pad, and Case tool

Research schedule

<table>
<thead>
<tr>
<th>Tasks and Activities</th>
<th>Time (month)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Project planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Gathering requirement and data analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Design Data Warehouse architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Develop Data Warehouse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Testing and Implementation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Conclusion and document</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER IV
EXPERIMENT AND RESULT

This chapter gives the detailed of experiment and its result. This project is involving several phases of development and can be separated into 6 steps. All issues in each step are essential to build a Data Warehouse (DW) and a Web-base On-Line Transaction Processing (OLAP) system. Each step will give the detailed design of creation of the DW and the expected result. There is loop feedback between each step and can performed repeatedly in a cycle until the system is completed. Several steps will be covered throughout this chapter including:

1. Analyzing the requirement of source data
2. Logical and physical dimensional database design
3. Design Data transfer structure (data flow)
4. ETL
5. Building OLAP Cube
6. Building web-base OLAP application with ASP.NET

Figure 4.1 6 Steps of experiment
1. Analyzing the requirement and sources data

This step is very important basis for developing the further steps. “Well begun is only half done”, so it is essential to ask the right questions in this phase. Knowing the data and requirements will lead to find the measures and identify the dimensions that describe the measures. Furthermore, finding the right measures and dimensions can help to identify their relationships and all other model in design phase.

Gathering requirements involves identifying several types of requirement such as:

- **Business requirements**: owner requirement such as business process, business objective, end-user requirement, and expected output report. Every requirement would be clearly defined before any development gets started because changes in the design will later on costly. Lack of understanding the requirement may result in a high risk of failure rate. Therefore, all definitions must be determined with great care.

- **Statistic Report or multidimensional report**: this requirement is very important because it will help to design database structure and select data from source (OLTP system) to the proposed system. Moreover, it is the ultimate goal of the project. Therefore, it should be clear and fulfill user’s need. As a developer team, specific report must be defined in the scope of project. If some report may be added or changed, that implies addition workload. In this phase, the specific report must be extracted and transformed to multidimensional format as show in figure 4.2 and 4.3.
Figure 4.2 Example of Multidimensional report requirement for Statistics of student who are in Research process
Figure 4.3  Example of Multidimensional report requirement of Statistics of advisor and ratio of student they are holding in the past (cover 5 years) and present
Analyzing the Sources of data

Examine the existing database Entity Relationship Diagram (ERD), structure, attribute, and platform to determine which tables held the information required to answer the question. Sorting out what is beneficial and what should be stored in the data warehouse. Knowing exactly throughout the data sources structure can be helpful to extract and select the desired field to use in DW database. For example of this project, source data are stored in an OLTP relational database system (MS access and Oracle). The entire database structures (relationship model) are captured and studied thoroughly (see Appendix B and C).

2. Logical and physical dimensional database design

Logical database design

After analyzing the requirement phase, a new database model is created from the acquired data. It is a logical design. The logical design is a conceptual and abstract. The technique of entity-relationship is used to model. The process of logical design involves arranging data into a series of logical relationships called entities and attributes. An entity represents a chunk of information. In relational databases, an entity often called a table and attribute called a column or field. An attribute is a component of an entity that helps to define the uniqueness of the entity. To be sure that the data is consistent, a unique identifier is added to tables. Therefore, it can differentiate between the same items when it appears in different places. In a physical design, this is usually a primary key. The logical design should result in a set of entities and attributes corresponding to the fact and dimension tables. (Recall Fact and Dimension tables definition in chapter II).

Surrogate key is added to the dimension table as a primary key and fetched to the fact table as a foreign key. Fact table in this experiment is a Fact-less (no numeric measure and using count aggregate function in OLAP cube). The multidimensional reports from previous step are used to design a set of fact and dimension tables. The numeric in a cross-cell of report is a measure, which come from
the fact table. The labels around that numeric come from dimension table. In the first iteration design, 1 set of entity relationship is designed for 1 report model as shown in Figure 4.4. The next iteration is merging all set of entity relationships together as shown in Figure 4.5 to create a single model of the target data warehouse schema.

Figure 4.4 A set of entity relationship before combining into a single ERD of DW database
From the above figures, each set of entity relationship is different on the fact table, but the dimension tables are always a duplicate (same table). Therefore, a single diagram can be created. The number of table should be decreased and the new overall entity relationship diagram is shown in Figure 4.5.

There are 4 fact tables and 4 dimension tables. A fact tables contains an additive key or a compound primary key. This key is constructed from the system-generated of each dimension table (Surrogate key). Fact tables hold a detailed data, which fetch from OLTP transaction table, whereas dimension tables hold a master data. A master data is a unique data, such as a students, professors, and Date/time. A dimension tables provide a detail to the fact tables. The approach to design the dimension table is similar to the object oriented programming (OOP) design. A dimension can be thought of as an interface to a fact table (cube), with the structure of the dimension used to provide attribute to a cube.
Moving from Logical to Physical Design

This step is to translate the expected logical schema into actual database structures. Physical designs in DW system are not different from OLTP system but the DW database designs are mainly driven by query performance and database maintenance. The entities are linked together by using relationships. Attributes are used to describe the entities. The unique identifier (UID) distinguishes between one instance of an entity and another. At this time, the following objects have to be mapped:

- Entities to tables
- Attributes to columns
- Relationships to foreign key constraints
- Primary unique identifiers to primary key constraints
- Unique identifiers to unique key constraints

Physical design is a creation of detailed design of the actual database, which contains table’s name and attributes (field). Each attribute must define properties such as data type, data length and constraint as shown in Figure 4.6.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default Value</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std_key</td>
<td>Surrogate key</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>PK</td>
<td>Identity</td>
</tr>
<tr>
<td>Std_id</td>
<td>Student id</td>
<td>Varchar</td>
<td>15</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_name</td>
<td>Student name &amp; surname</td>
<td>Varchar</td>
<td>100</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_field</td>
<td>Student field or Major</td>
<td>Varchar</td>
<td>30</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_admitDate</td>
<td>Admit Date</td>
<td>Date/Time</td>
<td>No</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_signOut</td>
<td>Flag for signing out status</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default Value</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std_key</td>
<td>Key from student dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Time_key</td>
<td>Key from time dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_key</td>
<td>Key from professor dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Thesis_key</td>
<td>Key from thesis dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_status</td>
<td>Store data where come from OLTP (in-process student)</td>
<td>Varchar</td>
<td>10</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.6 Physical design of actual database
In the next step, the database object is created with DBMS for the physical design. For this experiment, the SQL server 2000 is used to implement and manage the project. First, the database object name is created. There are 2 database object names in SQL sever 2000. The first one is the DW database, and the second one is the staging database. The staging database is a cache or a temporary area of ETL process. Some data from source system must be staged here before transferring to DW database. The database structure of the staging area is similar to OLTP system. There is no need to create relationship between each table. The easiest way to create the staging database is copying tables from OLTP system to this area without changing any attributes. The DW database contains dimension and fact tables that linked together in star schema as some example of tables in Figure 4.7 and 4.8.

In this project there are 2 types of table; 1) dimension table which represent by adding ‘dim’ in the end of table name, and 2) fact table which represent by adding ‘fact’ in the end of table name.

Figure 4.7  The User Interface (UI) of table creation in SQL Server 2000
Star schema for the report of advisee who are in Research process

Star schema for the report of graduate student in each academic year

Figure 4.8  Example of Star schema
The *time dimension* is a very important issue and is a salient characteristic in DW system because time defines the timeline from past to present. The time variance implies that every unit of data in the DW is accurate as a moment in time. Any record has a date of transaction. Therefore, special care are needed to create and manage time dimension.

Functions to generate date records are run at once and separate the date field into separated columns as month, year, and number of month for query performance and readability. There are around 3 thousand records stored in this table at once and next updated by adding new record year by year. Executing the following code for generation of the date.

```vba
Private Sub generateDateTime()
    Dim dbs As Database
    Dim rst As Recordset
    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("DW_thesis")

    Dim dteStart As Date
    Dim dteEnd As Date
    dteStart = #1/1/2000#
    dteEnd = #12/31/2005#
    Do While dteStart <= dteEnd
        rst.AddNew
        rst("TheDate") = dteStart
        rst.Update
        dteStart = DateAdd("d", 1, dteStart)
    Loop
    rst.Close
    dbs.Close
    Set rst = Nothing
    Set dbs = Nothing
End Sub
```

*Figure 4.9* Example code for generation the date of the time dimension table in DW database
3. Design Data transfer structure (Data flow)

From previous step, the new database structure can be obtained data for the DW project. Now is the time to move data from OLTP source to DW database. But before fetching the data, the data flow diagram is needed and should be designed before the ETL operation starts. This structure helps to identify which data to transfer, where to store, and in which platform to run. Including which step of task will process in sequential or parallel. This step is called preparing plan for ETL. The numbers in Figure 4.10 represent a sequential step of process.

**Figure 4.10** The architecture of preparing plan for ETL process.
4. **ETL (Extraction, Transfer, and Loading) program**

This step is the activities that relate to ETL which is the most time-consuming, complex, and intensive.

**Data Extraction**: Clearly identify all the elements of data source, structure, and computing platforms.

**Data transformation**: A function to map and prepare data before loading to DW.

**Data loading**: Execute or move data into DW database at once and set frequency of next update.

4.1 **Data Cleansing**

Data quality is very important issue to improve data quality. The general process of data cleansing is removal or correction data before loading into DW database. There are several methods to do such as detecting duplicate record (Domain relevance or sorting), using violating integrity constraints, K-way sorting method, and Union-find algorithm [5]. These processes are performed in staging area. Union and Intersection operations may be required to correct data as an example of SQL statement shown in Figure 4.11.

WHERE (((Professor.Status) Like "in*") AND ((ThesisTest.Sort)>1));


Figure 4.11 Example of SQL statement for data cleansing

4.2. Design Fact and dimension tables Transferred (Code preparing)

From previous step, the chosen table is transferred to physical database. The dimension table must be first transferred before the fact table. Transferring dimension tables by using Data Transformation Services (DTS) tool is very simple because the DTS tool provides IDE to ease the transformation. But transferring data to fact table is not as simply because it uses a multiple joined table in multiple databases and uses a surrogate key in dimension table. Therefore, the complex SQL command is used to run the specific queries. To make this task easier, a logical set of entity relationship should be drawn before writing a pseudo-code of SQL statement as show in Figure 4.12 and 4.13. This statement will be placed in fact transfer task of DTS package creation (in section 4.3).
Figure 4.12 An ERD for code preparing of the fact transferring

**Pseudo SQL statement for source data**

```
Select std_key from student table in DW database
    time_key from time table in DW database
    Prof_key from professors table in DW database
    ProfStatus from Onproposal in OLTP database

From thesis table in OLTP database
    inner join DW.tbl_students_dim ON OLTP.Onproposal.Thesis_id = DW.tbl_students_dim.std_id
    inner join DW.tbl_time_dim ON OLTP.PropoTestDate = DW.tbl_time_dim.GDate
    inner join DW.tbl_professors_dim ON OLTP.Onproposal.prof_id = DW.tbl_professors_dim.prof_id

Where (OLTP.Thesis.propoTestDate is not null and
OLTP.Thesis.ThesisDate is null)
```

Figure 4.13 Example of pseudo-code for SQL statement of the fact data transferring
4.3 Create DTS package for Loading

After a data source and preparing transfer plan complete, it is the time to create DTS package to operate and load data from source to destination. Data Transformation Services (DTS) is a tool for creation a simple or complex data movement (import and export) and integration data from disparate sources to a single database. The data movement can consist of a single DTS package or a number of interrelated DTS packages. Each DTS package contains connection objects, tasks, and workflow constraints. DTS tool can connect to several databases such as MS SQL server, MS access, Oracle, Excel, and OLAP database or other text file. Connection objects define data sources and data destination structure. Tasks define the actions that will be performed within the DTS package. Precedence constraints control the task order executed. When the package is executed, tasks without precedence constraints are executed in parallel. For this experiment, there are 2 DTS packages. The first one is for loading data to staging database and the second one is for loading data to DW database.

![Figure 4.14 DTS package for table creation and loading data to Staging area (run once)](image)

When the temporary table (in staging area) is ready for loading, the actual data is moved to DW database. The second DTS package is responsible for loading data from staging database to DW database. This package is used in both run once and run periodically, so cube processing and updating data are included. Before creation
DTS package, rearranging task may be needed in order to assign table priority, such as, master tables must be first transferred before detailed tables. Creation this package is divided into several packages and then merged them to a final single package as showing in Figure 4.15.

**Figure 4.15** DTS package for loading data to DW and OLAP database

From Figure 4.15, there are 4 main steps to process in sequential. The stripe lines represent a precedent job (on success job). If precedent jobs are not success or error, the following jobs cannot be executed. The solid lines represent transformation of data task of both fact and dimension tables. The complex SQL statements, which
prepare for the Fact table transfer in section 4.3 are placed in the transformation task. The transformation task can be SQL command and ActiveX script. In each step of the DTS package have several sub-processes which can run in parallel.

Step 1:  
- Cleansing data in OLTP data source  
- Transfer master tables to the dimension tables of DW database in SQL server where source tables are come from MS access and Oracle (export to text file format).

Step 2:  
transfer detailed tables to the staging area database in SQL server where source tables are come from MS access.

Step 3:  
transfer detailed tables from the OLTP staging area to Fact table in DW database where both tables are stored in SQL server.

Step 4:  
Update data to OLAP cube. Process in this step can be set to full process, refresh data or incremental update. In this experiment, full process is required for accuracy data purpose.

Batch copy (reload data) and incremental data update may be used to update DW database. Incremental update is more complex than batch copy. The DTS package might be re-created or change coding in SQL statement and ActiveX script for update data because only a different record or changed data will be transferred to DW database. This implies that the OLTP database should be changed, such as adding field for marking a point of selectable record. Therefore, an additional process is needed in preparing plan for ETL process. The first process is the first transfer and the second is update transfer as shown in Figure 4.16 and 4.17.
Figure 4.16 Architecture of the first data transfer to DW database and use during experiment.

Figure 4.17 Architecture of update transferring in periodic run
5. Building OLAP Cube

A cube is a precise definition of an OLAP cube. OLAP allows user to access summary data faster and easier in multidimensional view. User can drill down to see more detail of data (lower level) and roll up to see more summary data (upper level). A cube provides a multidimensional view of data that extend beyond standard 2-dimension and allow flexible data viewing. A cube must always have at least one measure, one dimension, and one fact table. A summary data in the cube is called aggregation, which pre-calculated. Each cell is an intersection of the dimensions of cube holds one value.

Creation OLAP Cube will follow 2 steps:

5.1 Create view
5.2 Create cube in SQL Analysis Services (OLAP engine)

5.1 Create View

Before creating a cube, a data source for operation in the cube must be prepared. Fact table and its dimension must exist. Data source of OLAP cube can be a view (virtual table) or actual table. The knowledge of MDX (Multi-dimensional expression) in OLAP database is not require, if the view in a native database capable to create a source data. From this experiment, the view is used to create a data source (Fact and dimension table) for a cube. A single view may come from several tables that link together as a star schema (1 view/several cubes, 1 cube/1 report).
5.2 Create cube in SQL Analysis Services (OLAP engine)

After the environment and basic needs of OLAP cube are ready. Creating cube becomes simple. A user-friendly interface tools such as Analysis Services (included component of SQL server 2000) is used to create cube. It provides a graphical user interface called Analysis Manager to work easily with cube as illustrated in Figure 4.19.

```
SELECT     dbo.tbl_students_dim.std_id,
           dbo.tbl_students_dim.std_name, dbo.tbl_students_dim.std_field,
           dbo.tbl_students_dim.std_signOut,
           dbo.tbl_thesis_dim.thesis_title_en,
           dbo.tbl_thesis_dim.thesis_type, dbo.tbl_professors_dim.prof_name,
           dbo.tbl_professors_dim.prof_degree,
           dbo.tbl_advisee_inprocess_fact.prof_status,
           dbo.tbl_time_dim.GDate, dbo.tbl_time_dim.GMonth,
           dbo.tbl_time_dim.GYear,
           dbo.tbl_advisee_inprocess_fact.std_key
FROM         dbo.tbl_advisee_inprocess_fact INNER JOIN
              dbo.tbl_professors_dim ON
              dbo.tbl_advisee_inprocess_fact.prof_key =
              dbo.tbl_professors_dim.prof_key INNER JOIN
              dbo.tbl_students_dim ON
              dbo.tbl_advisee_inprocess_fact.std_key =
              dbo.tbl_students_dim.std_key INNER JOIN
              dbo.tbl_thesis_dim ON
              dbo.tbl_advisee_inprocess_fact.thesis_key =
              dbo.tbl_thesis_dim.thesis_key INNER JOIN
              dbo.tbl_time_dim ON
              dbo.tbl_advisee_inprocess_fact.time_key =
              dbo.tbl_time_dim.time_key
WHERE     (dbo.tbl_students_dim.std_signOut IS NULL)
```

**Figure 4.18** Example SQL statement for a single view of data source for a cube creation of advisee in process report
From Figure 4.19, in the left pane side, it appears a list of database name and its cube. In the right pane, at the Meta Data tab, it describes a property and detail of the database that is highlight. The OLAP database in this experiment is named OLAP_thesis_rans.
Figure 4.20 shows a cube creation in a design mode. The source data of this cube come from a single view (virtual) in SQL server. This view table combines a measure and dimension attributes into single table. When creation a dimension or measure, drag and drop activity can be easily used. Multidimensional reports in Figure 4.2 and 4.3 are the best guide to drag and drop fields into dimension and measure folder. In the design mode, the MDX function may be used for specific purpose of report. In this project, the fiscal academic year to present a statistic report of graduate student in each academic year is used. Therefore, a special function to control time variant must be written.
After a cube creation, the dialogue as shown in Figure 4.21 will prompt to select the type of data storage. There are 3 options for the storage type with its description. In this project, MOLAP type is used to store data because it is faster and suitable for a pilot project (data mart) with a few data. MOLAP require more space than ROLAP and HOLAP because it copies the original fact and dimension tables into the OLAP database. The MOLAP cube takes more space, but it no longer requires the star schema.

![Figure 4.21 Design a storage type of a cube](image)

The final step of cube creation is to add browser data in multidimensional view. Drag and drop dimension label in the top pane side into dimension area in bottom pane side. Activate drill down and roll up function by double click at plus or minus sign as shown in Figure 4.22.
Figure 4.22 User Interface of browsing data in OLAP engine
6. Building web-base OLAP application

Even though data in OLAP tool can be browsed, but it is not flexible and uncomfortable for user, especially a decision-maker or any non-technical user. Therefore, the benefit of World Wide Web Internet technology is employed to provide OLAP data cube to user. It can support corporation wide business analysis, provide an informational resource, and also a wise delivery mechanism. This means that advanced analysis and reporting tools must be capable of interfacing with the data warehouse and OLAP database via the web. In general, such advanced analysis tools must be capable of running inside a web browser to take advantage of thin-client deployment.

One of the key features in MS SQL server 2000 analysis services is the capability to access data from the web. The PivotTable Service (PTS) OLE DB provider uses a special Active Server Page (ASP) web page to send and receive data using the HTTP or HTTPs protocol. The ASP page uses a single COM object, called the data pump, which acts as a conduit for Analysis Services data. The library that contains this COM object is referred to as the data pump library. Analysis Services required for Internet Information Services (IIS) to be installed on the same server, because the data pump library used shared memory to communicate directly with the Analysis server.

ASP.NET and the Microsoft Office Web Component (OWC) can enable Web-based OLAP reporting. The OWC controls include PivotTable and Chart components that can be embedded in a Web page and scripted by programmers. It helps to build sophisticated and user-friendly OLAP report easily. The OWC components also expose a rich programmatic interface, which provides a high degree of functionality, flexibility, and can build a custom solution.

In this project a web-base OLAP reporting is created by using ASP.NET, OWC, SQL server 2000, and Analysis Service (OLAP engine) because a reasonable benefit as describe above. There are several ways to create web-base OLAP solution. Several third party tools provide a free demo program via the Internet. All of them are also provide a user-friendly interface, help, and wizard, which can be easily used. For an advance programmer, custom a code may be written to connect to OLAP database.
and present the report via the Web. For this experiment, web-base OLAP is built under client/server architecture (2-tier) as shown in Figure 4.23.

![Figure 4.23: Client/Server architecture for web-base OLAP Implementation](image)

Before building the web-base OLAP application, IIS (Internet Information Services) must be first installed on the same machine of OLAP database also the .NET framework. Then create web site directory. In root directory, the single file called “msolap.asp” must be stored there. This file acts as the bridge between request coming over HTTP protocol on port 80 and Analysis Services data source. It must be located at the root web site directory where aspx files are stored. The ASP page use a single component object called PTS. The PTS attempts to access a specific file msolap.asp. The system append the string “/msolap.asp” to the URL and use HTTP protocol to send a POST request to the msolap.asp page, including the connection string & other data and meta-data as the binary part of the POST request. Here is an example of OLAP database connection string "OLAPConnectionString" value=

"Provider=MSOLAP.2;Integrated Security=SSPI;Persist Security Info=True;Data Source=http://10.106.2.251;Initial Catalog=OLAP_thesis_rans ;Client Cache Size=25;Auto Synch Period=10000".
An important thing when using OWC component as an aspx file connects to OLAP database via HTTP protocol is don’t forget to set user access role in each OLAP cube. If this property is not set, it is impossible to connect to OLAP cube. Here is the structure and components of web-base OLAP.

![Diagram](image)

**Figure 4.24** The structure and components of web-base OLAP (for each asp page)

In web-base OLAP security, if someone attempt to open a second HTTP session for the same process, the session cookies for the first session is dropped and the second connection is opened. From those figure and configuration desire, if everything has been configured properly, the web-base OLAP report should perform as shown in Figure 4.25.
Figure 4.24 The final web-base OLAP report that provide on WWW and using browser (Internet Explorer) to view as a dynamic web page.

Figure 4.25 The final web-base OLAP report that provide on WWW and using browser (Internet Explorer) to view as a dynamic web page.
It is the end product that allows user to drill down, roll up in pivot report and multidimensional view. With this OLAP report tool, a table can be easier rotated as just drag and drop a dimension to report (drop filter area). Bar graph as show in figure is the one of OWC component.

In this step, testing is needed to perform for error detection of system programming and data correction. All errors must be removed and incorrect data must be adjusted. When users see the final product, the changed requirement or added more requirements from user may be acquired, which is useful for future development. The next step of the project is the implementation. In the implementation phase, a plan for operation such as network diagram or software configuration must be laid out. For this project, 2 programs are installed in each client machine (PTS and OWC) which are connect to LAN or WAN of MUC-net. The network diagram for the installation is shown in Figure 4.26.

Figure 4.26 Wide area Network diagram for DW and web-base OLAP implementation
From this network diagram, OLAP server located at Salaya campus must install IIS web server, .NET framework, OLAP database, and SQL server 2000. Others client machine may be located at either Salaya or Phayathai campus and must install OWC and PTS program.

All experiment about Data warehouse and web-base OLAP system solution are presented. Even though only a single line of business is built, which is called data mart, but every step and issue in DW project are used in this pilot project. It is a highly complex job and has a lot of tasks to perform. This solution acts as a conduit for an “Information life cycle” which never ends.
Building a Data Warehouse (DW) is proved to be a challenging project. It is not just a software or application. It is beyond software engineering and organizing. The activities include managing, planning, modeling, analyzing, and specifying must be well studied and controlled. The DW provides more than theoretical foundations for building software, and also focuses on implement software in a controlled and scientific way. Furthermore, it needs continually looking for new techniques and tools to improve process and product.

Building a DW relies on a developer who are enrich in the skills of system analyze, computer programming, and database management with clearly understanding business processes. Therefore, knowledge of data modeling, relational algebra, query processing, transaction processing, and advanced database architecture are all prerequisites for DW techniques. In software engineering, the DW is quite a young discipline and does not yet offer well-established strategies and techniques for the development process. The success or failure of DW is closely related to several factors such as the existing database management system, planning, methodology, technique, and tools. Data and goal-driven method establish this project. The starting point of this project is from data mart in Thesis Operations data at school of nursing Ramathibodi hospital. According to the thesis objective and problem statements, defined in Chapter I, the DW methodology is used to solve the problems. It may lead to achieve the ultimate goal as anticipated. Working with this project is very exhaustive and time consumed, but the outcome is worthy and benefited.
Benefits of this project

As a developer: Though the DW and OLAP solution is a complex system. There are no silver bullets that will make project successful except hard work and disciplined approach. After finishing this project, it worthily returns a sophisticated technique and skills. Many people mistake to believe that DW is just a kind of data model and tools. Actually, it composes of many programming activities and processes especially the ETL process. Additional skills are the capability to extract and analyze data from structure or unstructured into multidimensional view. These skills are very important to create cube and adapt simple data to multilevel of aggregation. Moreover, they help to explore implicit need from decision-maker to take benefit form the historical and current data.

As a user: DW enhance job productivity by enabling users to perform their decision making faster, easier, and more comprehensively as well as reducing time to access data. It also helps to re-organize the workflow and create knowledge to support better decision-making.

As an organization: The DW with OLAP system acts as a BI (Business Intelligent) solution which can be used to improve business process and strength competitive in the world of IT. When building the DW, some traditional working process must be changed for the data quality, productive and performance of new system. Those imply that, they need standards of working process to apply to their business process.
**Problem of this research**

**Data cleansing**

At first glance, when data is moved from the legacy environment to DW, it appears that nothing more is going on than simple extraction data from one place to another. Actually, there are more complex activities such as:

- Extract data from the OLTP to the DW environment require the selection data only a quality record for extraction processing, some record may relate to several records in a variety of other files and tables.
- Operational input keys usually need to be restructured and converted before they are written to DW database. Because missing or incorrect data will cause problems in analysis and system integrity.

**Transferring data**

After data cleansing process, data transfer will be taken place. Transferring dimension table is much simple but fact table transferring is not simple. Because it uses a multiple joined table. The multiple joined tables are needed to link each row in the fact table. According to using Surrogate key in star schema database, it enforces the use of a complex SQL command to run a specific query. Cross inner join relationship is used to map tables between tables in OLTP and DW database. To solve this problem and to make it easier, a diagram is drawn to show all tables, elements, and its link before coding. It helps us to write a complex join of SQL statement easier. We draw 1 diagram per 1 fact table transferring.

**Data gathering**

Another major obstacle of the DW project is that there is not enough historical data stored in the application to meet the need of executive report. In beginning of project, the report requirements are gathered from user. The executive often cannot describe what they want. The DW project cannot start based on complete user requirements because some requirements cannot or should not be fully defined before the warehouse is first used operation.
Change requirement

Scope creep or user need more requirements is one major problem of development project. We assume that requirements are known at the start of design. But in the world of decision-maker analyst, new requirements usually are the last thing to be discovered. Most decision-makers always ask questions for a new requirement when they see an initial report screen. Therefore, in gathering requirement phase, great care must be taken at the decision-maker who may be a business person who is exploring for the real business needs.

In gathering requirement phase, it is common that the user don’t know what data is most significant to the organization, what question is very difficult to answer, what kind of information is most important, and why. So, we make interview with 2 majors groups of user:

- The first concentrate group is those who have the question (decision-maker)
- The second group is those who know the data (data owner)

The point of questions is to understand what data is most significant to each subject area.

Create web-base OLAP

According to objective, the project reports must be delivered on web. A robust report writer which is capable to produce and publish our data to user in a dynamic web page is required. The best solution to provide OLAP’s data on web must be determined. There are many OLAP front-end products such as Microsoft, Hyperion solutions, Cognos, Business Objects MicroStrategy, Oracle Express, SAP and etc. And there are plenty solutions that can access to OLAP server and present data via Internet in multidimensional view such as Crystal report, Reporting Service, and any third party components. The OWC (Office Web Component) which add on with Microsoft Office is most suitable for our OLAP front-end report. They provide a standardized OLAP API and OLE DB, which aimed at providing a COM component-based standard for accessing disparate data source. Using Microsoft OLAP product with OWC and .NET to gain web enable data warehouse is not easy. There are many activities and configuration to do such as setting up web server, configuration role
management of OLAP cube, and setting connection string when using HTTP protocol of aspx page to access OLAP server. Further more in implementation phase, all any activities must be clear defined such as client/server machine configuration and network diagram.

Tools

At the heart of the DW is architecture, so the architecture would be first and tools are second. But selecting the right tool can helps to meet goal faster, safer, save cost, and minimize learning curve. If we chose the right tool, we can concentrate to architecture, business process, and result. Remember that the ultimate goal of DW project is to present useful data to user. Users of this system are always an executive. They don't care what is needed to be done, nor how hard to get the result. They just need the accurate data that are prompted and easy to use whenever they need. So the data must be accurate. If some errors appear in data, the integrity of the system is compromise. Tools were selected in this project are Microsoft package which compose of:

- SQL server 2000 (for DW database and staging database)
- DTS (for ETL process)
- Analysis Services 2000 (for OLAP server)
- OWC and ASP with .NET framework (for information delivery on WEB)

The benefits of selecting this package are compatibility, decreasing error debugging rate, usability, and maintenance. From those tools ability, it can help us to meet goal on time, on budget and on further improvement.
Recommendation or further study

Security policy

The security policy is beyond the scope of this project. However, security in the web enabled DW is the first priority to implement. It is very challenge topic to apply and implement. There are many methods and technologies available to ensure data protection. The first technology is encryption, which uses sophisticated algorithms to scramble data when it is transmitted. There are also authentication services that request, validate and process user information to enable authorized users to log on. It is SSL (Secure Socket Layer) technology that sits on the top of the TCP/IP communication stack. The security policy can restrict to end user who can grant at physical table level or on LDAP-base network resource called a directory access protocol.

Real-time DW

This topic is about data movement or ETL process. This concept is how to move the data straight from source to single DW database. The movement takes place as soon as the original data are written and supports continuos of data stream, synchronous, and multi-point deliver of data [5].

Intelligent Agent

The intelligent agent can be used to automate the maintenance of DW by controlling data loads or monitoring the applications performance [5]. The agent can track the arrival of new data sources and monitor quantity of data flow from each department. This agent acts as a supervisor to monitor and control workflow of data. It should also has its own database and reporting services.
System Evaluation

Evaluation is a feedback from the users. It can help to understand and correct any errors of our project during development and after completion. It can also encourage us to improve efficiency of our process and quality of product. The objective of the evaluation is to assess the user satisfaction. We create the user satisfaction questionnaire, which compose of 8 items in 2 categories (System and Report). The users rate each item with satisfaction scale ranging from 1 to 5 where “1” means not satisfy and “5” mean very satisfy. Our target groups are the faculty executive and supervisors. Ten people are chosen to answer the questionnaire. The most significant of satisfaction is the ease of use and usefulness. They have 100% of the vote. The accuracy and integrity of report and Completeness of functionality items have a fewer scored (91.2% and 90%) because the users don’t know that the data (Thesis operations data) is of poor quality. The overall satisfaction of this project is 95.1% as illustrate in table below.

Table 5.1 The items of evaluation user satisfaction mean score of each item.

<table>
<thead>
<tr>
<th>Items</th>
<th>Satisfaction scale</th>
<th>mean</th>
<th>level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of use</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>[5]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usefulness</td>
<td>5</td>
<td>1</td>
<td>4.90</td>
</tr>
<tr>
<td>[4.5] [0.4]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance and response time</td>
<td>5</td>
<td>1</td>
<td>4.78</td>
</tr>
<tr>
<td>[3.5] [0.8]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for semi-structure and unstructured data</td>
<td>5</td>
<td>1</td>
<td>4.60</td>
</tr>
<tr>
<td>[3.5] [1.6]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall satisfaction of reports</td>
<td>5</td>
<td>1</td>
<td>4.56</td>
</tr>
<tr>
<td>[3.5] [0.8]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User-Interface and design</td>
<td>5</td>
<td>1</td>
<td>4.50</td>
</tr>
<tr>
<td>[2.5] [2.5]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In conclusion, the most significant concepts of the DW and OLAP are the architecture and plan. Data quality is also a major factor indicator that leads to the success or failure project. Actually, building the DW to run only one report is a costly waste. Starting from a small project to a wide enterprise project, the cost of building a DW is worthwhile. Cost justification of build DW cannot use a simple technique such a ROI to justify, because the DW is built incrementally. The first iteration can be done quickly with minimum effort and cost saving. Once the first portion is built, the analyst can start to explore the possibilities and justify the cost for the next move. The first iteration should be small enough to be built and large enough to be maintain. There should be direct loop feedback between the DW developer and decision-maker. No project is perfect in initial. There are always ways for improvement.
CHAPTER VI
CONCLUSION

Most modern organizations have an existing database and OLTP system, but those database are normally stored in heterogeneous sources, inconvenience to access and unavailable to use for immediate decision-making. They are limited access to IT staff and daily operations. OLTP systems produce a transaction data every day and grow rapidly with time. However, they do not fully support ad-hoc query, analyze and summarize of strategic information, which are useful to executive and the decision-makers. Information is a valuable asset of organization and is a power of competitiveness. They will be worthless if they are stored in unstructured model, unavailable to access, hard to use and difficult to retrieve. Furthermore the cost of maintenance will grow rapidly and wastefully. Moreover, if the data analysis is inaccurate, it will lead to poor decision making. At this point, we assume that all organizations need to make decision efficiently based on accurate data rather than rough estimate of incomplete information. The use of historical data for statistical analysis will improve decision making of the modern executive.

**Data Warehouse** (DW) is the one of emerging technology that can solve these problems. The DW concepts are created for the purpose of integrating and organize large amounts of data for the ease of retrieval and analysis. A DW is typically a blending of technologies, including relational and multidimensional database, client/server architecture, extraction/transform/loading, graphical user interface and the www information delivery. The main characteristics of DW are subject-oriented, integrated, non-volatile, time variant, and support decision-maker. The data model in DW is always different from OLTP system. In OLTP system, data model is always a relational with normalized form. But in DW is much simpler model by de-normalizes into star and snowflake schema.
Data that stored in DW come from several sources (existing OLTP or legacy system). Historical and current data from various sources must be cleansed and verified before it is deposited into the data warehouse. The ETL process is the most significant step of data quality before execution a data analysis, data summary, and presentation.

DW facilitates the analysis of stored data and can effectively deliver an On-Line analytical processing (OLAP) for decision support queries. It can also serve as an enabling technique for data mining. In this project, OLAP front-end is used to access to DW database, and is also used to analyze, summarize, and present information to the user via an Internet browser. User can use a browser to browse data in multidimensional view, drill-down, roll-up, slice and dice format.

Data source from School of Nursing, Ramathibodi Hospital, Mahidol University, which stored on RDBMS system (Microsoft and Oracle) and external flat file that covering 5-10 year period is extracted to form of DW database. Building this project is bounded to the project scope in chapter I and have an ultimate goal to produce online reports. We start building DW from a single subject, so we may call this project a data mart.

In experimental phase, we followed a guiding methodology, which are defined in chapter III. The major steps of building the DW are as follows:
Statistic report or multidimensional report is very important because it is the best guideline to design database structure and select field (data) from OLTP system to the new system. It also used in OLAP design and online reports. Moreover, it is the ultimate goal of the project. Therefore, it should be clear and fulfill the user’s need.

The results of this project are a new DW database model, ETL process, and OLAP front-end report with ASP.NET. The benefit of World Wide Web technology is used to deliver OLAP data cube to user. It can support Corporation wide business analysis, provide an informational resource, and also is a wise delivery mechanism. Report is the end product of the experiment that allows user to drill-down, roll-up in pivot report and multidimensional view. With this OLAP report tool, a table can be easier rotated as just drag and drop a dimension to report.
The project evaluation, satisfaction questionnaire is created to assess the project. The target groups are the faculty executive and supervisors. The highest satisfaction is found in two features: ease of use, and usefulness results (100%). And the items that have lowest score are accuracy and integrity of report and completeness of functionality (91.2% and 90%) because of data inaccuracy (Thesis operations data). The overall satisfaction of this project is 95.1%.

Success factor of building the DW project relies on creating DW, which determines what information is needed most, what decision-makers want to know and what the most important business questions are. The following guidelines are needed to be met:

- Planning and using appropriate methodology
- Starting from a small projects (a single line of business subject)
- Clear defined scope and objective
- Understanding the business processes and take more time to listen to the decision-maker or data owner to obtain their needs.
- Data completeness and data qualities are most significant success factor.

In gathering requirements phase, concentrate on data selection. Selecting the data that useless for user should be avoided. Try to decrease the number of the sources of data and select only integrity data sources.
- Selecting the right tool (HW/SW) which we are capable to manage and its capabilities can solve the problems
- Survey HW/SW environment such as Server/Client machine, network, and internet supported.
- Social technical skills such as approaching to data owner and decision-maker or executive
Building the DW is very challenging project. Many computer skills and techniques are used in this project. There are no silver bullets that will make project successful, except hard working and a disciplined approach of developer team. The ultimate objective goal of DW and OLAP system is ‘FASMI’ (Fast Analyst of Shared Multidimensional Information). It provides the mean to change raw data into information for making-decision, provides the right data, right people and right time.

Before starting a project, the main concept need not be perfect, but preparing must be close to perfect. Learning from doing is forever true. In the world of software solution, innovation and differentiated thinking are most important and are always come from doing.
REFERENCE

11. สมพร จิวรสกุล. SQL Server 2000 ฉบับสมบูรณ์. Inforpress DEV Book; 2545.
15. Somjintana Srettha. *The design and development of the three-dimension data warehouse framework (3-d dwfw).* Faculty of Graduate Studies, Mahidol University; 2002


19. [http://msdn.microsoft.com/ msdnmag/issues/03/10/OLAP/default.aspx](http://msdn.microsoft.com/msdnmag/issues/03/10/OLAP/default.aspx)


APPENDIX A

Functional requirement and use case diagram

Use Case Diagram

OLTP system
Research Operation system

OLTP system
...

Data warehouse with OLAP system
fetch data from OLTP system
run DTS package

Admin

user2

User

View report online via internet
Print or Export to file
drill-down, roll-up, slice and dice a report
ตารางแสดงความต้องการของผู้ใช้

<table>
<thead>
<tr>
<th>Subject area</th>
<th>Research Operation (ข้อมูลการทำงาน ของนักศึกษาพยาบาลรามาธิบดี)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System owner:</td>
<td></td>
</tr>
<tr>
<td>Date:</td>
<td>(C = Critical, E = Essential, NE = Not Essential)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>คุณสมบัติสำคัญของระบบ</th>
<th>C</th>
<th>E</th>
<th>NE</th>
<th>note</th>
</tr>
</thead>
</table>

**Functional requirement**

- สามารถดูรายงานผ่านทางเครือข่าย internet ได้ ด้วยตนเอง เฉพาะโดยไม่ต้องร่วมกับเจ้าหน้าที่ ที่ปฏิบัติงานประจำ  
- รายงานแต่ละอัน สามารถแสดงรายละเอียดในภาพรวม และรายละเอียดปลั้งปลิวได้  
- รายงานแต่ละอัน สามารถพิมพ์ออกทางเครื่องพิมพ์ได้

โดยมีรายงานที่ต้องการ และคุณสมบัติพิเศษ ดังนี้

1. รายงานสถิติ ของนักศึกษาที่กำลังทำงานวิจัย

2. รายงานสถิติ ของนักศึกษาที่ยังไม่ทำงานวิจัย

3. รายงานสถิติ ของนักศึกษาที่จบในแต่ละปีการศึกษา

โดยที่สามารถดูได้ว่า

1.1 จำนวนอาจารย์ และอาจารย์แต่ละคน ดูแลนักศึกษาในฐานะที่เป็น Major advisor และ Co advisor อยู่ในแต่ละคน มีใครบ้าง สาขาอะไร รหัสอะไร (เรียงตามลำดับบัตรประจำตัว) และกำลังทำเรื่องอะไร

1.2 จากจำนวนนักศึกษาที่กำลังทำวิจัยอยู่ ทำแผน ก.ก.ค. ทำแผน ข. ค.ค. มีใครบ้าง สาขาอะไร รหัสอะไร (เรียงตามลำดับบัตรประจำตัว) และกำลังทำเรื่องอะไร

1.3 นักศึกษาแต่ละรหัส ทำวิจัยอยู่กี่คน ใครบ้าง สาขาอะไร มีใครเป็น Major advisor และใครเป็น Co advisor

2. รายงานสถิติ ของนักศึกษาที่ยังไม่ทำวิจัย

3. รายงานสถิติ ของนักศึกษาที่จบในแต่ละปีการศึกษา

แสดงจำนวนนักศึกษาที่ยังไม่ทำวิจัย (ยังไม่สอบโครงร่าง) พร้อมรายละเอียดอื่นๆ เช่น รหัสบัตรประจำตัว ชื่อนักศึกษา สาขา จำนวนนักศึกษาที่เรียนในแต่ละปีการศึกษา (academic year) โดยสามารถแสดงรายละเอียดได้ว่า ในแต่ละปีการศึกษา จำนวนนักศึกษาที่ยังไม่สอบโครงร่าง มีใครบ้าง จำนวนนักศึกษาที่ยังไม่สอบโครงร่าง มีใครเป็น Major advisor และใครเป็น Co advisor.
### Subject area: Research Operation (ข้อมูลการทำงาน ของนักศึกษาสาขาการจัดการระบบ)

**System owner:**

**Date:**

(C = Critical,  E = Essential, NE = Not Essential)

<table>
<thead>
<tr>
<th>C</th>
<th>E</th>
<th>NE</th>
<th>note</th>
</tr>
</thead>
</table>

4. รายงานสถิติ โครงการธุรกิจการเป็นกรรมการสอบ

แสดงจำนวนการเป็นกรรมการสอบวิทยานิพนธ์/ สารนิพนธ์ ของอาจารย์แต่ละคน ในฐานะที่เป็น Major advisor, Co advisor และ Committee และสามารถติดตามและจดได้จากในระบบการสอบให้กับนักศึกษาได้ รหัสอะไร (เรียงตามลำดับปีเข้า) เรื่องอะไร สอบวันไหน เดือน อะไร ปีอะไร

### Non-Functional Requirements

1. **System Requirements**

- Windows platform
- Internet explorer browser

2. **Usability Requirements**

ผู้ใช้ระบบ (end user) ไม่จำเป็นต้องมีทักษะทางคอมพิวเตอร์มากนัก ถ้าสามารถใช้งานได้

3. **Performance Requirements**

- ความเร็วในการเรียกข้อมูลต้องไม่เกินกว่า 30 วินาที
- ความปลอดภัย และความลับของข้อมูล
- Reliability, Availability
- Capacity

4. **Scalability Requirements**
APPENDIX B

Entity Relationship Diagram

The existing ERD of Thesis Operations which rely on Microsoft access
APPENDIX C

Entity Relationship Diagram

The existing ERD of Thesis Operations which rely on Oracle
# APPENDIX D

## DATA SOURCE

Data source from spreadsheet (MS Excel®)
## APPENDIX E

Physical Design (Data dictionary) of Data Warehouse database for Thesis Operations subject area.

### Table name: Tbl_students_dim
**Description:** Store master data of unique students

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default Value</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std_key</td>
<td>Surrogate key</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>PK</td>
<td>Identity</td>
</tr>
<tr>
<td>Std_id</td>
<td>Student id</td>
<td>Varchar</td>
<td>15</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_name</td>
<td>Student name &amp; surname</td>
<td>Varchar</td>
<td>100</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_field</td>
<td>Student field or Major</td>
<td>Varchar</td>
<td>30</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_admitDate</td>
<td>Admit Date</td>
<td>Date/Time</td>
<td></td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_signOut</td>
<td>Flag for signing out status</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table name: Tbl_professors_dim
**Description:** Store master data of unique professors

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default Value</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof_key</td>
<td>Surrogate key</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>PK</td>
<td>Identity</td>
</tr>
<tr>
<td>Prof_id</td>
<td>Professors id</td>
<td>Varchar</td>
<td>15</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prof_name</td>
<td>Professor name &amp; surname</td>
<td>Varchar</td>
<td>100</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prof_status</td>
<td>Internal / External teacher</td>
<td>Varchar</td>
<td>30</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prof_degree</td>
<td>Position</td>
<td>Varchar</td>
<td></td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std_signOut</td>
<td>Flag for signing out status</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table name: Tbl_thesis_dim
**Description:** Store master data of unique thesis title

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default Value</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thesis_key</td>
<td>Surrogate key</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>PK</td>
<td>Identity</td>
</tr>
<tr>
<td>Thesis_id</td>
<td>thesis id (student id)</td>
<td>Varchar</td>
<td>15</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thesis_title_en</td>
<td>Thesis title (English)</td>
<td>Varchar</td>
<td>1500</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thesis_title_th</td>
<td>Thesis title (Thai)</td>
<td>Varchar</td>
<td>1500</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thesis_type</td>
<td>Research type</td>
<td>Varchar</td>
<td>30</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Tbl_time_dim
**Description:** Store Date/Month/Year (360 days in each year)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time_key</td>
<td>Surrogate key</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>PK</td>
<td>Identity</td>
</tr>
<tr>
<td>Gdate</td>
<td>Every date in each year</td>
<td>Datetime</td>
<td>8</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gmonth</td>
<td>Month of each date record</td>
<td>Varchar</td>
<td>30</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gyear</td>
<td>Year of each date record</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NMonth</td>
<td>Number of month</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Tbl_advisee_inprocess_fact
**Description:** Store detailed data of in-process research operation (student id and professor id)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std_key</td>
<td>Key from student dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Time_key</td>
<td>Key from time dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_key</td>
<td>Key from professor dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Thesis_key</td>
<td>Key from thesis dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_status</td>
<td>Store data where come from OLTP (in-process student)</td>
<td>Varchar</td>
<td>10</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Tbl_defense_committee_fact
**Description:** Store detailed data of professors work load

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std_key</td>
<td>Key from student dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Time_key</td>
<td>Key from time dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_key</td>
<td>Key from professor dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_status</td>
<td>Store data where come from OLTP (defense table)</td>
<td>Varchar</td>
<td>10</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Tbl_grad_student_fact
**Description:** Store detailed data of graduate student

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Size</th>
<th>NULL</th>
<th>Default</th>
<th>Key</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std_key</td>
<td>Key from student dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Time_key</td>
<td>Key from time dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_key</td>
<td>Key from professor dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Thesis_key</td>
<td>Key from thesis dim</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
<td>-</td>
<td>FK</td>
<td></td>
</tr>
<tr>
<td>Prof_status</td>
<td>Store data where come from OLTP graduate student</td>
<td>Varchar</td>
<td>10</td>
<td>No</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX F

METADATA

Physical Data Model for Metadata of the DW with OLAP system
APPENDIX G
User manual

หลังจากที่ติดตั้งโปรแกรม OWC และ PTS ที่เครื่อง client ก็สามารถเปิดโปรแกรม Internet Explorer แล้วพิมพ์ URL ของเครื่อง Web Server ที่เชื่อมต่อกับระบบ Web-base OLAP ได้ ทั้งหมด 1 จากนั้นให้ทำ short cut หรือ add favorites เพื่อจัดการความสะดวกให้ผู้ใช้ในการเข้าระบบครั้งต่อไปได้ ซึ่งหน้าแรกของ web page ที่แสดง จะเป็น metadata ที่ให้ข้อมูลที่ถูกต้องแล้วใช้ เพื่อให้เข้าถึงได้ระบบงานสถิติ หรือ cube เท่านั้นที่ได้สร้างไว้ใน OLAP server ได้ ดังรูป

เมื่อต้องการดูข้อมูลในแต่ละรายการ (1 รายการ/1 cube) ก็สามารถคลิก 1 ครั้ง แล้ว browser จะแสดงข้อมูลสถิติที่เป็นแบบหลายมิติ หลายระดับ ดังแสดงในรูปถัดไป
รูปที่ 2

ภาพที่แสดงดังรูปที่ 2 จะเป็นแบบ Multidimensional view หมายถึง มีข้อมูลด้านเลขานุชุดเดียวแต่สามารถดูได้หลายมิติ เช่น ในตัวอย่างนี้ มีบัคติกาล่างทั้งหมด 53 คน ในจำนวน 53 คนนี้ สามารถมองได้หลายมุม เช่น

- **Major Advisor** ที่ปรึกษาหลัก: ในจำนวน 53 คนนี้ นักศึกษาแต่ละคน มีที่ปรึกษาหลักเป็นใครบ้าง หรือสามารถมองในมุมกลับกันอาจารย์แต่ละคน มี Major advisee ยูคั้น มีรหัสอะไรใครบ้าง และกำลังทำเรื่องอะไร

- **Major and Co-advisor** ที่ปรึกษาหลักและรอง: ในจำนวน 53 คนนั้น นักศึกษาแต่ละคน มีที่ปรึกษาหลักและรองคือใครบ้าง และหากมองในมุมกลับกันจะเห็นว่าอาจารย์แต่ละคน มี Major/Co advisee ยูต้น มีรหัสอะไรใครบ้าง และกำลังทำเรื่องอะไร

- **Research Type** ประเภทงานวิจัย: ในจำนวน 53 คนที่กำลังทำวิจัยอยู่นี้ ท่าน ก. (วิทยานิพนธ์) ท่าน แทน ข. (สารนิพนธ์) ท่าน รหัสอะไรใครบ้างทำเรื่องอะไร และมี Major/Co advisor เป็นใคร

- **Student code** รหัสปีเข้าของนักศึกษา: ในจำนวน 53 คนมีรหัสบ้างที่กำลังทำวิจัยอยู่และแต่ละรหัสมีใครบ้าง มี Major/Co advisor เป็นใคร
รูปที่ 3 แสดงข้อมูล สติคิวาร์ติมานะ ถ้าสามารถมีการเรียงลำดับข้อมูลตามตัวแปรที่สำคัญ)

วิธีการจัดข้อมูล

รูปแบบการแสดงผลที่สามารถดูรายละเอียดได้โดยคลิกที่เครื่องหมายบวก (Drill down) เพื่อขยายรายละเอียดขึ้นไปเรื่อยๆ หรือปุ่มลบ (Roll up) เพื่อปิดรายละเอียดลงมาเรื่อยๆ จากบนสุด.
รูปที่ 4

รูปที่ 4 แสดงข้อมูลสถิติจำนวนอาจารย์ที่ปรึกษาและนักศึกษาในความดูแล ที่กำลังทำวิจัย ทั้งที่เป็น Major และ Co advisor รายงานสถิตินี้ ตอบคำถามว่า อาจารย์แต่ละคนดูแลงานวิจัยอยู่ที่ประเภท ในฐานะอะไร (Major/Co advisor) และ Advisee คือใคร รหัสอะไร เรียงตามลำดับเป็นอันดับ

วิธีการดูข้อมูล

วิธีการแสดงผลที่สามารถดูรายละเอียดถึงลึกซึ้งเกี่ยวกับในฐานข้อมูล ได้ระดับล่างไปเรื่อยๆ จากบนสุด จนถึงล่างสุด ด้วยการคลิกที่เครื่องหมายลบ Referred เป็น Drill down และในทางตรงกันข้าม หากต้องการดูข้อมูลรายละเอียดจากล่างสุดจนถึงบนสุด ได้ระดับขึ้นไปเรื่อยๆ เพื่อดูภาพรวม ด้วยการคลิกที่เครื่องหมายบวก Referred เป็น Roll up
รูปที่ 5

รูปที่ 5 แสดงข้อมูลสถิติจำนวนนักศึกษาที่กำลังวิจัย แยกตามรหัสประจำการศึกษาของนักศึกษา รายงานนี้ตอบโจทย์ที่ว่า มีนักศึกษาระดับใด่บ้างที่กำลังทำวิจัยอยู่ ทั้งแผนไหน (แผน ก /แผน ข)แต่ละแผน มีสาขาอะไรบ้าง และในแต่ละสาขา มีนักศึกษาคือใครบ้าง รวมทั้งมีใครเป็น Major/Co-advisor

วิธีการดูข้อมูล

รูปแบบการแสดงผลที่สามารถดูรายละเอียดได้ลงไปได้เรื่อยๆ จากบนสุด จนถึงล่างสุด ด้วยการคลิกเมาส์ที่ตรงข้ามมา หรือเรียกว่า Drill down และในทางกันข้าม หากต้องการเก็บข้อมูลรายละเอียดจากล่างสุดจนถึงบนสุด ให้ระดับขึ้นไปเรื่อยๆ เพื่อดูภาพรวม ด้วยการคลิกเมาส์ที่ตรงข้ามมาเรียกว่า Roll up
รูปที่ 6 แสดงสถิติจำนวนนักศึกษาที่ยังไม่สอบโครงวิทยานิพนธ์/สารนิพนธ์ รายงานนี้ ด้วยวิธีดูข้อมูลที่สามารถดูรายละเอียดลงไปเรื่อยๆ จากบนสุด จนถึงล่างสุด ด้วยการคลิกเครื่องหมายบวกเรียกว่า Drill down และในทางตรงกันข้าม หากต้องการเก็บซ้อนรายละเอียดจากล่างสุดจนถึงบนสุด โปรดคลิกเครื่องหมายลบเรียกว่า Roll up.

วิธีการดูข้อมูล

โปรแกรมแสดงผลที่สามารถดูรายละเอียดได้เปลี่ยนไปเรื่อยๆ จากบนสุด จนถึงล่างสุด ด้วยการคลิกเครื่องหมายบวก เรียกว่า Drill down และในทางตรงกันข้าม หากต้องการเก็บซ้อนรายละเอียดจากล่างสุดจนถึงบนสุด โปรดคลิกเครื่องหมายลบ เรียกว่า Roll up.
รูปที่ 7 แสดงสถิติจำนวนนักศึกษาที่จบในแต่ละปีการศึกษา โดย group ตามรหัสปีเข้าของนักศึกษา รายงานนี้ตอบโจทย์แต่ละปีการศึกษามีนักศึกษาจบกี่คน จากรหัสใด สาขาใด ชื่ออะไร มีใครเป็น Major Advisor

วิธีการสู่ข้อมูล

รูปแบบการแสดงผลที่สามารถสู่ข้อมูลได้โดยคลิกเข้าไปเรื่อยๆ จากบนสุดขึ้นถึงสุดล่าง ด้วยการคลิกเครื่องหมายบวกเรียกว่า Drill down และในทางตรงกันข้าม หากต้องการเก็บข้อมูลรายละเอียดจากล่างสุดจนถึงบนสุด ให้ระดับเข้าไปเรื่อยๆ เพื่อดูภาพรวม ด้วยการคลิกเครื่องหมายลบเรียกว่า Roll up
รูปที่ 8 แสดงข้อมูลสถิติภาระงานอาจารย์ การเป็นกรรมการสอบวิทยานิพนธ์ รายงานนี้ตอบโจทย์ว่า อาจารย์แต่ละคน มีภาระงานเป็นกรรมการสอบในฐานะ Major advisor, Co-Advisor และ Committee ในแต่ละปีเท่าไหร่ และสอบใครบ้าง ในเดือนไหน วันที่เท่าไหร่

<table>
<thead>
<tr>
<th>ปี</th>
<th>Year</th>
<th>works load</th>
<th>year 2016</th>
<th>works load</th>
<th>Total</th>
<th>works load</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>2614</td>
<td>10</td>
<td>2015</td>
<td>2016</td>
<td>2017</td>
<td>2018</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>5</td>
<td>3</td>
<td>24</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>5</td>
<td>3</td>
<td>43</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

ด้วยการคลิกเครื่องหมายบวก หรือ drill down เพื่อดูข้อมูลลึกซึ้งขึ้น และในทางตรงกันข้าม หากต้องการเก็บข้อมูลขั้นล่างขึ้นไปเรื่อยๆ เพื่อดูภาพรวม ด้วยการคลิกเครื่องหมายลบ หรือ roll up
## APPENDIX H
### QUESTIONNAIRE

The Data Warehousing and an OLAP system  
A case study for Thesis Operations of the school of Nursing,  
Ramathibodi Hospital, Mahidol university

<table>
<thead>
<tr>
<th>Functions</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td>System</td>
<td>4</td>
</tr>
<tr>
<td>1. Ease of use</td>
<td>3</td>
</tr>
<tr>
<td>2. User-interface and design</td>
<td>2</td>
</tr>
<tr>
<td>3. Performance and response time</td>
<td>1</td>
</tr>
<tr>
<td>Report</td>
<td></td>
</tr>
<tr>
<td>4. Usefulness</td>
<td></td>
</tr>
<tr>
<td>5. Clarity, accuracy and integrity of reports</td>
<td></td>
</tr>
<tr>
<td>6. Completeness of functionality</td>
<td></td>
</tr>
<tr>
<td>[The system provide a necessary information</td>
<td></td>
</tr>
<tr>
<td>and meet your need]</td>
<td></td>
</tr>
<tr>
<td>7. Support for semi-structure and unstructured data</td>
<td></td>
</tr>
<tr>
<td>8. Overall satisfaction of reports</td>
<td></td>
</tr>
</tbody>
</table>
BIOGRAPHY

NAME Miss. Srithong Polvises
DATE OF BIRTH 11 April 1971
PLACE OF BIRTH Nakornrachasima, Thailand
INSTITUTIONS ATTENDED Suandusit Institute, 1995:
Bachelor of Art (General Management)
Mahidol University, 2005:
Master of Science (Technology Of
Information System Management)
OFFICE 25/25 M.5 School of Nursing,
Ramathibodi Hospital,
Mahidol University (Salaya campus),
Thailand
Tel. 02-4414234 #312
E-mail: raspo@mahidol.ac.th
HOME ADDRESS 29/110 M.10 Chokchai 4 Road, Ladpraow,
Bangkok