THE DESIGN AND DEVELOPMENT OF A DATA WAREHOUSE USING SALES DATABASE AND REQUIREMENTS OF A RETAIL GROUP

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Software

by
Işıl GÜRATAN

July 2005
İZMİR

We approve the thesis of **Işıl GÜRATAN**

Date of Signature

…… **19 July 2005**

Prof. Dr. Sıtkı AYTAÇ

Supervisor

Department of Computer Engineering

İzmir Institute of Technology

……. **19 July 2005**

Prof. Dr. Halis PÜSKÜLCÜ

Department of Computer Engineering

İzmir Institute of Technology

……. **19 July 2005**

Prof. Dr. Halil ŞENGONCA

Department of Computer Engineering

Ege University

……. **19 July 2005**

Prof. Dr. Muhsin ÇİFTÇİOĞLU

Head of Department

İzmir Institute of Technology

…….

Assoc. Prof. Dr. Semahat ÖZDEMİR

Head of the Graduate School

ACKNOWLEDGEMENTS

I would like to express my special thanks to my advisor Prof. Dr. Sıtkı AYTAÇ for his guidance and insight throughout this study. I also express my sincere appreciation to Instructor Belgin ÖZAKAR for her support, suggestions, and valuable guidance.

Cooperation with the company personnel was of utmost importance while preparing this study. I would like to thank each of them for their feedback and interest.

I am also indebted to my friends Boğaç, Evrim, Kemal, Mine, Utku and Şükrü for their valuable support and encouragement. Besides, my special thanks go to Dehan for her valuable friendship, help and encouragement during my education at İYTE.

Finally, I would like to thank to my parents and my brother for their love, patience and constant support.

ABSTRACT

In the business world, many organizations store, organize and update their records of activities in large databases. As years pass, huge amount of data is gathered within the systems of the companies. As the system grows fast, some problems in the comprehension and analysis of the data collected occur. On the other hand, companies should analyze the accumulated data accurately and get information from them in order to carry out some crucial functions critical to the survival of the company such as planning, forecasting and managing. The difficulties mentioned above have compelled the technology experts to find new solutions to assist the people taking part in the decision process and lacking in technical knowledge. In the recent years, the concept of data warehouse has begun to appear as a new type of complex decision support systems. A data warehouse can simply be defined as a pool of data that supports users in making strategic decisions. These data are organized in a way that will present easy use and excellent reporting performance.

The subject of this study is to design and develop a data warehouse using the necessities and sales database of a retailer company. Firstly, data warehouse and the steps of data warehouse processes are defined, and then applied part is explained. The data warehouse application constituting the subject of this thesis is carried out in a middle-scale company that is in the food sector and has a nation-wide distribution net. This company has approximately 600 employees. In this study, a data warehouse has been designed in order to support the Sales and Human Resources Departments by using the data collected by Sales Field Organization of the company.

The main objective of creating a data warehouse is to enable a better control of the Sales Representatives working in the Sales Field Organization by Sales Managers and provide the gathering of consistent and accurate data that will be used by Human Resources Department while preparing a performance measurement system. These data will also be used by the Sales Department while determining a sales policy for the company. In this study, powerful and easy-to-use MS SQL Server 2000 that includes data transformation and analysis tools is used.

ÖZET

Yıllardır iş hayatında pek çok organizasyon kendi aktiviteleri ile ilgili kayıtlarını geniş veri tabanlarında saklar, düzenler ve günceller. Yıllar geçtikçe işletmelerin sistemlerinde devasa boyutlarda veri toplanır. Sistem büyüdükçe, bu hızlı ilerleme toplanan verilerin anlaşılması ve analiz edilmesinde birtakım problemlerin ortaya çıkmasına sebep olur. Öte yandan firmalar ayakta kalmalarını sağlayan planlama, tahminleme, yönetim gibi kritik fonksiyonların yürütülmesi için bu verileri doğru bir şekilde analiz edip, değerli bilgiler elde etmeye ihtiyaç duyarlar. Yukarıda bahsedilen bu zorluklar teknoloji uzmanlarını karar verme mekanizmalarında görev yapan, genellikle teknik bilgisi olmayan kişilere yardımcı olması amacıyla yeni çareler bulmaya zorlamıştır. Son yıllarda veri ambarı kavramı kompleks karar destek sistemlerinin yeni bir tipi olarak karşımıza çıkmaya başlamıştır. Basit olarak, veri ambarı stratejik kararların alınmasında kullanıcıları destekleyen verilerin tutulduğu bir veri havuzu olarak tanımlanabilir. Bu veriler kullanımı kolay olacak, çok iyi raporlama performansı sağlayacak şekilde organize edilir.

Bu çalışmanın konusu bir perakendeci firmasının ihtiyaçları ve satış veri tabanı kullanılarak bir veri ambarı tasarlayıp, geliştirmektir. Öncelikle veri ambarı ve veri ambarı işlemlerinin adımları tanımlanmakta, ardından uygulama kısmı anlatılmaktadır. Bu teze konu olan veri ambarı uygulaması yurt çapında dağıtım ağına sahip, gıda sektöründeki, yaklaşık 600 çalışanı olan, orta ölçekli bir firmada yapılmıştır. Firmanın Satış Saha Organizasyonu tarafından toplanan veriler kullanılarak Satış ve İnsan Kaynakları Bölümlerine destek olması amacıyla veri ambarı tasarımı gerçekleştirilmiştir.

Veri ambarının oluşturulmasının ana amacı, saha organizasyonunda görevli kişilerin çalışmalarının satış şefleri tarafından kontrolünün yapılabilmesini sağlamak ve İnsan Kaynakları Bölümünün hazırlayacağı performans değerlendirme sistemi için tutarlı ve doğru verilerin toplanmasını sağlamaktır. Bu veriler satış bölümü tarafından firmanın satış politikası belirlenirken de kullanılacaktır. Proje gerçekleştirilirken veri transfer ve analiz araçlarını da içinde bulunduran güçlü ve kolay kullanım imkanı sunan MS SQL Server 2000 kullanılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table

CHAPTER 1

INTRODUCTION

1.1. Definition of the Problem

Retail trade provides the coordination of the transfer of the goods carried from producers to the consumers. Today, retail trade is viewed as the representative of the producers, and at the same time, an authority of guarantee for the consumers. As an addition to the transfer of the products, this sector also determines the due times and the quantities of the products to be delivered. (Tek 1984, p.46)

Most of the market chains were established by the governmental authorities in 1950s, and afterwards, most of them were sold to the private enterprises. Thus, the sector has gained a more competitive identity in the recent twenty years. Retail sector has showed a fast pace of growing in Turkey, and many international brands and companies were aware of this potential.

The number of retailers which was 176.437 in 1996 slightly fell down to 164.593 in 1997. While the number of small markets which was 164.366 in 1999 fell to 148.925, the number of hypermarkets and chain markets rose from 1316 to 2421, and the number of markets increased from 10.755 to 13.247. It can be concluded that there is a decrease in the number of hypermarkets' openings, and the interest towards supermarkets and gross markets tends to rise. The number of the retailers and improvement of the sector between 1996 and 2003 are shown in Table 1.1.

According to the 1998 numbers, hyper and chain markets represent the 28%, markets represent %16, and the small markets represent the 56% of the whole Turkish commercial retail trade. If the gross incomes in year 2000 are analyzed, it will be projected that, %50 of the trade will belong to small markets, %36 to supermarkets and a final %14 will belong to the markets. The number of the hypermarket and the supermarket between 1996 and 2003 are given in Table 1.2.

Table 1.1. The Number of the Retailer and Improvement

(Source: Bocutoglu et al. 1999, p.10)

	1996	1997	1998	1999	2000	2003
Hyper/supermarket	1.316	1.682	2.135	2.421	2.636	3.500
Market	10.755	11.417	12.192	13.247	13.795	16.000
Grocer	164.366	159.171	155.420	148.925	147.715	131.000
TOTAL	176.437	172.270	169.747	164.593	164.146	150.500

Table 1.2. The Number of the Hypermarket-Supermarket

(Source: Bocutoglu et al. 1999, p.10)

	1996	1997	1998	1999	2000	2003
Hypermarket and supermarket	1.316	1.682	2.135	2.421	2.979	3.500
Hypermarket (2500 m²)	37	51	100	105	142	159
Supermarket_1 (1000-2499 m²)	95	135	178	227	302	350
Supermarket_2 (400-999 m²)	289	414	487	571	717	793
Supermarket_3 (100-399 m²)	895	1.082	1.370	1.518	1.493	2.198

1/3 of the retailers are located in three big provinces of Turkey. A significant number of them exist in Marmara Region, but Aegean and Middle Anatolia are other important regions. According to the Table 1.3, 66.692 of the retailers are running their business in Marmara Region while 38.691 are in Aegean Region, 37.070 are in Middle Anatolia, 35.629 are in Mediterranean Region, 18.064 are in Black Sea Region, 15.505 are in eastern and southeastern part of Anatolia Region. Importance of Istanbul comes from both the number of firms and the employees working.

Table 1.3. The Number of the Retailers in the Regions

(Source: Bocutoglu et al. 1999, p.11)

Name of the region	Supermarket	Market	Grocery	Other	Total
Marmara	1.044	5.361	43.669	16.618	66.692
Aegean	375	2.134	28.549	7.633	38.691
Middle Anatolia	388	1.642	27.752	7.288	37.070
The Mediterranean	145	1.165	28.108	6.211	35.629
The Black Sea	110	963	14.311	2.680	18.064
The Eastern- The South Eastern Anatolia	73	927	13.031	1.474	15.505
TOTAL	2.135	12.192	155.420	41.904	211.651

The increasing competitiveness in retail business has brought about new habits. In such a fast increasing competitive market, differences of the companies in technology and service quality are getting minor day by day. The parameter of the main difference in the market stems from creating a common value for the vendor-retailer-consumer relationship. This emphasized rise in retail sector has supported the demand of the various industries and services.

The rise in the number of the firms in charge of manufacturing different productions and performing services forced the managers to use recent and modern technologies, specialize on field and category management, and develop their systems of reaching the electronical data and the consumer efficiently. Market chains are struggling to create their “faithful consumers”. Personnel training, service quality, cost control and determining the behavior patterns and the shopping habits of the consumers are the key elements in this challenge.

As the number of the firms in the market increased, it became harder to differentiate between the firms that supply similar products and services. Most important issue in this differentiation became the marketing and advertising activities. Firms, hoping to create differences, launched great amounts of TV and newspaper advertising campaigns, and tried to increase their incomes. On the other hand, these campaigns were replied with controversial campaigns by the competitors, so the amount

of the monetary outcome of the all companies got bigger, and the positive results of these campaigns could not be obtained.

If the expenses are corrected according to the sales regions and then rated to the sales revenues, it will be seen that the competition works negatively for the profit margins. Despite showing slight waves, it is seen that the ratio of marketing and distribution expenses to the net sales is growing due to the competition. On the other hand, since it is getting harder for the firms to grow in big cities, they are moving to the rural areas where logistics activities is surely harder, and this causes an additional increase in distribution expenses. So, the companies began to seek the most efficient ways of using wider sales and distribution systems and tracking.

One of the most fundamental missions of the sales-logistics organizations is to provide the products and services periodically and continuously in accordance with the given channel politics and merchandising standards. This mission is the basic performance criterion from field specialists' liability area to the district field as the organization structure. In Turkey, under the present retail trade circumstances, companies come across many problems in practicing their determined logistics & merchandising standards and continuing them.

The company mentioned in the project has a distribution network, and the field specialists of the firm, representing the smallest authority range of the organization, pay visits to the customers, and they collect data via their palm computers. Flow of data can hardly be continued if the continuous tracking of the distribution organization cannot be maintained. It is necessary to process the knowledge and convert it to "meaningful information".

In addition, the data belonging to the previous years should be reached easily and quickly while the statistical reports are populated. As we climb to the upper levels of the organization, it is seen that the authorized managers face many difficulties in reaching some critical data and using them in their decision support processes.

The company is in need of an efficiently working structure that is configured to measure and manage the distribution performance. To solve these problems, it is planned to start from the lowest level of the hierarchy, and as lower levels are improved, top levels would be reached more quickly and easily.

Field personnel are the "mirrors" of the company for the consumers. Therefore, their performances must strictly be tracked to see whether the performances are in accordance with the specifications and rules of the firm's distribution and logistics

channels or not. Field chiefs are responsible for the control of the field personnel. Under these circumstances and due to these reasons, it is aimed to establish an information system for the analysis and management of the performance of the sales and distribution personnel.

Information management system to be established will be in harmony with the environment and it should be able to find organizational management solutions according to the information technologies. Data will be modified and transformed into various forms for the company. At this point, data warehouses, which have gained a high popularity in recent years and have been introduced as a new solution for complex decision support systems, occur as an answer.

1.2. Aims of the Study

High-level managers and specialists often give critical decisions affecting the future of the companies. Large amounts of data are kept to be reached and modified to obtain the appropriate information under some given rules in the future. Databases are only used in the keeping phase of this process. The next part of the problem is how to obtain correct, appropriate and useful information. Data warehouses are developed to solve this part of the problem.

The purpose of this study is to define a data warehouse, draw a roadmap for its preparation, and design and implement this warehouse according to the requirements of a company in retail trade sector.

First, it is aimed to study the requirements and the problems of the firm and analyze the given operational system. After determining the data needed under the scope of required information, the second step is to prepare a database design by using MS SQL Server. As a following step, the data are transferred, and finally it is aimed to present an analysis opportunity of the information via MS SQL Server OLAP tool to the client.

As the last step, an information system structure (to be used in decision support systems) that can keep previous data and measure the product distribution performance by backloggings and penetration as well as a system that can measure the merchandising criteria such as price label penetration, cooler standards and promotion penetration is formed.

1.3. Outline of the Study

This study has three parts organized according to the problem and the aims mentioned above.

In Chapter 2, the concepts of data warehousing are explained.

After this introduction to data warehousing, in the next chapter, the process of designing a data warehouse is given. A brief explanation of data loading and transformation is also given in this part. A general knowledge about Online Analytical Processing (OLAP), features of OLAP and storage architecture techniques also take place in Chapter 3.

In order to give an example of creating a data warehouse to assist users in decision support mechanisms, an application is carried out, and this study is explained in Chapter 4. In this chapter, all steps of designing the data warehouse for performance management system of chosen company are presented.

The flow of the project's process is explained briefly in the discussion part.

Finally, the summary of this thesis and some future work are briefly presented in the conclusion chapter.

CHAPTER 2

INTRODUCTION TO DATA WAREHOUSING

2.1. What is a Data Warehouse

A data warehouse is storage of convenient, consistent, complete and consolidated data, which is collected for the purpose of making quick analysis for the end users who take place in Decision Support Systems (DSS).

These data is obtained from different operational sources and kept in separate physical store. A data warehouse is not only a relational database that contains historical data derived from transactional data but also it is an environment that includes all the operations and applications to manage the process of gathering data, and delivering it to business users such as extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools.

Data warehouses have no standard definition and the people who work on data warehouse subject have defined it in many ways as follows:

"The basic data warehouse architecture interposes between end-user desktops and production data sources a warehouse that we usually think of as a single, large system maintaining an approximation of an enterprise data model." (O'Donnell 2001)

"A data warehouse is a copy of transaction data specifically structured for querying and reporting." (Kimball et al. 1998, p.19)

William Inmon defined a data warehouse as a "subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process". (Lane and Schupmann 2002, pp.42-43)

As set forth by William Inmon:

- **Subject-Oriented:** Data warehouses are designed to aid in decision making for a specific subject. For example, sales data for applications contains specific sales of specific products to specific customers. In contrast, sales data for decision support contains a historical record of sales over specific time intervals. If designed well, subject-oriented data provides a stable image of business processes, independent of legacy systems. In other words, it captures the basic nature of the business environment.

- **Integrated:** Data warehouse consists of different kind of data which are collected from separate legacy systems and this can create conflicts and inconsistencies among units of measure. Because of this, they have to be put in a consistent format and by this way they become integrated.
- **Nonvolatile:** Nonvolatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable a user to analyze what has occurred. New data is always appended to the database, rather than replaced. The database continually absorbs new data, integrating it with the previous data.
- **Time variant:** There is difference between operational data and informational data from the point of time variancy. Operational data is valid only at the moment of access-capturing a moment in time. When performance requirements are demanded, historical data is needed. Due to the data warehouse data represents data over a long time horizon; historical analysis can be easily performed. (Lane and Schupmann 2002, pp.42-43)

2.2. The Goals of a Data Warehouse

The fundamental goals of the data warehouse are (Kimball et.al. 1998, pp.10-11):

1- "Makes an organization's information accessible." The contents of the data warehouse are correctly labeled and obvious. It is very easy to reach to data because they are one click away and there is no need to wait for this. These properties are called as same in the above order; understandable, navigable and fast performance.

2- "Makes the organization's information consistent." Consistent information has a key importance for the data warehouses since they get data from different parts of an organization. They have to be matched properly. If two measures of the organization have the same name then they must mean the same thing. Conversely, in two measures don't mean the same thing, they are labeled differently.

3- "To be an adaptive and resilient source of information." It enables to add new data and ask new questions without any change in existing data and the technologies due to it are designed for continuous change.

4- "To be a secure bastion that protects owner's information asset." The data warehouse not only controls access to the data effectively, but also gives its owners great visibility into the uses and abuses of that data, even after it has left the data warehouse.

5- “To be the foundation for decision-making.” The data warehouse provides the right data for the decision makers. The decisions are output of the data warehouses.

2.3. Data Warehouse Applications by Industry

The data warehouse applications can be performed in the following industries. (Moss and Adelman 2000, pp. 315-316)

- Consumer Goods
- Distribution
- Finance and Banking
- Finance - General
- Government and Education
  - Federal Government
  - State Government
  - University
- Health Care
- Hospitality
- Insurance
- General
- Health Insurance
- Workers Compensation
- Manufacturing and Distribution
  - General
  - Appliance
  - Automobile
  - Clothing
  - Computer
  - Food
  - Pharmaceutical
  - Steel
- Marketing
- Multi-Industry (conglomerates)
- Cross Industry
- Retailers
- Services
- Sports
- Telephony
- Transportation
- Utilities

Below there are three samples with what the data warehouse provides for each of them.

A Retailer Company

- Sales analysis
- Forecasting
- Inventory tracking

- Market basket analysis - e.g. Do products on sale generate other sales?
- Individual items contribution to profits
- Vendors (suppliers) have access to how their products are selling
  (Moss and Adelman 2000, p.330)

  Manufacturing – Textile
- Supply chain measurement and analysis
- Customer profitability
- Product profitability analysis
- Strategic partnering - Negotiating with suppliers, customers
- Customer purchases to identify who should get new Material Safety Data Sheets
- Contract negotiation
- Profitability by business
- Facility specific analysis for productivity and quality
  (Moss and Adelman 2000, pp. 332)

  A Food Manufacturer Company
- Measure against competition
- Ability to project how a new product will do
- Sales analysis by region and by store
- Ability to show grocery managers how product is selling at competitive stores, at stores within the same chain and against competitive products
- Using agents, monitors conditions that require attention including variances in prices and volumes in company's and competitors' products
- Analyses fixed costs, equipment utilization
- Analyses manufacturing costs and performance
- Analyses productivity
- Analyses inventory levels
- Planning and forecasting
  (Moss and Adelman 2000, p.327)

2.4. Basic Elements of the Data Warehouse

2.4.1. Source System

A source system is called as legacy system that captures business data and transactions. Source system has to be uptime and available and it gives a chance to share basic dimensions as product and customer with other legacy system in the organization. It is the largest source of data for analysis systems therefore it is a burden to create queries and management reports directly from these systems.

2.4.2. Data Staging Area

A data staging area is an initial storage area where set of processes- that clean, transform, combine, de-duplicate, household, archive- are performed on the data in order to use them in the data warehouse. The data staging area acts as a bridge between the source system and presentation server. Data staging area can be spread over a number of machines and does not need to be based on relational technology. Unlike the presentation service, which will be describe below, the main restriction of data staging area is that it never provides query and presentation services.

2.4.3. Presentation Server

A presentation server is a physical machine that stores the processed data for the end user's querying and reporting requirements. It is fed from data staging area. If the queryable presentation resource for an enterprise's data organizes around an entity-relation model, understandability and performance will be lost. Also the tables will be organized as star schema if the presentation server presents and stores data in a dimensional framework.

2.4.4. Dimensional Model

Dimensional model, which is designed to provide higher query performance, resilience to change and to be more understandable, is an alternative model to entity-relation model. The dimensional model consists of fact table and dimension tables.

A fact table contains measurement of the business that is preferred to be numeric and additive. There has to be a set of two or more foreign keys that helps to join dimension tables to fact table.

A dimension table is complementary to the fact table. Most of them have many textual attributes. It also has primary key enables to make a relation with the fact table.

2.4.5. Data Mart

Data mart is a logical subset of the complete data warehouse and prepared for a single business process in an organization. When they come together, an integrated enterprise data warehouse is formed. Data marts must be built from shared dimensions and fact. By this way they can be combined and used together.

2.4.6. OLAP (On-Line Analytic Processing)

OLAP enables querying and presenting text and number data from data warehouses for end users. OLAP technology is based on multidimensional cube of data and OLAP databases have multidimensional structure.

2.4.7. End User Application

These applications help end users to prepare queries, make analysis and perform other activities which are targeted to support business needs such as end user data access tool and ad hoc query tool.

End user data access tool works with SQL session and provides to the user a report, a screen of data or another forms of analysis.

Ad hoc query tool facilitates preparing queries by given an opportunity to the user to use pre-built query templates.

2.4.8. Modeling Application

Modeling applications enable to transform or make a summary from the data warehouse by forecasting models, behavior scoring models allocation models and data mining tools.

2.4.9. Metadata

Metadata contains information and definitions about the data, which is stored.

The basic elements of the data warehouse are given in Figure 2.1.

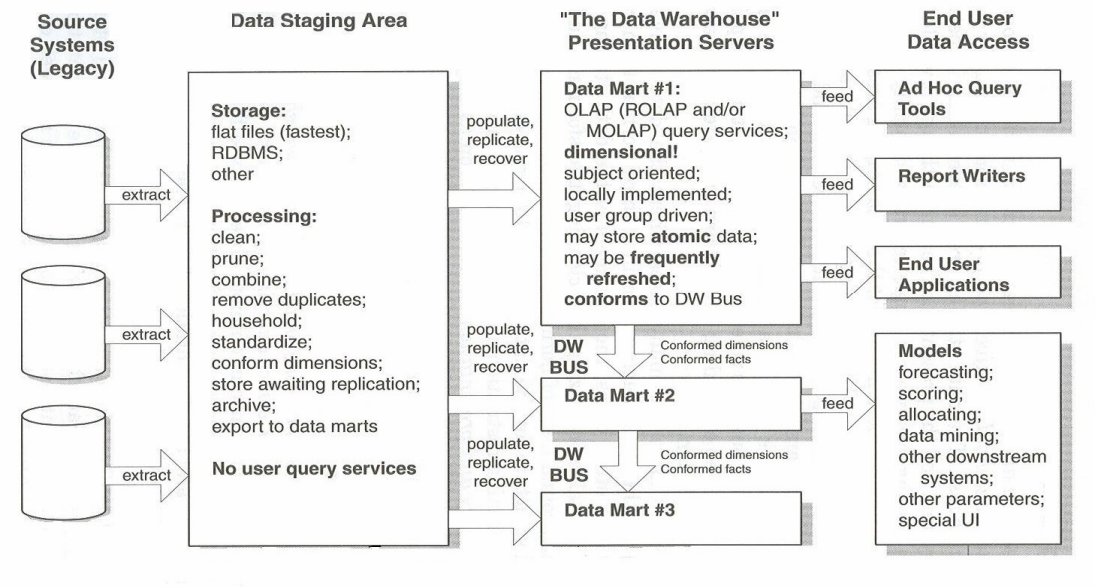


Figure 2.1. The Basic Elements of the Data Warehouse

(Source: Kimball et al.1998, p.15)

2.5. Data Warehouse Architectures

Data warehouses and their architectures vary depending upon the specifics of an organization's situation. Three common architectures are: (Lane and Schupmann 2002, pp. 45-47)

- Data Warehouse Architecture (Basic)
- Data Warehouse Architecture (with a Staging Area)
- Data Warehouse Architecture (with a Staging Area and Data Marts)

2.5.1. Data Warehouse Architecture (Basic)

By this simple architecture for a data warehouse seen in Figure 2.2, end users directly access data derived from several source systems through the data warehouse.

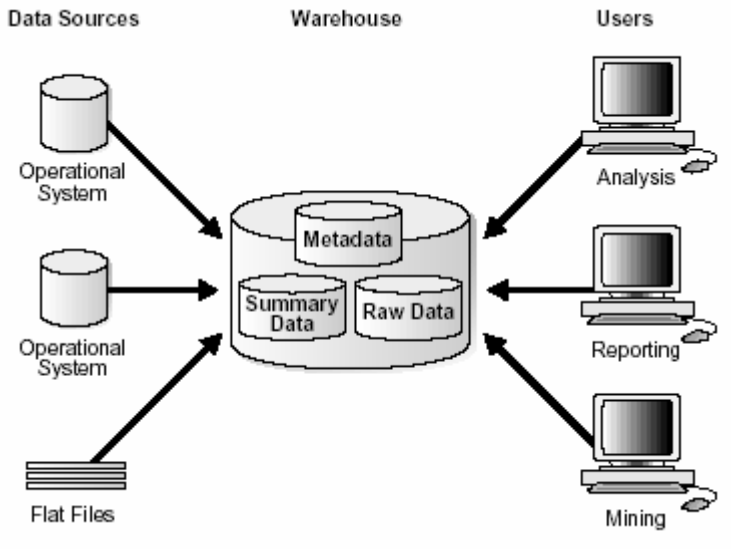


Figure 2.2. Architecture of a Data Warehouse
(Source: Lane and Schupmann 2002, p.45)

An additional type of data, summary data is very valuable in data warehouses because they pre-compute long operations in advance. For example, the result of the query that is about sales of last year is retrieved by adding sales data.

2.5.2. Data Warehouse Architecture (with a Staging Area)

The most data warehouses use a staging area in order to clean and process the operational data before putting it into the warehouse. A staging area simplifies building summaries and general warehouse management. The quite common architecture is shown in Figure 2.3.

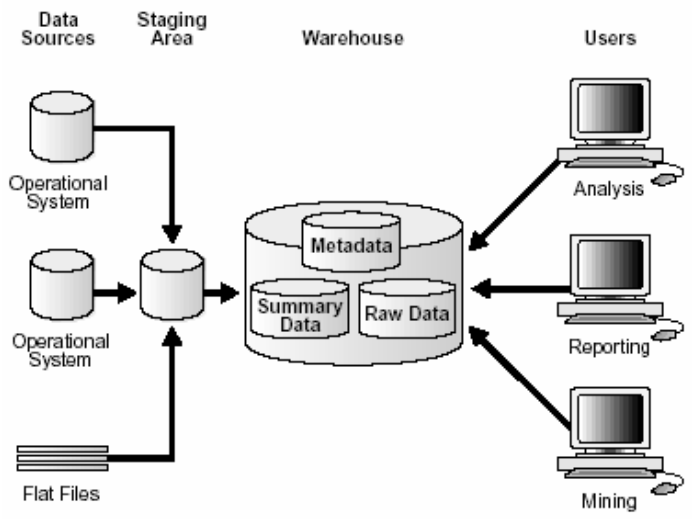


Figure 2.3. Architecture of a Data Warehouse
with a Staging Area
(Source: Lane and Schupmann 2002, p.46)

2.5.3. Data Warehouse Architecture (with a Staging Area and Data Marts)

A warehouse's architecture can be customized for different groups within the organization by adding data marts, which are systems designed for specific parts of business.

The following Figure 2.4 shows an example. In this example, there are three data marts which are designed separately for purchasing, sales, and inventories. This architecture gives an opportunity to analyze historical data for purchases and sales.

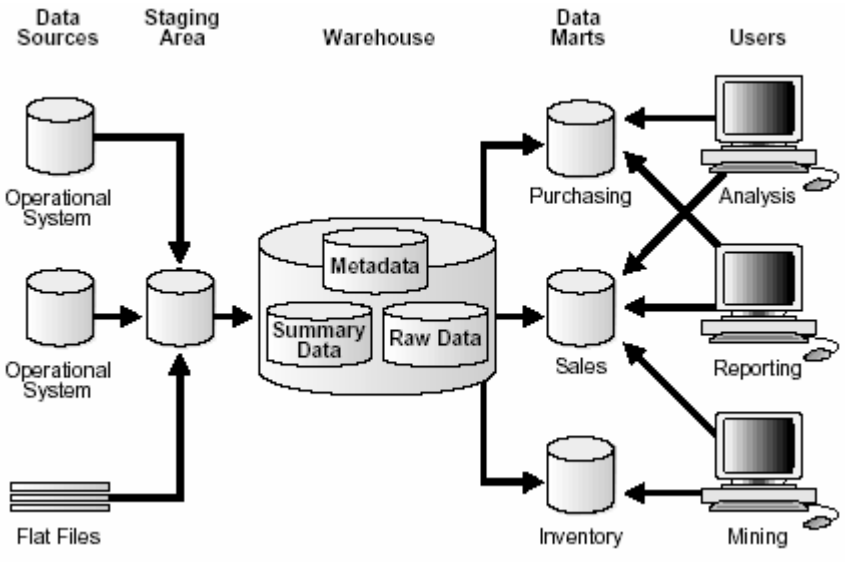


Figure 2.4. Architecture of a Data Warehouse with a Staging Area and Data Marts
(Source: Lane and Schupmann 2002, p.47)

2.6. Data Mart

2.6.1. What is a Data Mart

The data mart is a model, which represents the same data structure with the data warehouse. They are prepared for specific requirements of the whole organization or a part of it. The data mart contains less data that gives to users some advantages. Firstly it enables to work with faster queries. Another advantage is mobility due to it requires less hard disk space so the user can carry the data mart with the laptop. During the designing process of the data marts, it is possible to follow up two different methods in order to collect the data. One option is to collect the granular data from the enterprise data warehouse and then process it according to the needs around which the data mart was prepared. The second option is to collect shaped data directly to the data mart. The data, which is designed up to the requirements of data mart, then is kept in the central repository of all enterprise data. In Figure 2.5 the options can be seen.

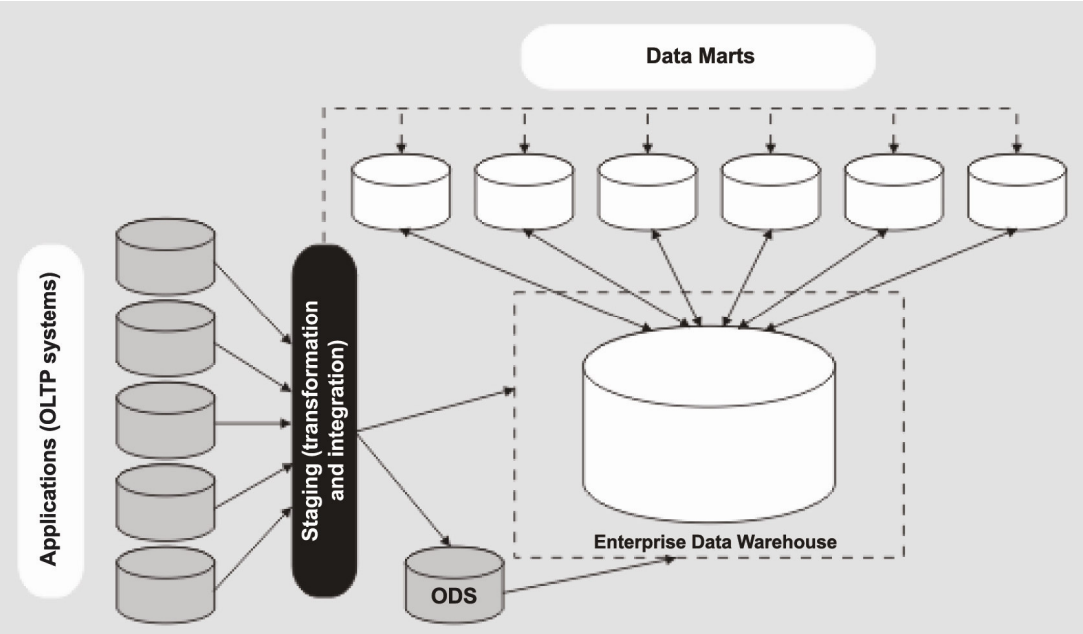


Figure 2.5. Enterprise Information Architecture (Source : Bain et al. 2000, p.76)

Data marts can have dependent or independent structure. If the characteristic of the data marts' dimensions are defined at the beginning, as they would be compliant to each other then these data marts will have dependent characteristic.

In some situations it is better to have independent data marts. This time the characteristic of the other data marts will not take in the consideration during the preparation of the data mart. However, this can prevent future integration and add development cost if there will be an interest in sharing information across departments.

2.6.2. Data Marts Development Approaches

There are three main approaches for building data marts; top-down approach, bottom-up approach and federated approach. (Bain et al. 2000, pp. 79-83)

2.6.2.1. Top-Down Approach

As shown in the Figure 2.6 below the data firstly comes to the data staging area from the operational sources and in this area some of the processes are performed to the data. After this it is transferred to the data warehouse which then feeds it to the dependent data mart.

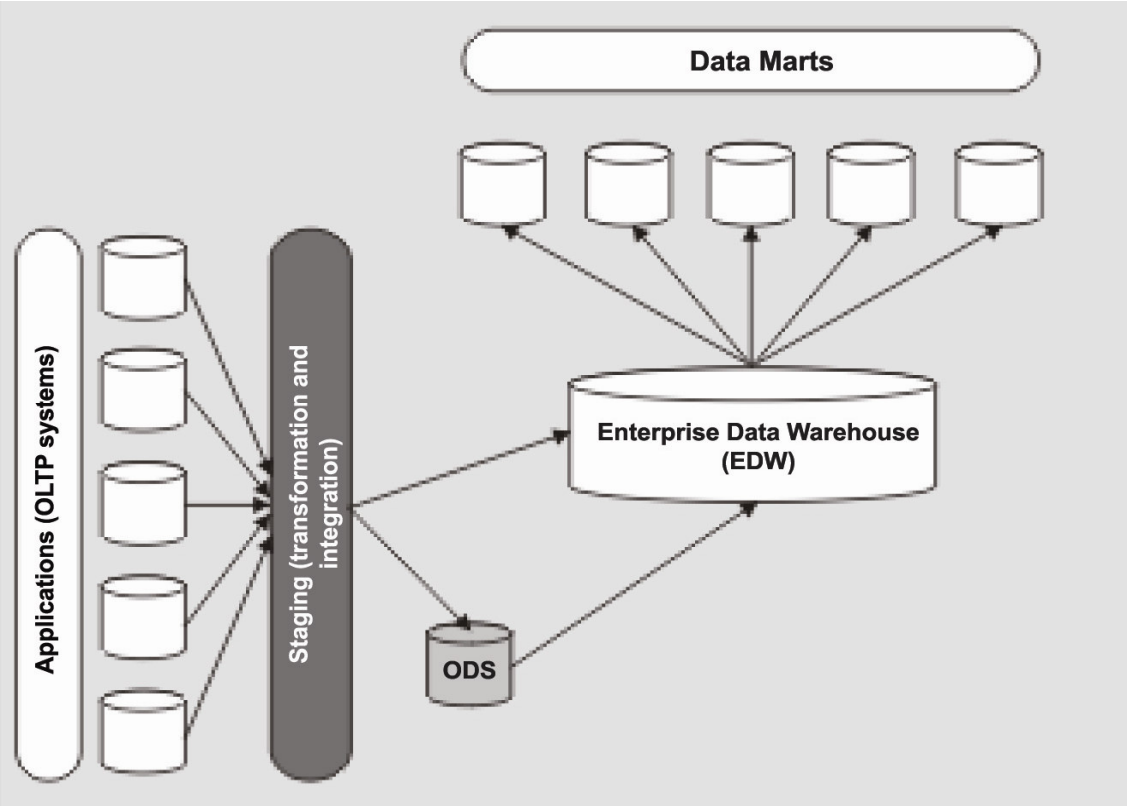


Figure 2.6. Top-Down Approach to Data Mart Development
(Source: Bain et al. 2000, p.80)

2.6.2.2. Bottom-Up Approach

In this approach, the data, which comes from legacy systems to the staging area, flows directly into the independent data marts and then these data marts feed the enterprise data warehouse as it is illustrated in Figure 2.7.

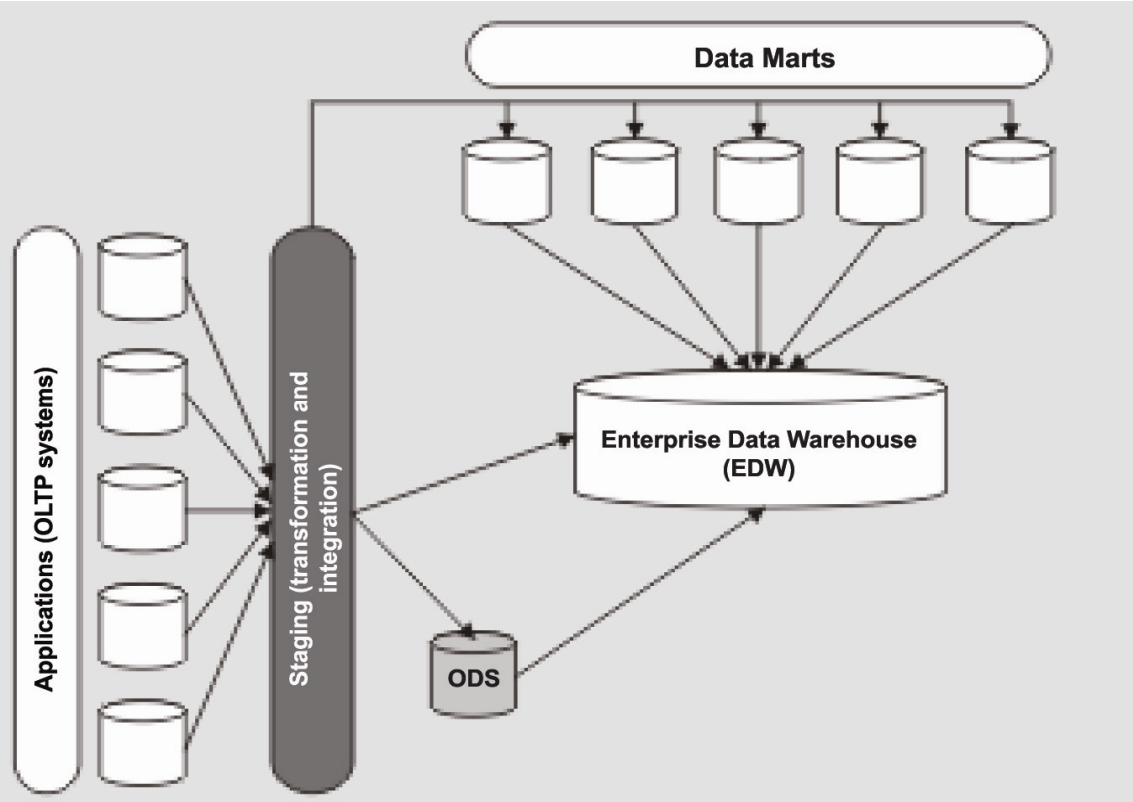


Figure 2.7. Bottom-Up Approach to Data Mart Development
(Source: Bain et al. 2000, p.81)

2.6.2.3. Federated Approach

With the federated approach, there are both dependent and independent data marts as shown in Figure 2.8. The independent data mart gathers the data from the staging area directly, and the dependent data marts get the data from the data marts.

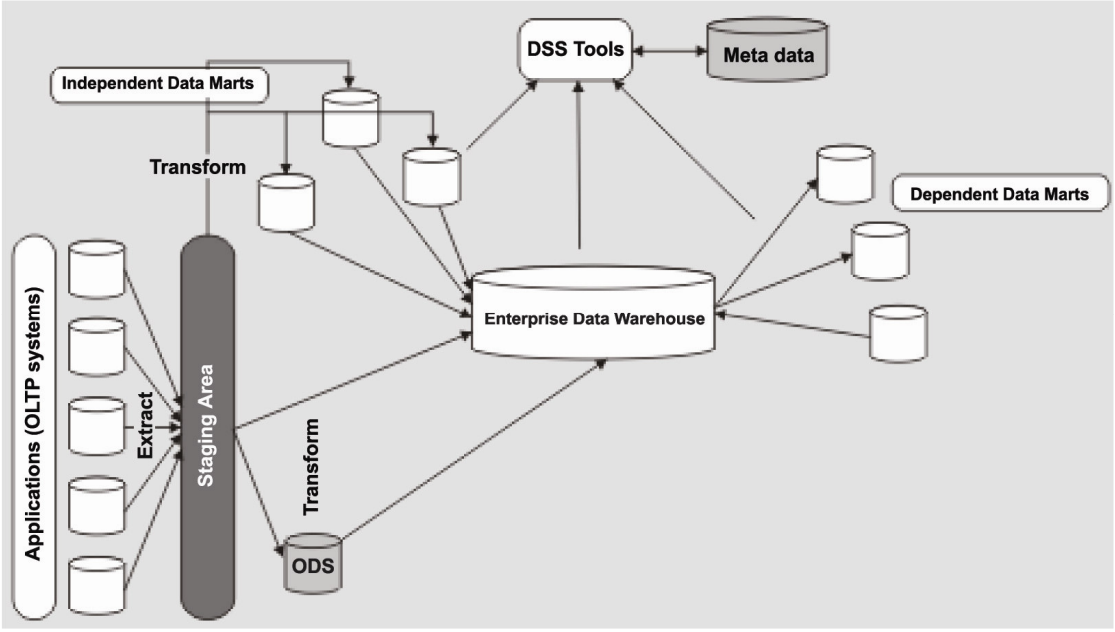


Figure 2.8. Federated Approach to Data Mart Development
(Source: Bain et al. 2000, p.82)

2.6.3. The Differences between Data Mart and Data Warehouse

When the data mart is compared with the data warehouse, two fundamental distinctions can easily be noticed. One of them is that data mart is a subset of the data warehouse and it is requirement-oriented. Against this data warehouse holds the enterprise data without taking care about any specific requirements. But of course, during the design of data mart the structure of the whole warehouse has to be considered, if not it will be very hard to integrate the data marts later.

The implementation of the data mart is much faster and costs cheaper, since a data mart contains only a specific part of the data warehouse whose implementation is more time-consuming and costs much more.

There are some data mart solutions that are developed by the many decision support systems (DSS) vendors. But using them to design a data mart for the specific requirements needs to spend much more effort to customize them; due to this solutions are produced for general purposes.

The other main difference of the data mart from the data warehouse is that the data in the data mart can be more granular than the data warehouse. Since the requirements of the data mart are more defined than those of the data warehouse,

preaggregation can be afforded to the data along the requirements. So the extraction of the data can be done faster and more efficient.

CHAPTER 3

DESIGN AND DEVELOPMENT OF A DATA WAREHOUSE

3.1. Pre-Requirements

A well-designed project creates less problem and this causes optimum resource usage and less time loose. Due to this reason, design step has a vital importance for the project life cycle. The following Figure 3.1 shows a life cycle of a data warehouse project. As it is shown in the diagram cycle starts with defining and gathering of the requirements. After requirements are defined, it is possible to prepare a prototype in order to give a decision about the best design. Then the data analysis and design can be performed with development, testing and documentation. As a result of test, if there is any bug the following step will be removing of them and then the project is released but this doesn't mean the project is completely finished. There can be some modifications, which has to be done according to clients' requirements or organizational needs. (Bain et al. 2000, pp. 124-128)

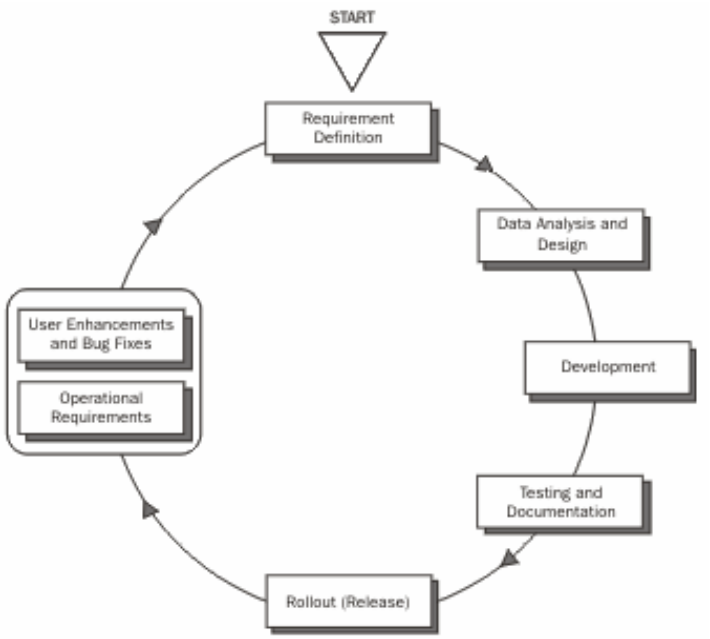


Figure 3.1. The Life Cycle of Defining and Gathering Requirements
(Source: Bain et al. 2000, p.124)

As the data warehouse become functional, additional requirements can appear which were not considered at the beginning so it is very important to get every details about all possible operational requirements, such as user interface changes that might affect the data structure, who will be using the data warehouse, and where it is to be deployed, etc. in the design stage.

3.1.1. Client Side

During the data-gathering period it is very important to talk with right person about right subject in order to get right information. Firstly, the executives are informed about the importance of a data warehousing project and how they will use the system. Then it is time to talk with the people who will produce these reports. They don't have to be from IT department, they can be member of any other department such as secretary. The important subject about selection of these people is that they must be the ones who can provide best information about the business and collected data. In the future, these people will become user of the data warehouse.

In order to select the essential technology which will be used for the data warehouse project, it is necessary to communicate with the information technology (IT) people in the client's organization.

Also it is very important to talk with the people who are using current production system which will feed the data warehouse. This interview will be helpful during the design of the data transformation system from their data store to the data warehouse. (Moss and Adelman 2000, pp. 13-15, Bain et al. 2000, p.125, Kimball et al. 1998, pp. 95-111)

3.1.2. Selection of the Project Team

For developing successful data warehouse project, the project team should be selected very carefully, since project members will affect directly the performance of the project. To have experts in the team will provide positive contributions such as their guidance and anticipation, saving time in the different stages of the project. This kind of expert can be outside consultants but this time it is possible to face some problems. For example, maybe they cannot understand users' requirements properly or after the project is completed and these consultants leave; it is time to solve all the problems by

the local sources of the organization because outside sources cannot be available all the time when they are needed. So it is better to have well-informed specialists in company to support the applications at all times this project runs. Of course when a project team is assembled, this point shouldn't be considered that to keep these project team members together during project life is very significant for the accomplishment of project.

3.1.3. Choosing the Tools

To develop a successful data warehouse project, the next point is having convenient tools which are compound of hardware and software.

Hardware

From the point of hardware it is important that capable machines which allow performing all processes such as data transfer and store the data in the system. The data in the warehouse should be backed up to other media that can be kept and protected off-site, for use when database recovery is needed. The network side of the hardware elements is also so significant due to the performance of the network effects data transfer rate, and provide successful distribution to every end user.

Software

Software side of the project consists of operational database systems, data warehouse database management system, analysis services that used as an OLAP technology, the data transformation tools and client tools which are used by the end users for reporting and querying of the data.

When feeding the data warehouse, the data, which is stored in the operational database, should be easily accessible and queryable. The data warehouse DBMS should be open to data corrections and has enough capacity to carry the immense amount of data. In order to transfer the data from its source to the target data warehouse properly and quickly, the data transformation tools have key role so they should be so efficient. The Analysis Service should be powerful and capable of performing aggregations and summary tables quickly. Finally, the client tools should be user- friendly and allow powerful tables and graphs of the data to be built in response to the user's requirements.

3.2. Designing the Data Warehouse

After the determination of the pre-requirements, before building the data warehouse the strategy has to be designated.

The strategy, which will be valid during the whole project, consists of following basic components:

Policy: It includes the rules which arrange administration and usage of the data warehouse such as maintenance frequency and security authorization.

Transformation: The data, which will be transferred from the operational source, has to be changed into a consistent form that fits to the data warehouse system by applying some processes like cleaning and transforming.

Meta data: This is the information storage of the data concerning the data warehouse system. Meta data can be used in future again and again.

Storage: Data storage in the warehouse that includes transformed data should be flexible, accessible and manageable. (Bain et al. 2000, p.128)

3.2.1. Requirement Analysis

A data warehouse project starts with collecting the requirements of client. To get good results a convenient interviewing plan should be followed up. By this plan, in the organization it will be possible to talk with executives in order to get their expectations and to identify the people who can make positive contributions to the project.

To learn the current business system and its rules and to get the useful data, the right questions should be asked to the right people. Thus the measures and the dimensions, which will take place in dimensional model of data warehouse, can be easily and properly identified. For instance, the client management and/or business analysts can tell that they are interested in knowing about total sales by region, data, salesman, customer, etc. They may also tell that they are interested in knowing their inventory and its relationship to the number of orders at any given time. If during the designation of these measures and dimensions any mistake is made then this will mean that the project has to turn back to the beginning.

Business Requirements

To determine the business requirements means to identify the goals of the data warehouse or the data mart and designate the scope of the project or other subjects which are necessary according to the clients.

Business Objectives

Business objectives have two characteristic: broad objectives and specific objectives. To improve customer service can be given as an example for the broad objectives. Besides to determine the inventory level of products for given period and given store can be an example for the specific objectives.

Sources of Data

To fix the useful data source and prepare the convenient plans for gathering data and sorting out necessary ones in order to carry them to the data warehouse database has vital importance.

As a result of identifying good business requirements, specifications for the data warehouse can be defined in terms of the following:

- Subject Areas: Topics of interest to various business functions. "As an example, the marketing department may have interest in one or more of the following topics: market research, competitive analysis, buyer behavior, market segmentation, product (market matching), pricing and budgeting decisions, product decisions, promotion decisions, channel decisions, and forecasting trends." (Bain et al. 2000, p.131)
- Prototyping: This is an important step that will yield specific subject areas. "In the example of the marketing department, specific subject areas can be designated such as orders, promotions, markets, sales, and time cycle." (Bain et al. 2000, p.131)
- Granularity: The higher the granularity, the more the amount of detail increases. This is defined by increasing the number of levels in the dimensions.
- Dimensions: Based on the analysis of the requirements, the following dimensions could be recognized:

    - Time with a calendar hierarchy (day, week, month, quarter, year)
    - Customer groups (customer, market segment, market, industry)
    - Product families (product, product family, product line)
    - Geography and location (sales rep, territory, district, country region, country, international region, corporate)

- Organization structure (department, business unit, division, strategic business unit, subsidiary, corporation) (Bain et al. 2000, p.131)

End-User Requirements

The customer of the data warehouse system will be the end user because of this reason; to design the system according to the requirements is so important. End users want to prepare queries and reports, to browse the data warehouse data, to view the models that they build and to make data analysis easily and without facing any problem. Briefly, it is very important to build a user- friendly GUI that the user approves of and finds easy to use.

3.2.2 Modeling the Database

3.2.2.1. Data Modeling Techniques

To build and create a database, to construct a data model, which represents the business process, its attributes and relationships between them is the first step.

3.2.2.1.1. Entity Relation (ER) Models

The Entity Relation (ER) model technique makes a search to eliminate the unnecessary data. ER modeling is based on two basic concepts: entities and the relationships between those entities. ER models are used to produce a data model for the specific area of interest. The Figure 3.2 shows a representation of a typical ER diagram. (Kimball et al. 1998, pp. 140-144, Bayoğlu 2000, pp.48-50)

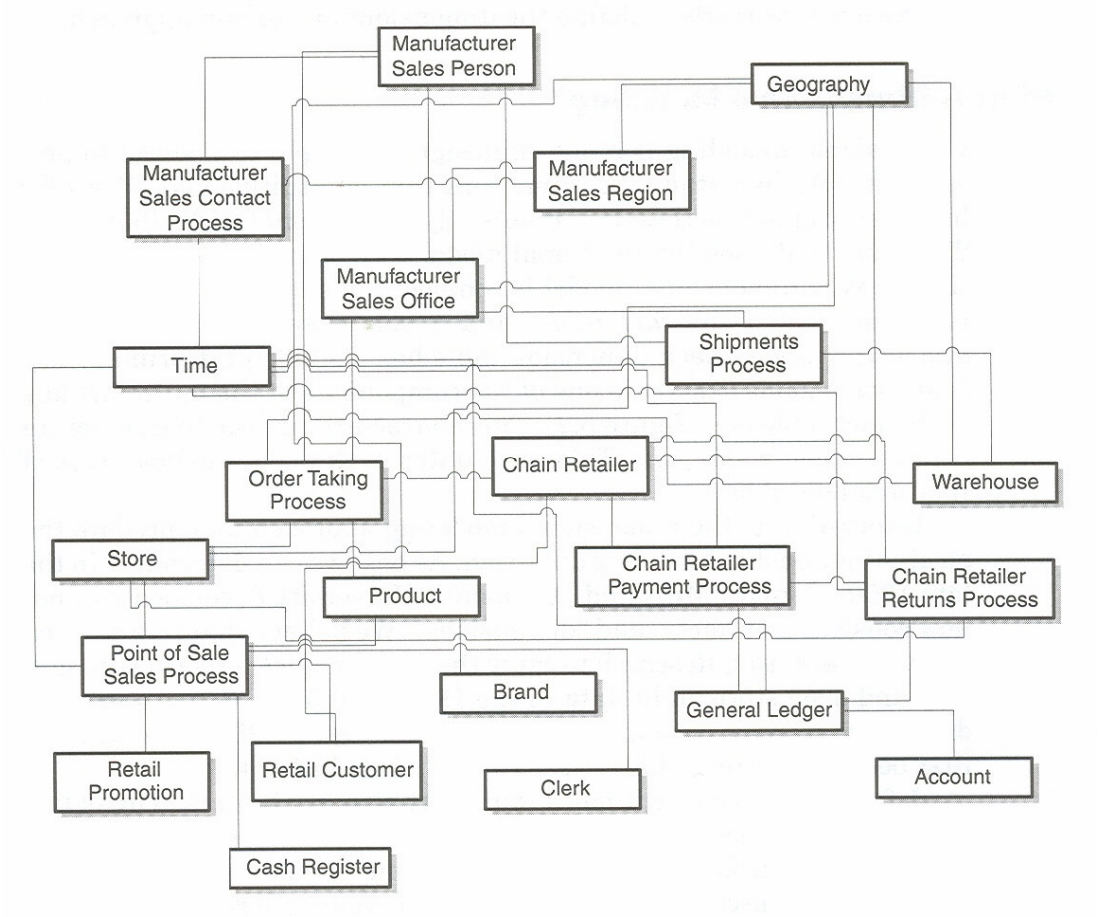


Figure 3.2. An ER Model of an Enterprise that Manufactures Products, Sells Products to Retailers and Measures the Retailers' Sales
(Source: Kimball et al. 1998, p.143)

Entities represent real world objects and they have characteristic attributes which separate them from the other objects. In order to describe what is included and what is not they have their own definitions. Every entity has an unique identifier.

Entities represent real-world objects that can be observed and classified by their properties and characteristics. An entity has its own business definition and a clear boundary definition that is required to describe what is included and what is not. In the detailed ER model, it is critical for entity to have unique identifier, which is called candidate keys that used to select the key that referred to the primary key.

Characteristics of properties of the entities are given by their attributes. An attribute has an unique and self-explanatory name.

An entity and an attribute in the logical model match with respectively a database table and a table column in the physical model.

A domain, in the database structure, is arrangement of valid values and categories according to the attributes; for example the data type such as float, date, and character, provides a clear definition of domain.

If an attribute is a unique identifier of the record in the table and not null, it can be used as a primary key. Also more than one attribute can come together and form one single primary key together, which is called as composite primary key.

A foreign key enables to create a relationship between two entities that are represented by different tables. But in this case one of the entities should have the same attribute as its primary key.

If the entity relationship modeling technique is used for modeling the data warehouse, some disadvantages can be occurred. It is impossible that to present immensely complex schemas in ER Model to end-users. From the point of end users these model are not understandable and navigable and also can be easily forgotten. There is no graphical user interface that allows user to browse the model. An entity relationship model cannot be usefully queried. To retrieve the data intuitively and effectively that is the aim of the data warehousing cannot be performed by using the entity relationship modeling technique.

3.2.2.1.2. Dimensional Modeling

Dimensional modeling is a logical design technique that tries to present the data model in an accessible and intuitive standard frame. This data is visualized as a set of measures that are defined according to the business. This model, which keeps numeric data, such as values, counts, weights, balances, and occurrences in the context of database tables, summarizes and rearranges the data and presents views of the data usefully to support data analysis.

In data warehousing, dimensional modeling is simpler, more expressive, and easier to understand than ER modeling. Unlike ER modeling, dimensional modeling is powerful in representing the requirements of the business user. (Kimball et al. 1998, pp.144-147, Bayoğlu 2000, pp.50-53) There is an example of a dimensional model of an enterprise in the following Figure 3.3.

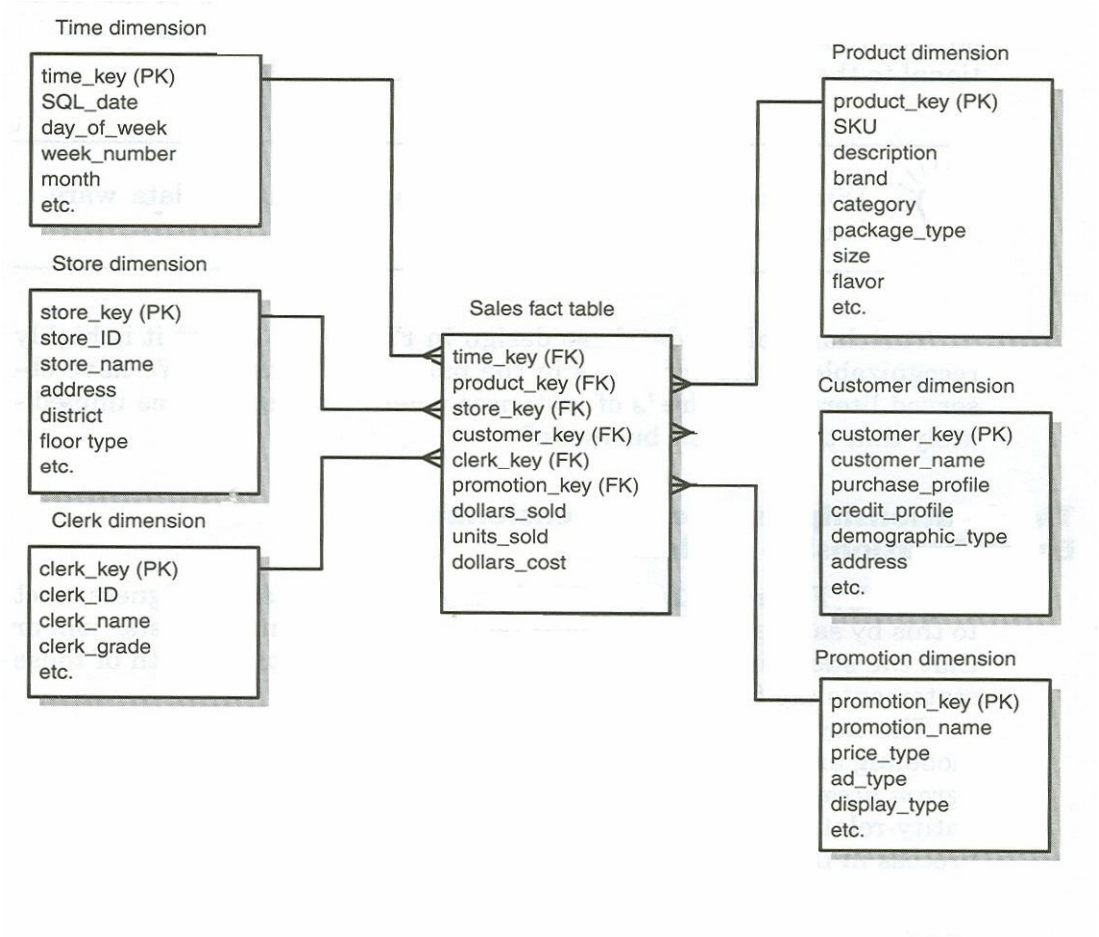


Figure 3.3. A Dimensional Model of an Enterprise
(Source: Kimball et al. 1998, p.145)

A dimensional model is composed of two basic concepts; facts and dimensions.

Fact

A fact is made up of measures and context data. This set of related data items are presented in a table with a multipart primary key consisted of two or more foreign keys that every data point corresponds to one and only one member from each of the multiple dimensions. The most useful fact tables also have one or more numerical facts. For example, the total revenue and net profits can be chosen as facts so that to analyze the sales of the company by store, by quarter.

Dimension

A dimension is a collection of properties, which is the most often descriptive textual information that organized within the dimension tables. Each dimension table has a single part primary key is associated with one of the foreign keys in the fact table. Dimensions allow us to view the fact in different contexts. For instance, the total sales can be viewed in terms of store, city, or region. These three levels are members of the geographical dimension of the sales process.

3.2.2.1.2.1. The Strengths of the Dimensional Modeling

First strength of dimensional modeling is that predictable structure of star join schema has a resistant to the unexpected changes, which occurs at the user side.

The second strength is that by the help of the predictable structure of the dimensional model it is possible to make hypothesis about the data, which is necessary for presentation and performance.

The third strength of a dimensional model it provides a chance to work with aggregates that enables to increase query performance without making an upgrade for the hardware system.

As a fourth strength of a dimensional model it gives an opportunity to add new data elements which are not planned during the design of the data warehouse to use in the future. Briefly, dimensional models open to the changes without reprogramming the query tool or reporting tool.

The final strength of the dimensional modeling is that there are standard approaches for organizing common modeling situations which include pay-in advance databases, event-handling databases, slowly changing dimensions and heterogeneous products. (Kimball et al. 1998, pp.147-153)

3.2.2.1.3. Modeling the Data Warehouse by Using Dimensional Modeling

3.2.2.1.3.1. Data Warehousing Objects

The following types of objects are commonly used in dimensional data warehouse schemas:

Fact tables are the large tables in the warehouse schema that store business measurements. Fact tables typically contain facts and foreign keys to the dimension tables. Fact tables represent data, usually numeric and additive, that can be analyzed and examined such as sales, cost, and profit.

Dimension tables, also known as lookup or reference tables, contain the relatively static data in the warehouse. Dimension tables store the information, which are normally used to contain queries. Dimension tables are usually textual and

descriptive and they can be used as the row headers of the result set such as customers or products.

Fact Tables

A fact table typically has two types of columns as seen in Figure 3.4 those that contain numeric facts (often called measurements), and those that are foreign keys to dimension tables. A fact table contains either detail-level facts or facts that have been aggregated. Fact tables that contain aggregated facts are often called summary tables. A fact table usually contains facts with the same level of aggregation. Though most facts are additive, they can also be semi-additive or non-additive.

Additive facts can be aggregated by simple arithmetical addition. A common example of this is sales. Non-additive facts cannot be added at all. An example of this is averages. Semi-additive facts can be aggregated along some of the dimensions and not along others. An example of this is inventory levels, where someone cannot tell what a level means simply by looking at it.

A fact table must be defined for each star schema. From a modeling standpoint, the primary key of the fact table is usually a composite key that is made up of all of its foreign keys.

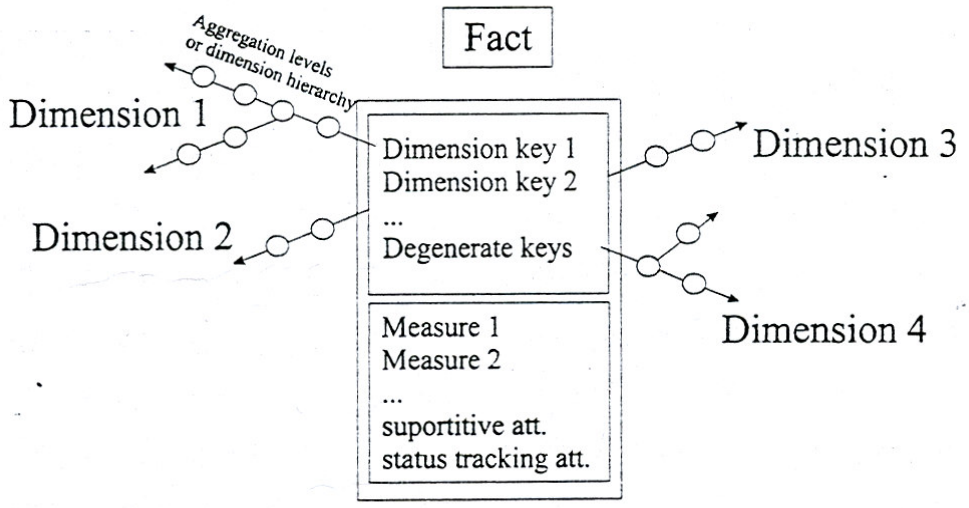


Figure 3.4. Notation Technique for Schematically Documenting Initial Dimensional Models (Source: Bayoğlu 2000, p.104)

Dimension Tables

A dimension is a structure that categorizes data in order to enable users to answer business questions. Commonly used dimensions are customers, products, and

time. For example, each sales channel of a clothing retailer might gather and store data regarding sales and reclamations of their Cloth assortment. The retail chain management can build a data warehouse to analyze the sales of its products across all stores over time and help answer questions such as: (Lane and Schupmann 2002, p.196)

- What is the effect of promoting one product on the sale of a related product that is not promoted?
- What are the sales of a product before and after a promotion?
- How does a promotion affect the various distribution channels?

(Lane and Schupmann 2002, p.196)

The data in the retailer's data warehouse system has two important components: dimensions and facts. The dimensions are products, customers, promotions, channels, and time. One approach for identifying the dimensions is to review reference tables, such as a product table that contains everything about a product, or a promotion table containing all information about promotions. The facts are sales (units sold) and profits. A data warehouse contains facts about the sales of each product at on a daily basis.

A dimension is a structure, often composed of one or more hierarchies, that categorizes data. Dimensional attributes help to describe the dimensional value. They are normally descriptive, textual values. Several distinct dimensions, combined with facts, enable the user to answer business questions. Commonly used dimensions are customers, products, and time. Dimension data is typically collected at the lowest level of detail and then aggregated into higher level totals that are more useful for analysis. These natural rollups or aggregations within a dimension table are called hierarchies.

Hierarchies

Hierarchies are logical structures that use ordered levels as a means of organizing data. Hierarchies are also essential components in enabling more complex rewrites. A hierarchy can be used to define data aggregation. For example, in a time dimension, a hierarchy might aggregate data from the month level to the quarter level to the year level. A hierarchy can also be used to define a navigational drill path and to establish a family structure.

Within a hierarchy, each level is logically connected to the levels above and below it. Data values at lower levels aggregate into the data values at higher levels. A dimension can be composed of more than one hierarchy. For example, in the product

dimension, there might be two hierarchies—one for product categories and one for product suppliers.

Dimension hierarchies also group levels from general to granular, with the root level as the highest or most general level. A level represents a position in a hierarchy. Query tools use hierarchies to enable users to drill down into data to view different levels of granularity. This is one of the key benefits of a data warehouse.

When designing hierarchies, the relationships must be considered in business structures.

Hierarchies impose a family structure on dimension values. For a particular level value, a value at the next higher level is its parent, and values at the next lower level are its children. These familiar relationships enable analysts to access data quickly.

A dimension is composed of several levels and their members. For example, the time dimension can be consist of day, month and year levels and also day level's members would be Monday, Tuesday, etc., or Day1, Day2, … Day7. The location dimension can be made up of city, region and country levels. Figure 3.5 illustrates a dimension hierarchy based on customers that composed of region, sub region, country name, and customer. These dimension members are used to decide a data items' position in the dimension with which a certain record in the fact table is matched.

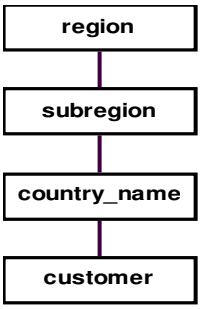


Figure 3.5. Typical Levels in a Dimension Hierarchy
(Source: Lane, P., Schupmann, V. 2002)

Multiple Hierarchies

A dimensions' member can be organized into one or more hierarchies that can also have multiple hierarchy levels. A member of a dimension may be located on more than one hierarchy structure. For instance, the time dimension is a good sample to consider the multiple hierarchies of a dimension. The Figure 3.6 shows defined two hierarchies for the levels of the time dimension. The reason of making two different

definitions is that if the data is to be aggregated on a weekly basis, the first hierarchy does not enable to this because the week member does not follow it. A week can span two months and so the second hierarchy is defined. If there is no need to analyze data on a weekly basis, second hierarchy is not necessary.

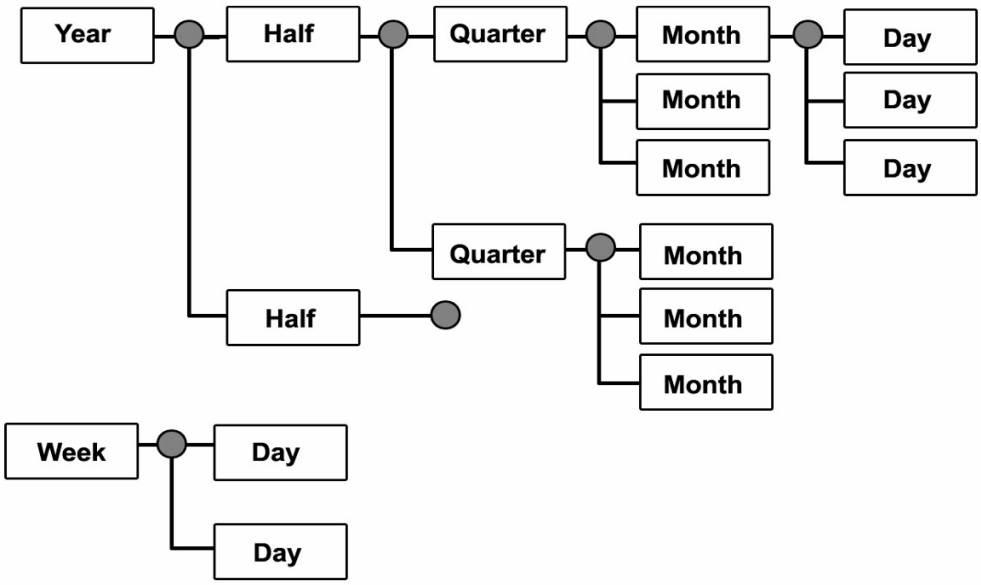


Figure 3.6. Two Hierarchies for the Levels of the Time Dimension
(Source: Lane, P., Schupmann, V. 2002)

Typical Example of Data Warehousing Objects and Their Relationships

A common example of a sales fact table and dimension tables customers, products, promotions, times, and channels are seen in Figure 3.7.

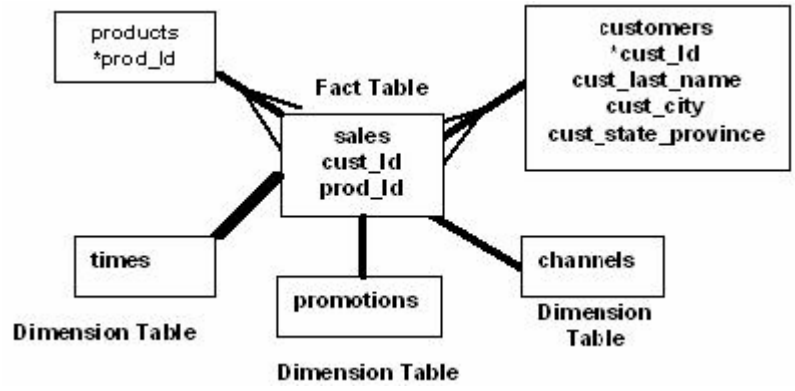


Figure 3.7. An Example of Data Warehousing Objects and Their Relationships
(Source: Microsoft Corporation, Microsoft SQL Server 7.0, 2000)

3.2.2.1.3.2. Data Warehousing Schemas

The schema is a database design containing the logic and showing relationships between all existing tables. The schema objects are organized in the schema models designed for data warehousing in a variety of ways. Most data warehouses use a dimensional model. The data warehouse schema is designed according to the model of source data and the requirements of users. There are two main types of database schemas: the Star Schema, the Snowflake Schema.

3.2.2.1.3.2.1. Star Schemas

Star schema is defined as a simple structure with relatively few tables and well-defined join paths (Poe et al.1998). This database design, in contrast to the normalized structure used for transaction processing databases, provides fast query response time and a simple schema.

Star schemas are physical database structures that store the factual data in the "center" surrounded by the reference (dimension) data. It can be very effective to treat fact data as primarily read-only data, and reference data as data that will change over a period of time (Murray and Anahory 1997). The Star schema uses denormalization to provide fast response times, allow database optimizers to work with simpler database design in order to yield better execution plans. A star schema is denormalized in the sense that dimension tables are not broken down into normalized tables thus providing familiar end-user views. Star schemas exploit the fact that the content of factual transactions is unlikely to change, regardless of how it is analyzed. (Ahmad and Azhar, 2002) A graphical representation of a star schema for sales is shown in Figure 3.8.

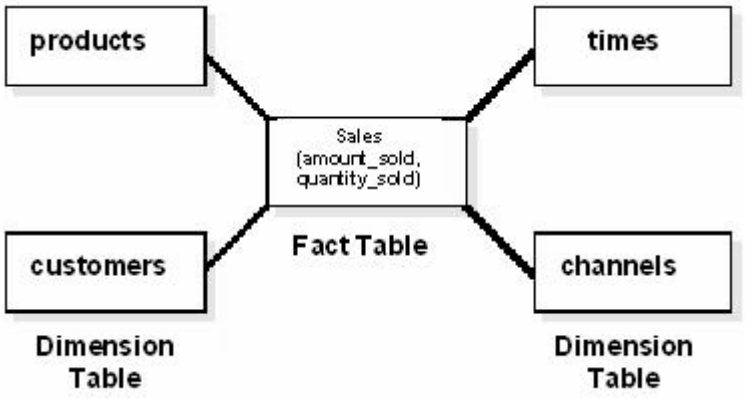


Figure 3.8. An Example of Star Schema for Sales
(Source: Lane and Schupmann, 2002, p.54)

3.2.2.1.3.2.2. Snowflake Schemas

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 products 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. Figure 3.9 presents a graphical representation of a snowflake schema.

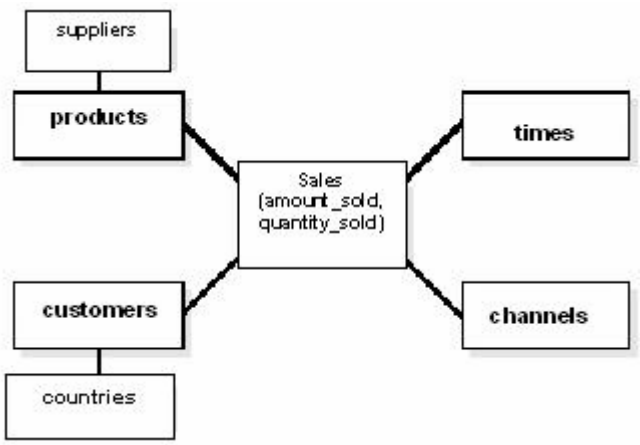


Figure 3.9. A Graphical Representation of a Snowflake Schema
(Source: Lane and Schupmann, 2002)

3.2.2.1.3.2.3. Fact Constellation Schema

Fact constellation makes use of multiple fact tables to separate the detail and the aggregated values. Each level in a focus dimension is associated to a fact table.

For each star schema it is possible to construct so-called fact constellation schema. Star schema structures create a proper subset of constellation schemas.

For example, let suppose the star schema is used to design sales information with store, product, and date. A single sales fact table with three dimension tables can be designed; store, product, and time. This schema is shown in Figure 3.10.

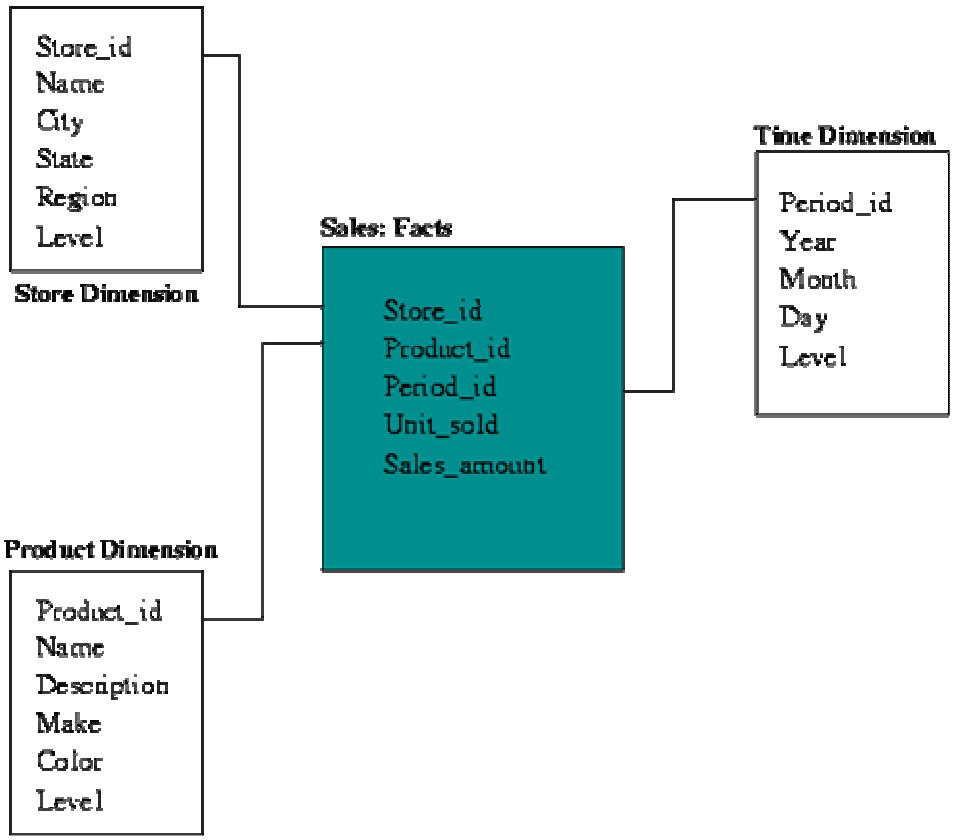


Figure 3.10. An Example of a Star Schema
(Source: http://www.olap.com)

In order to construct the sales star schema as the fact constellation schema, the store hierarchical dimension seen in Figure 3.10, would be separate three facts tables for store, state, and region. This is depicted by Figure 3.11.

The main disadvantage of the fact constellation schema is a more complicated design because many variants for particular kinds of aggregation must be considered and selected. Moreover, dimension tables are still large. Another disadvantage is that sometimes a single query must be broken into pieces before it can be answered. For example, a query that need performance comparison between the total sales at the state and region level needs two different queries to the state and region fact tables.

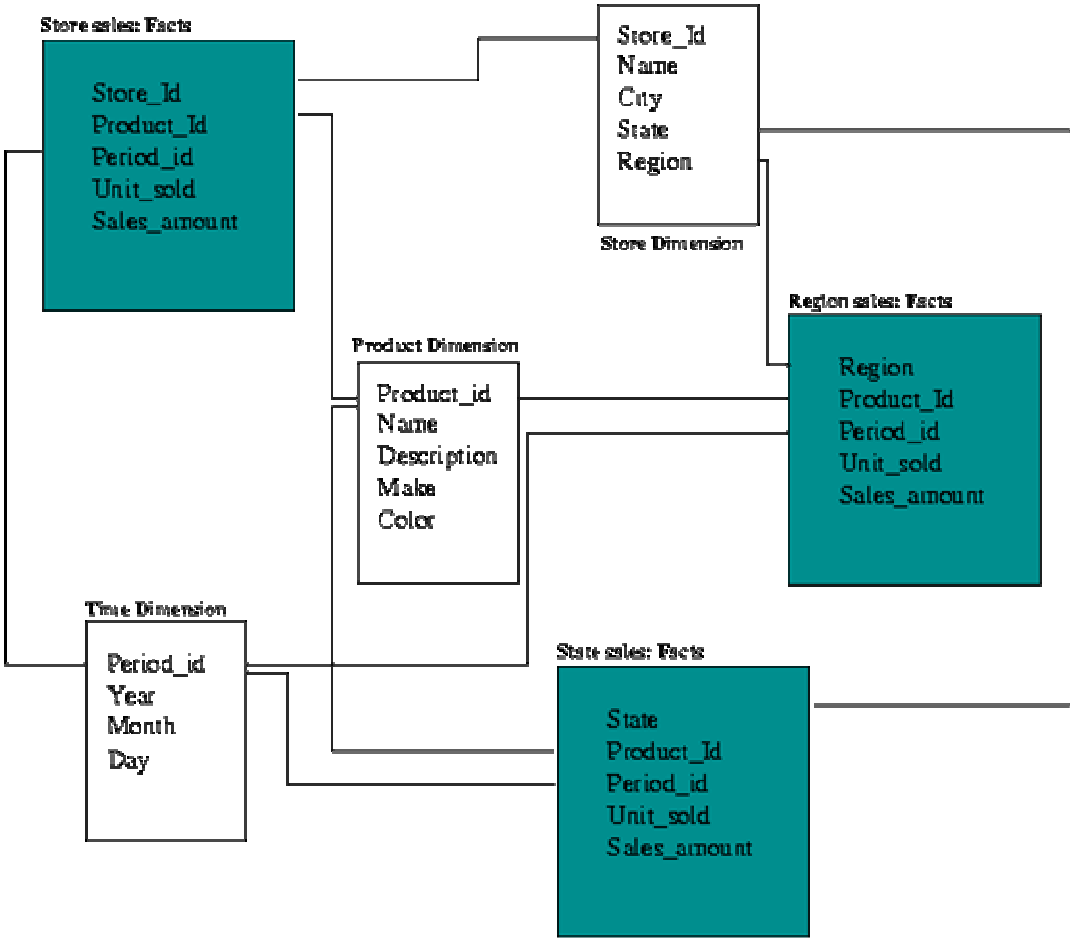


Figure 3.11. An Example of a Fact Constellation Schema
(Source: http://www.olap.com)

3.3. Data Loading and Transformation

Another part of the design of the data warehouse is the design of the data loading and transformation strategy. Loading the data into the data warehouse goes through several stages. These stages are: data capture, data transformation, and data application. (Bain et al. 2000, pp. 150-154)

3.3.1. Capturing the Data

The required data is collected from the operational systems and other external sources in different file formats and both relational and non-relational database management systems. The data capturing techniques include source data extraction;

DBMS log capture, triggered capture, application-assisted capture, timestamp-based capture, and file comparison capture.

Source Data Extraction: This technique can be used when a continuous history isn't required because source data extraction provides a static snapshot of source data at a specific point in time. It gives good result both when it is necessary to capture complete block of data and to change the current data in the warehouse with the new one, which is gathered from the same source, but at a different snapshot in time. A school might be a good example for the usage of this technique. Its data warehouse is updated yearly and the old data, which belongs to previous year, is transferred to the archive at the end of the year.

DBMS Log Capture: By the DBMS Log capture technique; data is collected from the DBMS logging system. Therefore DBMS logging has to be supported and turned on. Also the format of the log records has to be clear enough to understand easily and there should be convenient programming in order to reveal the content of the log files.

Triggered Capture: Triggers and stored procedure are prepared to be executed when the predefined events are occurred like insert, update and delete scripts. They are supported by many database management systems. The trigger realizes the events and activates the procedure. It is users' responsibility to check and maintain the procedure. The trigger opens to changes because it is under the control of the writer of the procedure not the database management system.

Application-assisted capture: The technique that includes writing programs to collect the data from the operational sources is called as application-assisted capture. This technique is completely under the control of programmer involving testing and maintenance responsibility. If it is not well enough tested and proved that this application will catch the target, it is better to use existing software instead of developing customized one.

It is possible to mention that DBMS log capture, triggered capture, and application-assisted capture methods provide an incremental record to work with continuous historical model.

Timestamp-based Capture: In this technique there is a timestamp value, which is used as a flag that indicates if the record has changed after the last capture or not. If there is any change or additional record, it is transferred to another file or table for next

step. There has to be some structural changes in the existing system in order to apply this technique.

File Comparison: A snapshot of the data source is taken at a specific point in time of data capture and is saved in a file and then it is compared with the previous snapshot file. If there is any change or additional record, it is transferred to another file or table for next step.

Timestamp-based capture and file comparison capture methods may not provide continuous historical model, since between two snapshot times there can be some changes that will not be captured.

All these techniques should be considered when planning data capture based on the sources that the data is captured from, which kind of data is required to capture from these sources, and the nature of capture history, continuous versus based on points in time. Because all techniques have some advantages for different steps that mentioned above and it can be better for the planner to use a combination of these techniques in order to get best result. (Bain et al. 2000, pp. 150-154)

3.3.2. Transforming the Data

There are several processes, which are used for transforming the data to the data warehouse. These processes should be designed to ensure three main functions; data validation, data scrubbing, and data transformation.

By data validation function, the data which will be transferred in to the warehouse becomes compliant and integrated with the existing format of warehouse such as adjusting the currency data collected from the source according to the currency field in the data warehouse. Validation also ensures referential integrity in terms of primary and foreign keys in the database.

The data collecting from different sources to feed the same destination table in the target data warehouse database can cause some discrepancies or conflicts, these should be removed by data scrubbing function. For instance, if a product has two different definitions at the source side, then while it is taking into the data warehouse system one of them should be chosen to avoid conflicts.

The data transformation process converts the captured source data into a format and structure suitable for loading into the data warehouse. The data transformation process prevents the anomalies in the source data to increase data quality.

Transformation of data can occur at the record level or at the attribute level. The data transformation process can perform in three different ways such as structural transformation for changing the structure of the source record according to the target database, content transformation for changing data values in the records, and functional transformation for creating new data values in the target records based on data in the source records.

3.3.3. Populating the Data Warehouse

After creating the data warehouse model, the following important step is the populating the data warehouse. The data availability and quality of data warehouse can be affected by the restrictions in the legacy system data. The data model that is the draft for the data warehouse should be examined in order to designate the populating process.

The data warehouse model determines what source data will be needed, the format of the data, and the time interval of data capture activity. If the data required is not available in the operational system, it will have to be created. For example, sources of existing data may have to be processed to create a required new data element. The Sale Fact may need profit per each product, which does not take place in the data source. So it has to be calculated by subtracting the product cost from the sales price.

Sometimes the data may need to be processed before capturing into the warehouse or may need to be removed from the model because of capturing cost or its unavailable situation.

3.4. On-Line Analytical Processing (OLAP)

Popularity of operational databases such as handling order processing or share dealing etc. has grown incredibly after the mid 1980s. Corporations adopted these databases and used them to gain efficiency in their business and transactions. A great portion of these databases are relational. Numerous database vendors supported relational databases and presented powerful tools that provided clients, easy, efficient and practical database management. Inevitably, many companies adopted their data to relational databases that are needed in many fields such as operation and control activities known as transactions. Relational databases are running the daily processes of banks such as controlling the daily operations of customers transferring, withdrawing,

or depositing funds in their accounts. Referential integrity, good fault recovery, support for a large number of small transactions, etc. are the key factors to the widespread usage of relational databases.

The popularity of data warehouses increased as companies noticed how important it is to join the data they collected by their operational systems, for future projections and decision making processes. Supposing that they used the transactions, they needed to design queries that summarized data and fed management reports. At the other hand, these queries would be significantly slow, because they generally rendered large amounts of data, while using the database engine with everyday transactions, which in turn dramatically affected the overall performance of operational systems. If only the data used for reporting and decision making were separated there would be a solution. Hence, data warehouses were designed to keep this sort of data so that it can be used in further strategic works in enterprise.

Databases of the most important relational database vendors like Microsoft, Oracle, Sybase, and IBM are presented as tools for building data warehouses. Following the big pioneers, lots of smaller database companies also integrated warehousing within their products as database warehousing has gained a more accepted reputation as a powerful part of a database, rather than an addition. Informational reports are produced from the data stored in these warehouses and they cope up with the questions such as "who" or "what" about the original data. As an example of this, a data warehouse can be used to find which department of the company yielded the maximum income for a specific period of the year or what the overall cost was for the same interval.

Data warehouses are based on relational technology. OLAP analysts gain the advantage of through insight into data in a fast, consistent and interactive way with a wide variety of possible views. By the help of the OLAP's ability that transforms the raw data to useful information, relevant personnel can plot the real factors affecting or enhancing the processes and transactions throughout the enterprise.

OLAP systems enable "what if?" analysis and this is a significant advantage. What if analyses help the analyst, to see the results of each different scenario. By the help of this brilliant feature, the best course of action for the firm's business can be determined and this is why it is such a unique decision making tool. OLAP and data warehouses are complementary. While data warehouses are responsible from keeping and storing the data, OLAP transforms this data into meaningful information. OLAP

techniques may vary from the simple navigation and browsing of the data (referred as 'slicing and dicing'), to more complex studies, such as time-series and modeling.

Data warehouses collect, reorganize, store and manage the raw data according to a special schema. OLAP gives a final and useful shape to this raw data. As an addition, advanced OLAP analysis and some other tools, like data mining may further transform the information into more meaningful knowledge that provides the analysts to make forecasts of the future performance of an entity based on past data.

The term OLAP was first mentioned by E.F. CODD, N. PENDSE and R. CREETH simplified the definition of it in their further analysis as, the applications that deliver fast analysis of shared multidimensional information (FASMI). FASMI can be studied categorically from the scope of its characteristics: WEB_11 (2004)

> **Fast:** The client running these applications is an interactive one who expects to have the required data on time and periodically on constant basis. The client needs the queries in five or less seconds, and most of the queries will show an adhoc characteristic contradictory rigidly predefined reports. For example, such a client should have the flexibility of generating a report that is formed by combining several different attributes from the data stored in data warehouse.
> **Analysis:** Basic numerical and statistical analysis of the data should be able to be performed by OLAP applications. Application developer can pre-define the calculations or the client may define such ad-hoc queries. This is the core of OLAP that makes it so powerful, allowing the addition of hundreds, thousands, or even more records to come up with the hidden information within the raw data.
> **Shared:** Numerous people should be able to share the data delivered by OLAP applications.
> **Multidimensional:** Multidimensional database schemas are an essential characteristic of OLAP and OLAP applications are based on data warehouses or data marts which are built on those multidimensional databases schemas.
> Information: All data and information related to the application must be open to OLAP applications and the access of OLAP applications should be granted. As an example, an OLAP application may require to access historical transactions in order to calculate and process the correct information for an annual interest case. The point is not only the need to the data that is located but also its volume is liable to be large.

3.5. About Applications of OLAP

Various divisions of the enterprises such as sales, marketing, finance, even manufacturing may use OLAP tools and their applications to run their transactions.

Some activities of the finance and accounting in an enterprise, like budgeting, financial performance analysis and financial modeling may use OLAP tools. An analysis performed over the OLAP outcomes can be used for accurate budgeting of the specific period to reflect the expenses of the organization and indicate possible alerts to finance department (as an example) to avoid some budgeting deficits. These OLAP reports are also valuable to identify the weak points of the organization processes that should be reengineered or eliminated (or advanced as well) as well as to continue and/or develop some powerful processes of the enterprise.

Sales department may benefit from OLAP applications to perform predictions and forecasts for projections and sales analysis. These tools bring a more healthy insight to sales department in order to determine the best sales techniques. After analyzing the reports of OLAP, products which are found to be more or less profitable may be decided to be produced in smaller or bigger batches even some of them may be forwarded to be dropped from the production line.

Like sales department, marketing can also use OLAP tools in its own activities such as market research analysis, sales forecasting, promotions analysis, customer analysis, and market/customer segmentation. OLAP applications reveals the markets that yield good/bad incomes. As an addition to its advantages mentioned, OLAP analyses enable the analyst to perform segmenting of sales according to the age, gender, income, geography etc properties of customers. Analyses can also be used to determine the possible markets for a given product that will be presented versus some other brands of different firms. For example, it is proper to present some products of a specific segment in a market where the local people of the area also belong to this very segment.

The most important advantage of OLAP is these analyses which enable the analyst to prepare more prosperous forecasts and give better decisions.

Common manufacturing OLAP applications involve production planning and defect analysis. These applications will provide basis to determine the effectiveness of quality control and quality assurance (QC/QA), as well as to point the best way to produce a certain product, and the most efficient vendors providing the raw materials. The reports of OLAP systems may emphasize or even find some problems which are not yet encountered and well-hidden behind the numbers that may be misleadingly indicating good performance. As a result, the problems are identified and solutions for better efficiency of the related processes are easily taken.

OLAP delivers the information to all of the OLAP users mentioned above, to be used in making effective decisions in the organization's line of business and future directions. Information brought by OLAP is rendered so fast and on time it is needed. Fast delivery property of the information is a major key asset for successful OLAP applications. Since the value of accurate information is highly dependent upon the time it is needed and collected, time is the most critical piece to make effective decisions.

Outcomes of OLAP applications can be concluded to present complex relationships and are often calculated on the fly. However, analyzing and preparing models for complex relationships is not practical if response times are not consistently short. In addition, since the data relationships may not be known in advance, models the data must be flexible, so that it can be modified whenever new findings occur. A flexible data model is a guarantee of the OLAP systems to respond the ever-changing needs of the enterprise for effective decision making. WEB_11 (2004)

3.6. What are the Features of OLAP

OLAP concept is applicable to a wide range of functional fields of an enterprise. However, all OLAP applications must handle some common key features as indicated below: WEB_12 (2004)

Multidimensional views of data (data cubes)

Calculation-intensive capabilities

Time intelligence

3.6.1. Multidimensional Views

Strictly related to the daily requirements of the daily business life, business models are built multidimensional. As an example, a business model created by a Human Resources analyst should include name, address, telephone, gender etc information related to employees working for the firm. Despite of being so simple, that model even includes several dimensions and it must surely be underlined that more sophisticated analyses like financial models built to answer the various needs of the firms include lots of different dimensions as a result. An office depot can be taken as a daily life case: Several dimensions like time, location, product, people etc can be identified for the business of that company. Since sales may vary from quarter to quarter

or from year to year time dimension must include some more sub-hierarchies. The location, or geography dimension, may also be divided to multiple levels such as city, state, country, and so on. The product dimension is the same in these two dimensions. It can have several levels, such as titles (computers, printers, etc.), and more aggregated levels (printer cartridge, printer paper, etc.).

This property of OLAP tools provides to “slice and dice” the data, as well as providing flexible access to information embedded in the database. The data should be available to be able to be analyzed across any dimension, at any level of aggregation, with equal functionality and ease by the help of OLAP applications. For example, costs at a given period (a particular month), for a certain product subcategory (or brand name) in a given region can be obtained accurately using such applications. OLAP software should be capable to view these data in a natural and responsive fashion, insulating the clients from complex query syntax. Managers should not feel the necessity of writing structured query language code (SQL), understand complex table designs or difficult table joins.

Data cubes are multidimensional data views. Not likely a too simplistic point of view such as a cube with three dimensions, in actual life data cubes can have as many dimensions as the business model permits.

3.6.2. Calculation-Intensive

Most OLAP applications can perform simple data aggregation along a hierarchy like a cube or a dimension, some of these tools are able to calculate more complex operations, like taking the percentage of totals, and allocations that use the hierarchies vertically. It is an important asset of an OLAP application to be designed capable of making such complex but practical calculations because these calculations add great benefits to the final solution.

Another important OLAP feature is the ability to carry out trend analysis. Since trend analysis include algebraic equations and complex algorithms, such as moving averages and percent growth these analysis are complex in nature.

While OLAP systems are used to create information from the collected data that may lead to new knowledge, OLTP (On- Line Transaction Processing) systems are used to collect and manage data. OLAP applications conduct (basic to complex) calculations

on the raw data and then transfer them into meaningful information and then later to knowledge.

3.6.3. Time Intelligence

The universal dimension for all of the OLAP applications is “time”. It is almost impossible to find a business model OLTP (On- Line Transaction Processing) systems are used to collect and manage data. Business processes are compared and judged from this point of view. For example, the monthly income statements of two different regions (south-north) may be compared and performances can be measured. Or the cost figures of a firm in a given month can be compared to its cost figures following (chosen) month.

At the other hand, time dimension is not always treated the same way to other dimensions. It is an overall comparison criterion and is different from the others from this point of view. As an example, a manager may want to see the sales totals of the first two weeks of a given month but is not likely to ask about the sales of the first two products in the product spectrum. OLAP systems are to be built to easily permit the concepts like “year to date” and “period over period comparisons” to be defined.

“Balance” is another important issue to be analysed deeply when time is considered in OLAP applications. Balance of computers built in a quarter must be understood as the sum of computers built in the individual months of the given quarter. For example, if the firm produced 30 computers in January, 10 in February, and 25 in March, the balance of the computers produced for the first quarter will be 30 + 10 + 25 = 65 computers. At the other hand, the balance of cash flow for a quarter must be examined regardless from the cash flow balance of the individual months in the quarter and it is usually meant to be the ending balance at the end of the quarter. For example, if the balance at the end of January is $30.000, February is $15.000, and March is $25.000, the balance at the end of the period is $25.000. Finally, the balance of the employees in a given period is the average balance of the number of employees in the firm in the individual month. If the number of staff working in January is 10, February is 5, and March is 15, then the balance of employees for the first quarter is (10+5+15)/3=10.

3.7. OLAP Storage Architecture

There are 3 different alternatives of storages based on the data cube storage method. These alternatives are given below: WEB_9 (2004)

1- Multidimensional OLAP (MOLAP),

2- Relational OLAP (ROLAP),

3- HOLAP which is the hybrid form of OLAP and MOLAP.

Each of the above 3 alternatives have various benefits, depending on the size of the database in use while causing some drawbacks relatively.

3.7.1. Multidimensional OLAP (MOLAP)

MOLAP is a multidimensional data storage format that provides a high performance. The data feeding the cubes is kept with MOLAP on the OLAP server as a multidimensional database. MOLAP is a specifically optimized solution for multidimensional data queries and due to this cause-and-effect it gives the best overall query performance.

MOLAP is so convenient for small to medium-sized data sets. MOLAP needs copying all data and converts its format conveniently to fit the multidimensional data store. Copying process of all of the data for such datasets would not need a significant loading time or utilize large amount of free disc space.

3.7.2. Relational OLAP (ROLAP)

Data feeding the cubes in the original relational tables is kept in relational OLAP storage. Aggregation data in this OLTP system is stored and referenced by a separate set of relational tables. These tables declared are not downloaded to the DSS server. Tables holding the aggregations of the data are named as materialized views. When the cube is created dimensions are set and these tables store the data aggregations as defined by these dimensions.

ROLAP concept orders the presence of dimension and measure fields of the aggregation tables. All dimension columns are indexed A unique composite index is also created for all of the dimension fields. ROLAP is convenient for large databases that are rarely used. As an addition, generating reports from this system or processing

the cube data may affect users of the operational database reducing the performance of their overall transaction processing and this is the only drawback of the system.

3.7.3. Hybrid OLAP (HOLAP)

HOLAP can be accepted as the combination of MOLAP and ROLAP systems. So similar to ROLAP architecture, data is kept in its relational database tables in HOLAP too. Multidimensional format is chosen to store the aggregations of data performed. While taking the advantage of the faster performance of the multidimensional aggregation storage, HOLAP provides connectivity to large datasets in relational tables and this is the most significant advantage of this concept. However, the efficiency of HOLAP is adversely affected from the amount of processing between ROLAP and MOLAP systems.

CHAPTER 4

IMPLEMENTATION OF THE DATA WAREHOUSE

Today's competitive world conditions dictate 'performance measurement' as the most important issue to the enterprises. Neither the management challenges nor the technological changes are the most significant parameters that affect the success of a company due to the recently advanced techniques that are used to implement new solutions to the new problems. The new era brings about concepts like continuous improvement and process control to let the employees work with high performance and motivation.

Companies adopt new systems to scan and develop the efficiency of the employees according to some pre-defined criteria. In this critical scenario, the key factor belongs to 'Management Information Systems' that provide coordination and accuracy and build up infrastructure.

Therefore, enterprises feel the need of a complete management information system that is integrated into their different branches such as manufacturing, accounting, human resources, etc. more than ever to be able to get information aggregated from data collected from all the departments.

In this chapter, a data mart with a real data source is applied in a manufacturing and distribution company having a large-scale distribution network throughout the country. The system, implemented in the company, is a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision-making and control processes in an organization. It is called "information system".

In this part regarding the business objectives and end user requirements, project scope is determined, and then the data mart is created and populated with the data that are cleaned and integrated before. Following these steps, Online Analytical Process is applied to create data cubes for managing data mart.

4.1. Definition of the Project

The goal of this project is to create a data warehouse for 'Performance Analysis and Management System' by analyzing the current operational system according to the company's requirements and sales database. This system will be used by Sales and Human Resources Department to evaluate the performance of the field organization employees. 'Performance Analysis and Management System' will also be constructed in order to trace the application of company standards.

4.2. Scope of the Project

In this project, operational data source, where data is collected about the efforts of Sales Representatives' to implement these key performance indicators, will be examined, and a departmental data mart including user feedback and existing operational data source will be created. After the field organization related data mart is created according to the standards, the data will be presented to the managers who will use these to make decisions in performance management system.

4.3. Profile of the Company

The company constituting the subject of this study is a manufacturing and distribution company that has a large-scale distribution network all over the country.

Sales operation is done by the 'Sales Representatives' at the lowest level presented in Figure 4.1. Sales representatives are the employees who carry out the sales operation, build a face-to-face relationship with the customers, and have the key role in the 'Sales Operations' level.

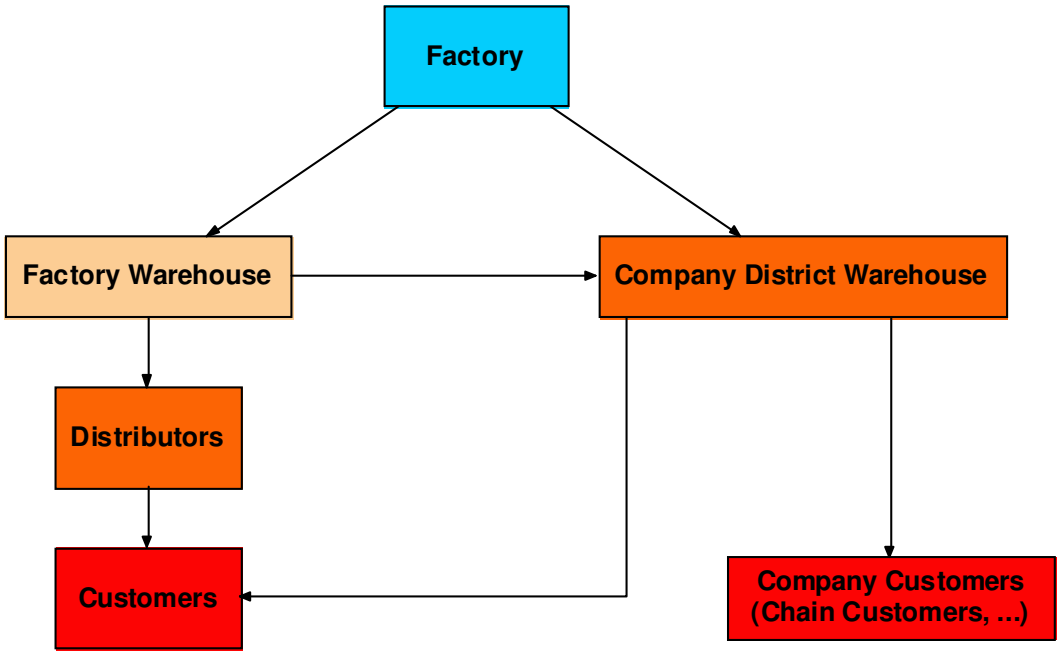


Figure 4.1. Product Distribution Diagram

4.3.1. Sales Representative and Sales Manager Workflow

"Route: Sales Representatives' Order of Visit." On their routes, Sales Representatives (SRs) visit their customers to sell products, to maintain additional promotions and to ensure the maintenance of company assets that are kept in the customer premises, etc.

Sales Representatives pay visits to the customers the whole week. In the morning, Sales Representatives load products from the warehouses onto the vehicles and start their routes. The order of the customers to be visited is strictly determined in the central office by using some route optimization tools. When they arrive at the customers' premises, they start to control the availability of the products, conditions of product freshness, asset conditions, and they sell products that the customers' order or request.

At this stage, the company needs to assess the performance of the Sales Representatives using some pre-defined criteria. The mission of the Sales Representatives is not only to sell products but also to share the company's vision and implement the standards that have been accepted as the basis of the sales operations. It is the duty of the Sales Manager to check whether the Sales Representatives regard

these standards during the sales operation or not. This helps the company measure the performance, and lets it make continuous improvement in the sales operations.

In our project, Sales Managers have the leading role. As shown in organization chart (Appendix A), Sales Managers lie on the top of the Sales Representatives.

From time to time, Sales Managers, along with the Sales Representatives, pay visits to customers. Before starting the visit, at the warehouse, the Sales Manager fills in a questionnaire in which some critical issues regarding the Sales Representatives are evaluated. Afterwards, the products are loaded onto a vehicle and they start their visit together. During these visits, by filling in the questionnaires, Sales managers collect data about issues such as sales, time spent at the customers', the availability of rival firms' products at the customer, the condition of the product stands of the firm and promotion of the firm. They check whether the firm complies with pre-defined field organization standards or not. Besides, the Sales Managers compare the data they collect with the questionnaire the Sales Representatives fill in beforehand, and they make an evaluation. If necessary, they give a warning to Sales Representatives, and they ask them to complete the deficiencies until a certain time.

During the visit the collected data are transferred to the operational source via pocket PCs by the employees in charge at the warehouse that the Sales Representative is responsible for. These data should be transferred to the central databases within seven days at the latest. These data are used for the reports that are taken from the operational source.

4.3.2. Data Flow for Field Organization

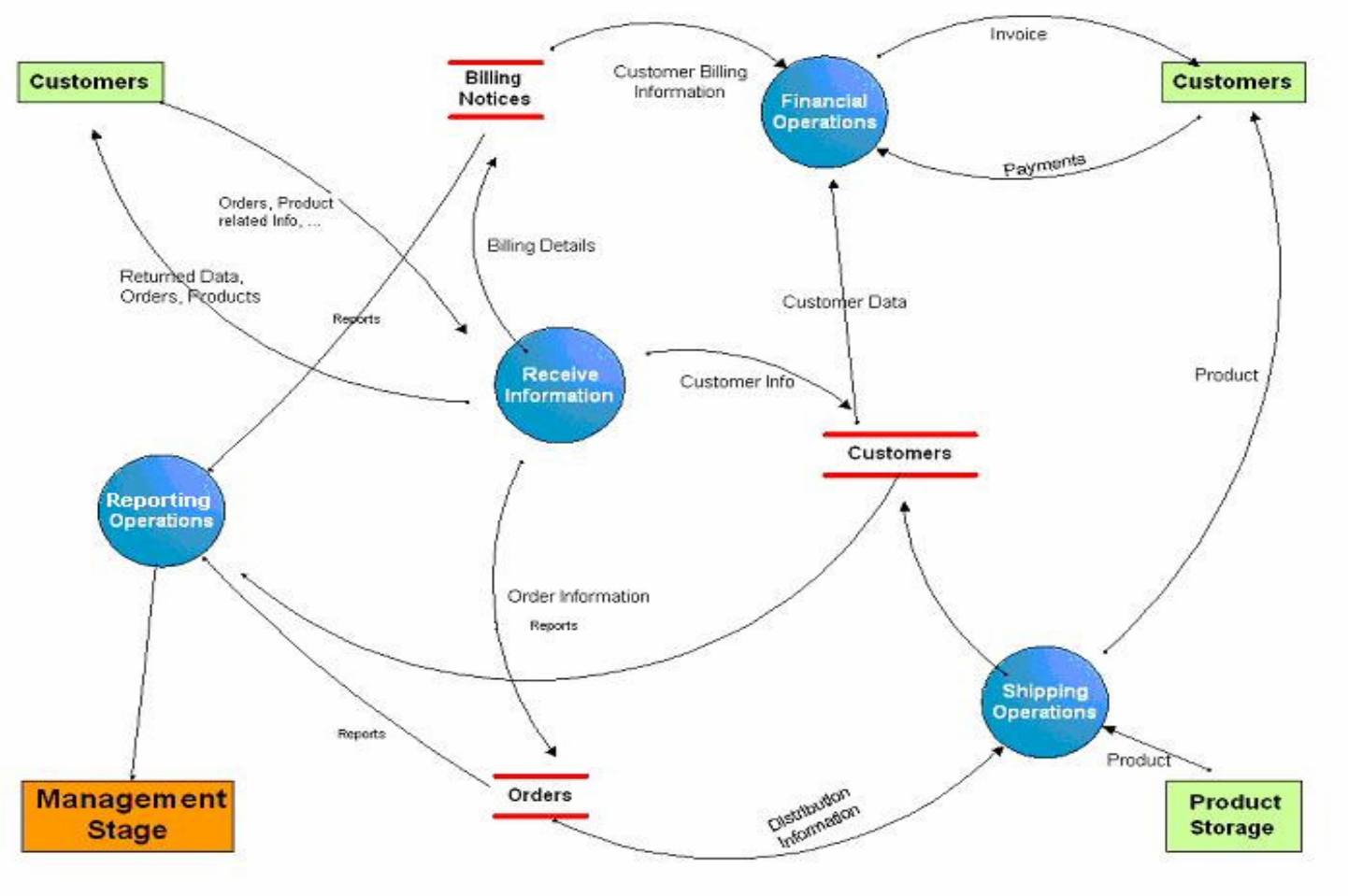


Figure 4.2. Data Flow Diagram of Sales Operation

The Sales Representatives take orders from the customers, bring invoices prior to sales, enter data about the customer into the system, deliver products, give promotions and inform the customers about the new campaigns of the company as seen in Figure 4.2.

Customer billing details are forwarded to finance department in the head office, and financial operations such as invoice and payments are completed. Moreover, information regarding sales, customer visits and orders of the customers are sent to the head office.

The orders are delivered to the warehouse information system to let the warehouse management system handle production stock and shipping operations.

The head office sends the aggregated order information to the warehouse management system.

Finally, after all these operations, managers have the chance to take reports from the information system about several subjects such as stock turning ratios, sales, employee performance, etc.

4.3.3. System Architecture of the Company

Operational System: The existing operational system is Oracle8i Database Server (8.1.7) that works on Microsoft Windows 2000 Server Network architecture. In the company, the daily operations are completed by using the intranet system, MS Internet Explorer Interface and Compaq iPAQ pocket PC.

Compaq iPAQ pocket PCs has MS Windows CE as their operating system, and Embedded Visual Tools family is used during the development stage. The existing system structure of the company is shown in Figure 4.3.

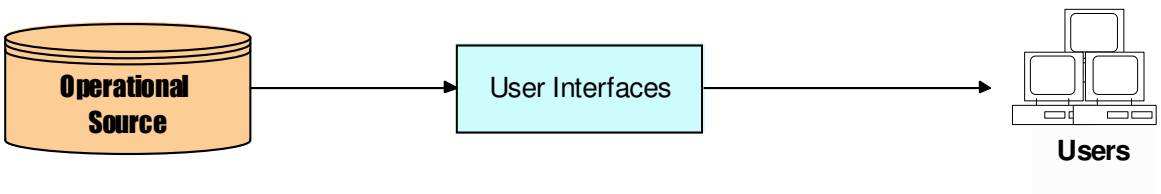


Figure 4.3. Existing System Structure of Company

4.4. Subject Project Architecture

In this study, Oracle 8i is used as the operational source. As the data is captured in the data staging area, PL/SQL is used to prepare the data that will be transferred to the data warehouse by using Microsoft Data Transformation Server. MS SQL Server 2000 was selected as the data warehouse development tool.

Ms SQL Server comes with the OLAP tool 'Analysis Services'. Thus, owing to the integration of and communication between the OLAP and database environment, it was selected as the OLAP development tool.

Microsoft SQL Server 2000 also includes an integrated set of tools for data transformation between the origin and the destination data warehouse.

Workflow of the subject project architecture is illustrated in Figure 4.4.

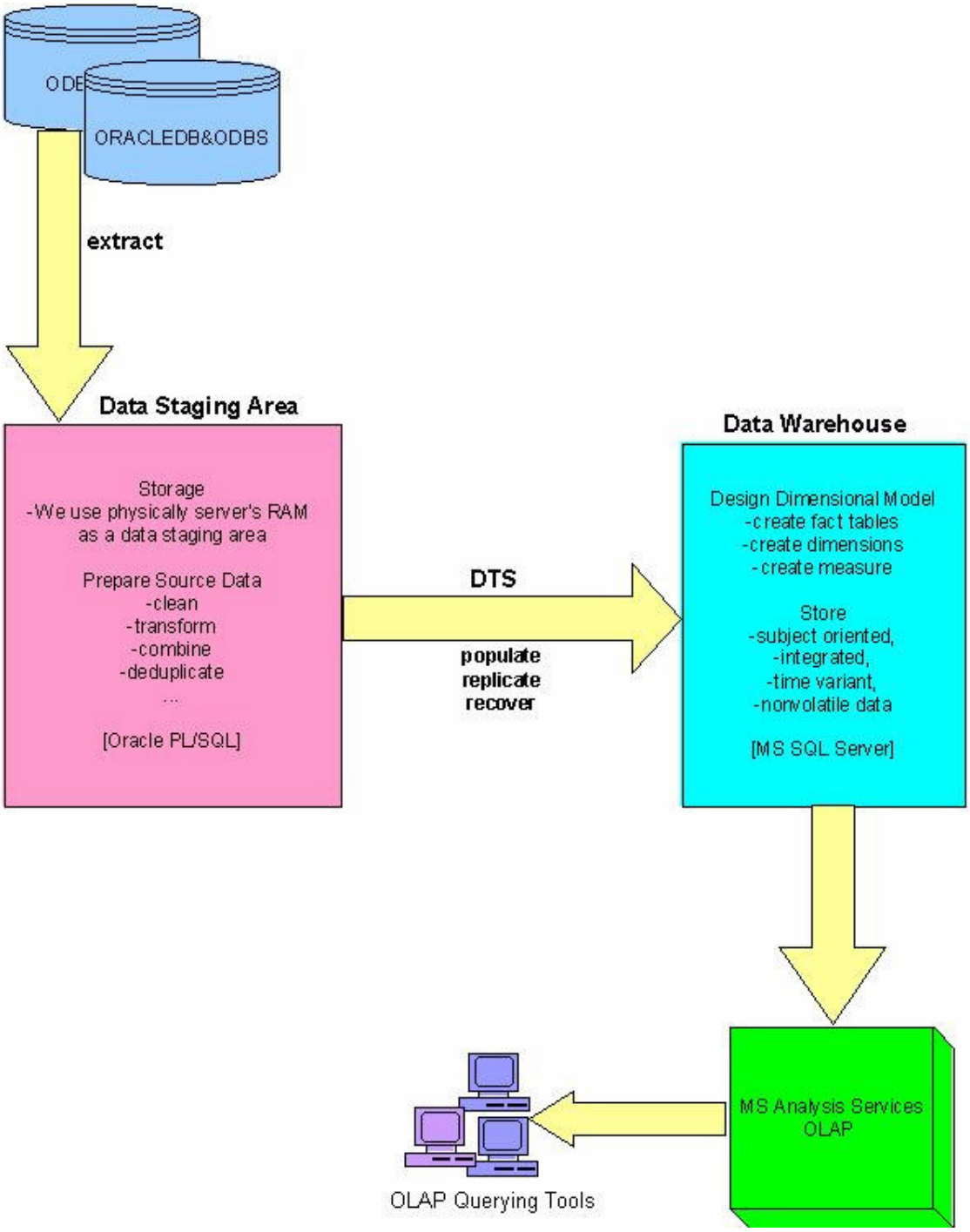


Figure 4.4. Subject Project System Flow

4.5. Development Method of the Data Warehouse

Spiral development method that is shown in Figure 4.5 has been developed for several years, and it is based on the experience with various refinements of the waterfall model as applied software projects.

The starting point is to perform a specific business data analysis process to identify the data warehouse implementation project by a group of end users. All steps will be followed to acquire solution integration.

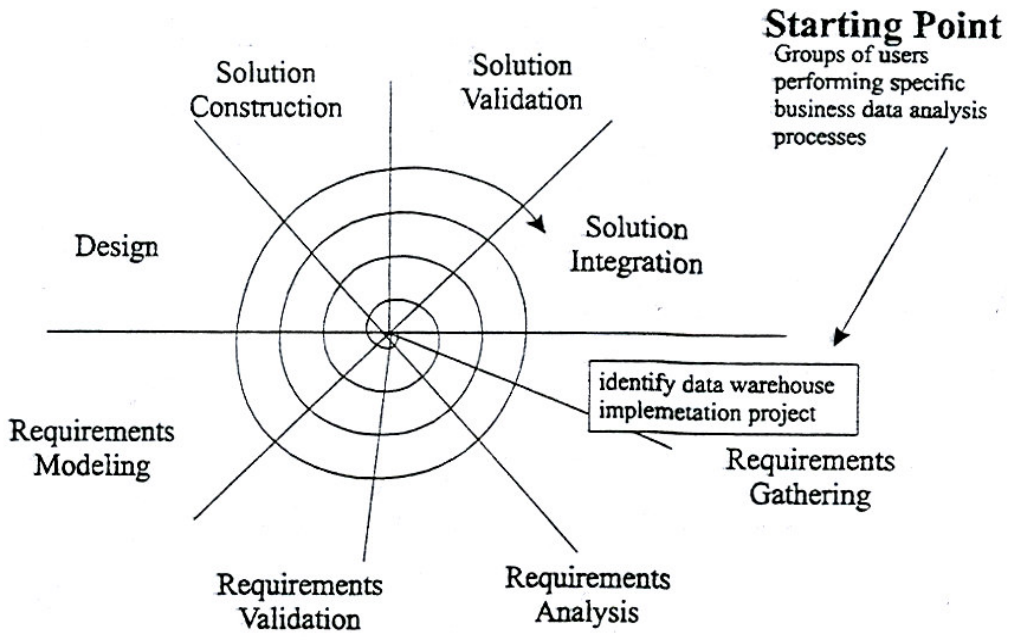


Figure 4.5. Spiral Development Method of the Data Warehouse
(Source: Bayoğlu 2000, p.95)

4.5.1. Gathering the Requirements

End user requirements are collected and documented during the process of gathering the requirements. Gathering of the requirements is often taken as a kind of activity that includes end users in the business process and information analysis environments. On the other hand, it is an approach aiming to determine the core of the problem domain, in which the modeling will be implemented as a next step. End user needs are mostly documented informally at this stage, or they are not plotted on the schemas in detail.

Traditional techniques such as interviews with end users, study of existing documents and reports, and monitoring the ongoing information analysis activities are the methods for gathering the requirements. As an addition, experience with information analysis and business process reengineering are important assets that contribute significantly at this stage. The outputs of the process of gathering the requirements are utilized as the basis for the further dimensional model producing stage.

Client side needs of a data warehouse-modeling project can be listed in two categories. Process-oriented requirements represent the major information processing elements that end users need in common, or desire to perform while the data warehouse is being prepared. Second item is the information-oriented requirements that represent the major information categories and data items that are required by end users for their data analysis activities.

Generally, all requirements can be put in any or both of those two headlines. The kind of needs and the degree of precision with which the needs will be mentioned often depend on two parameters: the kind of information analysis problem being thought for the data warehouse implementation project and the degree that the end users are able to define their requirements, scenarios and the strategies they use in their information analysis activities.

4.5.1.1. Process-Oriented Requirements

Several types of process-oriented requirements may be available.

Business Objectives

The highest-level expressions of information analysis objectives expressed in business terms are ‘business objectives’. Numerous business objectives can be set for a specific data warehouse implementation project.

In this project, the business objectives could be stated according to the interview held with the user. These objectives;

- inform management stage about:
    - The sales operations
    - Key Performance Indicators (KPI)
    - Customer Relationships
- help to make decision about the future actions on customer relations

- help to make assessment about how successfully the company's mission and vision are implemented.

Business Queries

Various queries, hypothesis and analytical questions the end users issue and try to resolve in the course of the information analysis activities are modeled by business queries. Business queries, which could not be formulated precisely and could not be coded in the terms of SQL, are expressed in business terms.

Defining the Symptoms

The idea of carrying out the study comes with the occurrence of some problems. In the very first steps of this study, it is important to separate the symptoms from the problems. The symptoms are the obvious side of the problems. Therefore, this step will help us define the basic problems accurately.

The symptoms found in this project are as the following:

- Decrease in endorsement
- Decrease in customer satisfaction
- The increasing number of customer complaints

Describing the Problems

The main objective is to overcome the problems that led to the preparation of this project. Understanding the problems and the reasons of the symptoms is the most important step of the following flow:

- Defining the Symptoms
- Describing the Problems
- Defining the Performance Criteria Set

We will find out specific solutions for these problems. Thus, the solution and the real problem correspondence will affect the success of the project. Otherwise, it will cause high time consumption and high cost.

The problems are:

- to maintain the requirements of the standards and policies defined in the organization.
- the lack of assessment of the field organization employees.
- the lack of training of the field organization employees.

Defining the Performance Criteria Set

There are four kinds of companies. Some companies do not collect data about the internal operations. They think they do not need any reviewing sessions for internal operations. Some other companies collect data, but they never investigate these data. The third kind of companies collects data, do some analysis, yet they do nothing more than these. These companies do not know why they collect data. The successful companies are in the fourth group. They make analysis and they make decisions. After these sessions, they take actions. These actions are based on some performance key indicators in this study.

Key Performance Indicators

- Product-based standards
- Customer-based standards
- Salesperson-based standards

Key Performance Indicators Based on the Product

- Freshness
- Product availability (reachable)
- Variability

Key Performance Indicators Based on the Customer

- Payment type
- Due day (for credit sales)
- Promotions (free products)
- Price labeling
- Product location (shelves status)
- Competitors' Products

Key Performance Indicators Based on the Salesperson

- Truck condition
- Availability of tools (calculator, ...)
- Endorsement
- Visit duration
- Visit number
- Precautions

When managers start to assess the salesperson based on the criteria set, they will give points to each criteria element. These points will show their success ratio.

Managers collect data about the sales representatives' efforts at implementing these standards. After collecting data, it is very important to analyze the data and get some information about what is going on in the field.

Salespersons do their routine daily operations. However, managers are responsible for assessing the salespersons according to the performance criteria. For these criteria set, managers collect data from customers. When analysis session ends, it is time to make a decision on the performance of the employee, and, if necessary, take action to improve it.

4.5.1.2. Information-Oriented Requirements

Information-oriented requirements represent an initial perception of the kind of information end users use in their information analysis activities.

Information subject areas

Information subject areas are high-level categories of business information. Information subject areas are usually used to build the high-level enterprise data model. When available, information subject areas indicate the scope of the data warehouse project.

In this project, information subject areas of interest are: customer, sales manager, and sales representative. They are depicted in Figure 4.6.

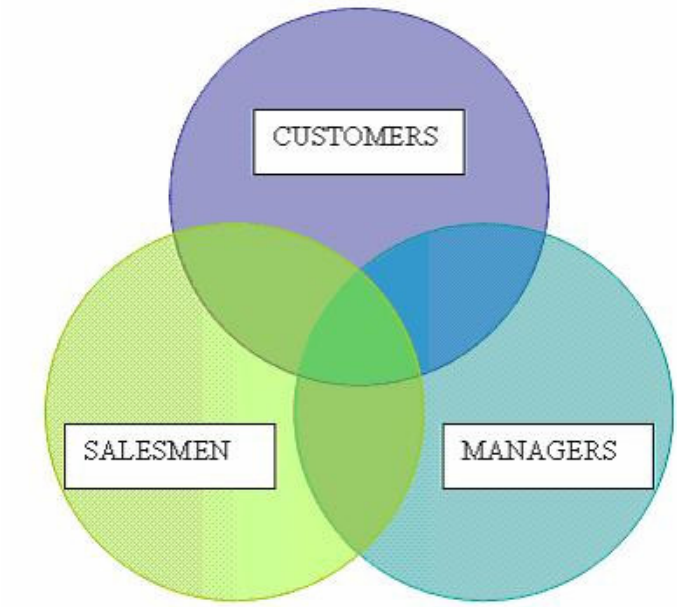


Figure 4.6. Entities in the Project

4.5.2. Analyzing the Requirements

Requirement analysis phase includes the investigation of the informal requirements of the end users and the preparation of the dimensional models that show the facts, measures, dimension case and dimension hierarchies. Dimensional hierarchies may consist of parallel hierarchical paths. The output models of the requirements analysis phase must be simple in nature, since they are not yet discussed and confirmed by the end users.

Requirements analysis techniques are utilized to prepare the leading dimensional model reflecting the end-user requirements that are noted previously in an informal way. Requirement analysis gives a schematic representation of the model that the information analyst can contribute directly. The outputs of requirements analysis will be the primary input for data warehouse modeling once the requirements validation phase is successfully accomplished.

The scope of work of requirement analysis can be summarized as follows:

- determining measures, facts, and dimensions including the dimension hierarchies
- determining granularities
- building the initial dimensional model. Overview of initial dimensional model is given in Figure 4.7.

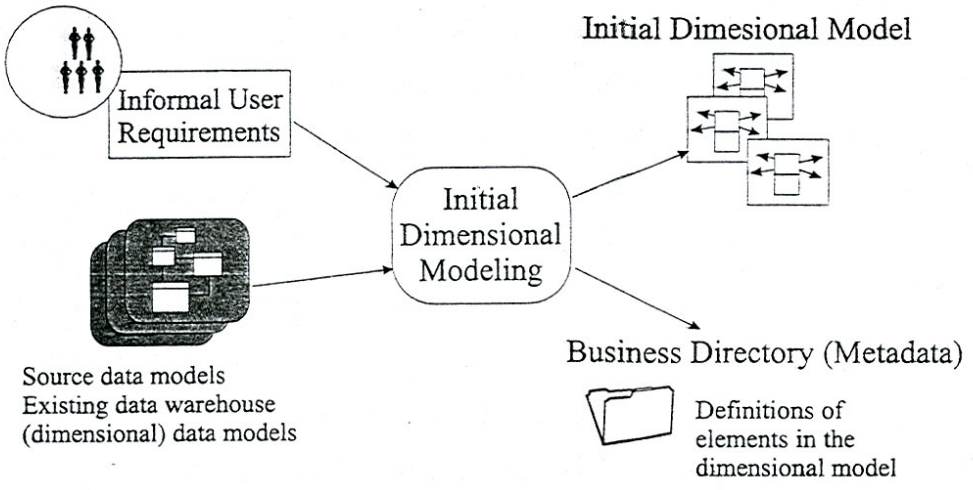


Figure 4.7. Overview of Initial Dimensional Model
(Source: Bayoğlu 2000, p.104)

The Figure 4.8 shows a notation technique that can be used to document the initial dimensional model schematically. It illustrates fact tables with the measures and the dimension hierarchies or aggregation paths associated with the facts. Dimension hierarchies are represented as arrows showing intermediary aggregation points. The dimensions may include alternate or parallel dimension hierarchies. Dimension hierarchies are given names drawn from the problem domain of the information analyst. These initial dimensional models also formally state the lowest level of detail –the granularity- of each dimension.

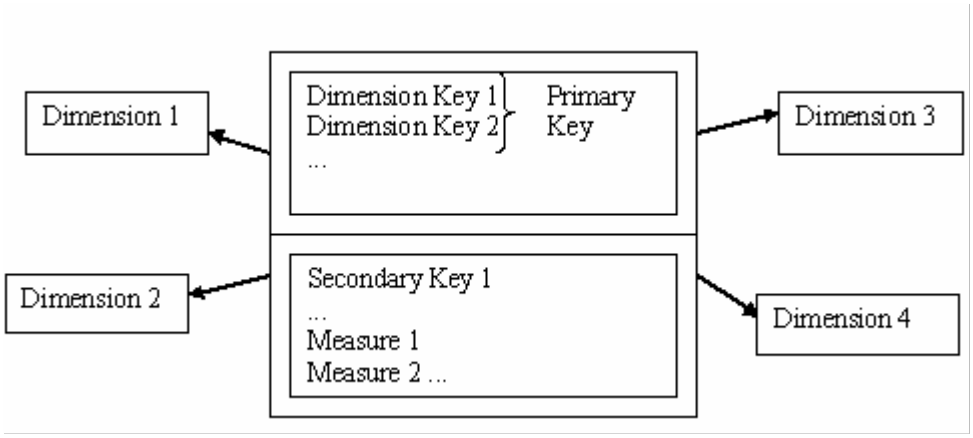


Figure 4.8. Schematic Representation of Initial Dimensional Models

4.5.2.1. Determining Measures, Dimensions and Facts

Building process of a dimensional model requires the declaration and arrangement of the following fundamental elements:

- Measures
- Dimension and dimension hierarchies
- Facts

The basic elements of a dimensional model can be determined by using several methods. In practice, the specialists merge the power of more than one approach to find the appropriate elements of the model. After this step, they integrate their works into an initial dimensional model that merges several separate views in reality.

Their way to identify the modeling elements is the key factor that forms the difference between these approaches. Here are some of the most widespread approaches:

- In ‘query-oriented approach’ the measures are determined initially and following this step, associated dimensions and facts are defined in order. The requirement specialists regard these client queries as the first source of inspiration, and that is why this approach is referred as ‘query-oriented’.
- Unlike the first approach, ‘business-oriented approach’ requires the facts initially, and then, the other two elements, dimensions and the measures are determined. Fundamental elements of the business problem domain (facts and measures) are determined at the beginning, and the remaining details are dealt with at the following stages.
- Third approach follows the procedure beginning with the dimensions, and it goes on with measures, and then comes to facts. This approach is commonly used when the source data models are chosen as a basis to determine the candidate elements of the initial dimensional model. The method mentioned in this section is known as ‘data-source-oriented approach’.

In this project, the ‘business-oriented approach’ is used during the identification of the elements of dimensional modeling. According to this approach, facts should be determined firstly.

Determining Facts

Facts are the key elements of a dimensional model, and they should be analyzed from a clear business scope. A clear vision of the problem domain can enable the end users to make profound analyses of the business area even beyond the requested and expected level. Those selected business facts may also support extension of the use of the data warehouse model to other end user problem domains.

Facts can indeed represent several fundamental things related to the business:

- A fact can represent a business transaction or a business event.
- A fact can represent the state of a given business object.
- A fact can also represent changes to the state of a given business object.

The fact table used in this study is the sum of all data collected while visiting the customers with the sales managers who work with the sales representatives in the operations of the company. The fact table of the field organization includes many measures that support business decision process, and this table regards the foreign keys as the primary keys of many dimensions. These keys will form a composite key for the fact table. Business objectives of the company are the purposes of the data warehouse

project, and these objectives require a clear determination of the necessary measures and dimensions in the following stage. Measures will be summarized appropriately for each specific level in the fact table, depending on the degree of summarization in the dimensions.

Facts are a pack of measures that are obtained from a function of the attributes that are found in various dimensions. These measures may be quantitative or qualitative. The fact table includes foreign keys as the primary keys of many dimensions. These keys are the main composite keys for the fact table. This fact table is highly normalized. The number of the fact tables is not limited, and many fact tables can be found in the same project. On the other hand, a star schema may have at least one fact table, and at least one measure is found within a fact table.

Business decisions are supported through these measures found in the fact tables. The measures are summarized appropriately for each level in the fact table depending on the degree of summarization in the dimensions. Because of this flexibility, 'drill' analysis at different levels may be performed. The 'Time Key' attribute from the time dimension also enables the analysts to perform a 'trend' analysis.

Determining Measures

Measures are elements of primary importance for a dimensional model. These measures are determined by the help of the end user queries and their requirements in general. They all should be expressed in business terms, and be understandable by the end users. Good measures are numerical, and they are usually involved in aggregation calculations, yet not every numerical attribute is a measure.

Following the interviews with the end users and related people, the measures that will be used are determined. In respect to the key performance indicators that were presented in the requirement-gathering step, the measures are as the following.

The measures may be qualitative or quantitative:

A qualitative measure is a text measure that is a distinct value for a combination of attributes from various dimensions. They are also called 'non-additive' or text facts. A qualitative measure cannot be the primary key of a fact table. In addition, a fact table may contain one or more qualitative measures. In the field organization fact table there will be one qualitative measure. This is 'precaution priority'.

Payment type code, status code, promotion code are secondary keys.

Precautions are actions that are given to the sales representatives by the sales managers if the company's standards are not implemented.

During the visit paid to the customer, as to result of control process, a certain number of precautions can be defined, and precautions have priority numbers ('precaution priority'). These priority numbers represent importance of the precautions.

There are two informational attributes in the fact table.

These are called 'precaution deadline' and 'precaution completion date'. 'Precaution deadline' is the last date until which the precaution has to be completed. The date on which the action has to be finished by the sales representative is 'precaution completion date'.

A quantitative measure is a numerical measure that may be a derivative of a mathematical aggregate function such as maximum, minimum, sum, average, or round at higher levels of summarization. They are also called 'additive' or numerical facts. A quantitative measure cannot be the primary key of a fact table. In addition, a fact table may contain one or more quantitative measures.

In the field organization fact table there will be four quantitative measures. These are 'precaution point', 'duration', 'endorsement' and 'number of visits'.

Every precaution priority has importance weight represented by a number called 'precaution point'. 'Endorsement' represents the monetary amount of sales sold to the customers by the sales representative. 'Duration' is time interval between beginning of the customer visit and end of the visit. 'Number of visits' attribute lets us calculate the number of visits paid to the customers and the number of actions defined for a sales representative in an arbitrarily selected period.

Determining the Dimensions

Dimensions are business aspects or 'variables' in a business that are used in deriving the measures. There is no limitation on the number of dimensions in a multi-dimensional model. The primary key of the dimension must exist as a foreign key in the fact table.

There are six variables or dimensions in the business. The dimensions required for measure interpretation are customer, employee, customer payment type, promotion, precaution and precaution status.

'Customer dimension' consists of data about customer type, identity knowledge and region. Sales representatives deliver the products of the company to the whole

country, and they sell these products. The company has different kinds of customers such as groceries, cafés, hypermarkets, organized trade etc.

The data about sales managers is stored in the 'employee dimension'.

Customer can pay the money in different types such as cash, credit etc. These data are stored in the 'payment type dimension'.

The company has standard promotion campaigns for customers in order to increase the sales. The knowledge of promotion type is stated in the 'promotion dimension'. For example, if the sales of a customer group fall, the company resorts to promotion activities such as giving complimentary products, and then how sales change is observed.

'Precaution dimension' includes descriptions of duties that are defined in accordance with the failed applications of the standards. For instance, the sales manager can give precaution that the sales representative should sell X product. It is important whether the company's standards are conformed to or not by the employees in sales and marketing branches, so these precautions are followed and considered as criteria for performance evaluation. The employees should complete these given precautions successfully and in a certain period.

Sales managers control whether the precautions, given preceding the visit, are completed or not. This status' description is gathered in 'status dimension'. The completed or uncompleted precaution ratios are considered while evaluating the performance of the employees.

A 'Time' dimension exists in our model as it does in every model. 'Time' information is stored in the fact table. That is why we only need to see the realized data on our reports. This dimension provides the ability to store historical information and perform trend analysis in a data mart. A 'Time' dimension also provides the ability to move from daily data to a monthly and/or yearly data. In addition, users may 'drill down' from summarized or aggregated yearly information to the monthly or daily level or vice versa, because the data is stored or summarized based on a hierarchical level such as day, month, and year.

4.5.2.2. Determining the Granularities

A data warehouse typically stores data in different levels of granularity or summarization depending on the data requirements of the business. If an enterprise

needs data to assist strategic planning, then only highly summarized data is required. The lower the level of granularity of data required by the enterprise, the higher the number of resources (specifically data storage) required to build the data warehouse.

The granularity of a measure can be defined intuitively as the lowest level of detail used for recording the measure in the dimensional model. Measures are usually associated with several dimensions. The granularity of a measure is determined by the combination of the recorded details of related dimensions. Different measures can have identical granularities. The granularity of a measure shows the depth at which end users will be able to perform information analysis using the data warehouse or the data mart. Determining the right granularities of measures in the data warehouse model has extreme importance. Because if the right granularity is chosen for recording data in the data warehouse model, a good detailed analysis of information can be obtained. However, it also increases the volume of data that will be kept in the data warehouse. It has great impact on the size of the data warehouse, and it affects the performance and resource consumption of end user activities.

In this project, all measures are elements of time. The granularity chosen for all measures is day. The employees work on the routes based on customer locations. Every day employees make a plan for their next visit, and they visit customers in accordance with this plan. All this work is done on a daily period.

4.5.2.3. Constructing the Initial Dimensional Model

The initial dimensional model can be created according to the measures and dimensions that were determined during the steps of gathering and analyzing the requirements. The most straightforward way of identifying facts is through combining their base dimensions. By this way, in this project the fact is identifiable naturally through combining the customers, promotions, and payment type, employees, precautions and status dimensions. The following Figure 4.9. illustrates the initial dimension model by using star schema.

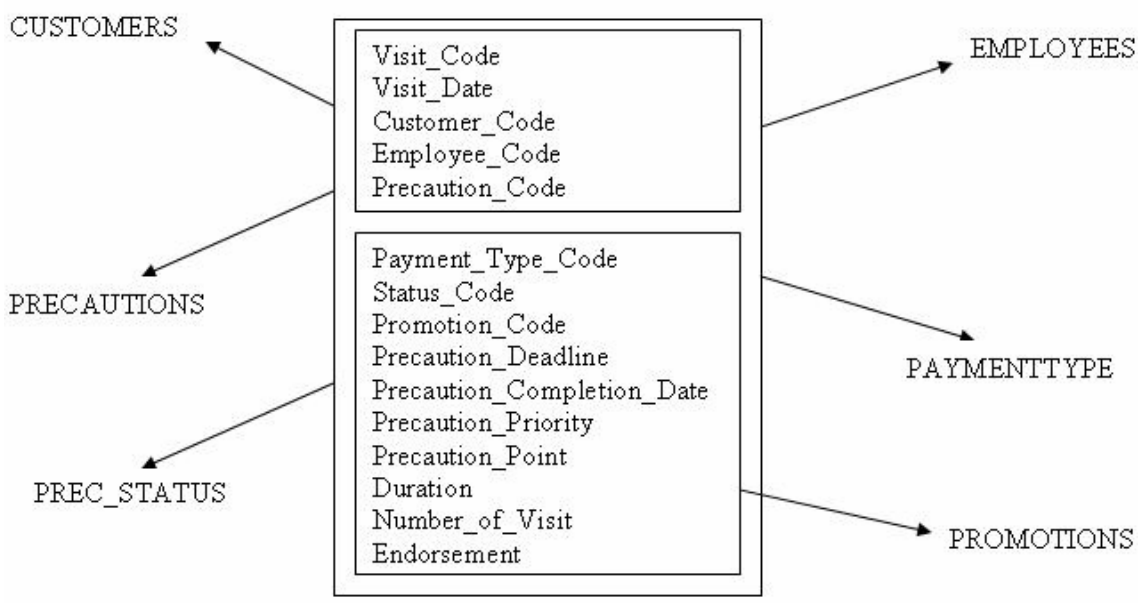


Figure 4.9. Initial Dimensional Model

4.5.3. Requirements Validation

Initial dimensional models are used in the process of validating the end user requirements and for assessing the scope and impact of the development project. At the requirements validation step, the results of requirement analysis are assessed and validated against the initially captured end user requirements. Besides, as a part of requirement validation, candidate data sources on which the end user requirements will have to be mapped are identified and recorded.

The initial model was analyzed with the end users in order to check the coherence and completeness of the initial dimensional model and validation against the end user requirements. The data items that take part in the initial model are available in the company's central operational source. By the help of all these investigations, the content and structure of the dimensional model is prepared for construction.

4.5.4. Modeling the Requirements

Validated initial models are further developed into detailed dimensional models, showing all elements of the model and their properties. Detailed dimensional models can further be extended and optimized. Many techniques in this area should be thought

of as advanced modeling techniques. Not every project requires all of them to be applied.

Requirement modeling consists of several activities that are performed with the intent of producing a detailed conceptual model representing the problem domain of the information analyst.

A detailed dimension model should incorporate all there is about the structure of the dimension as well as all of its attributes. One approach consists of producing the dimension models in the form of a flat dimension table. In this approach, the models are called star models or star schemas. In this project, these models are preferred for modeling the data mart. Another approach produces ER models that are called snowflake models or snowflake schemas.

While creating the dimensional model, Platinum ERwin 3.5.2 tool should be used.

ERwin has many powerful features that let the user design entity relation data models and dimensional models. With ERwin databases, many different target servers can be created and maintained. However, perhaps ERwin's most powerful feature is its simplicity and ease of use.

ERwin uses many of the standard Windows features and conventions. Just as a user can create, modify, save, and print documents in other Windows applications, the user can perform these same tasks in ERwin using familiar Windows dialogs.

Customer Dimension

Customer dimension describes every customer to whom the company's products are sold and delivered. As it is seen in Figure 4.10, location attributes, such as city, district, are used to define customers. Customers are in a district, and the district is in a city. This dimension has static hierarchy structure. The hierarchy has four levels and these levels are; customer type, city, district and customer name. So, this aggregation path is used to model the customer dimension.

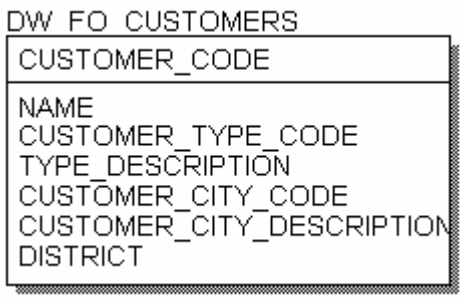


Figure 4.10. Customer Dimension

Employee Dimension

The descriptions of the sales managers and the sales representatives are stored in organizational hierarchy. In this hierarchy, there is a parent-child relationship. It has a flexible structure, thus a new employee can be added easily. Employee Dimension is depicted in Figure 4.11.

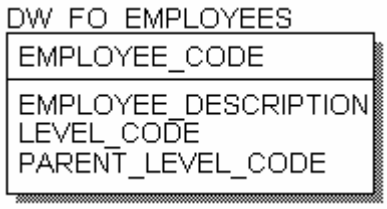


Figure 4.11. Employee Dimension

Promotion Dimension

One of the simplest dimensions is the promotion dimension as seen in Figure 4.12. Promotion description and promotion cost are stored in this dimension.

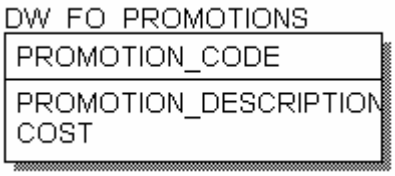


Figure 4.12. Promotion Dimension

Payment Type Dimension

The other simple dimension is payment type dimension in which definitions of the customer payment type are stored. Payment type dimension is shown in Figure 4.13.

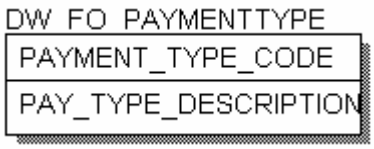


Figure 4.13. Payment Type Dimension

Precaution Dimension

Only one attribute called precaution description exists in this dimension as shown in Figure 4.14.

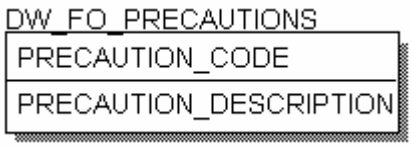


Figure 4.14. Precaution Dimension

Precaution Status Dimension

This dimension presented in Figure 4.15 stores definition of the precautions' status such as completed or not.

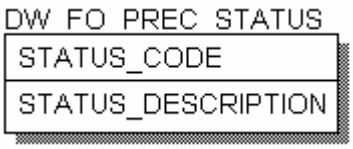


Figure 4.15. Precaution Status Dimension

Fact Table

The fact table contains the primary keys of the customer, employee, payment type, status, and precaution and promotion dimensions as foreign keys. It is

presented in Figure 4.16. These foreign keys are the composite primary keys for the fact table. Customer code, employee code, precaution code respectively belongs to customer dimension, employee dimension, precaution dimension. In addition, visit day code is kept as a primary key in order to designate the visit route. Visit date is stated in the fact table in order to make business analysis regarding time. A multi-dimensional model or a star schema should have at least one variation of the 'Time'.

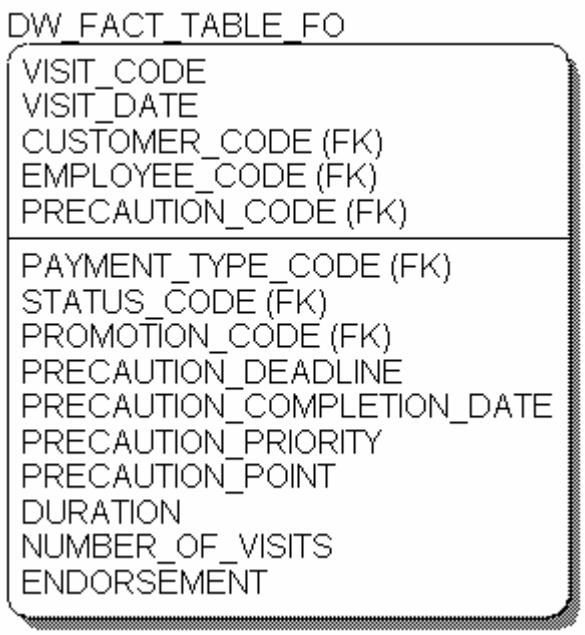


Figure 4.16. Fact Table

4.5.5. Design of the Data Mart

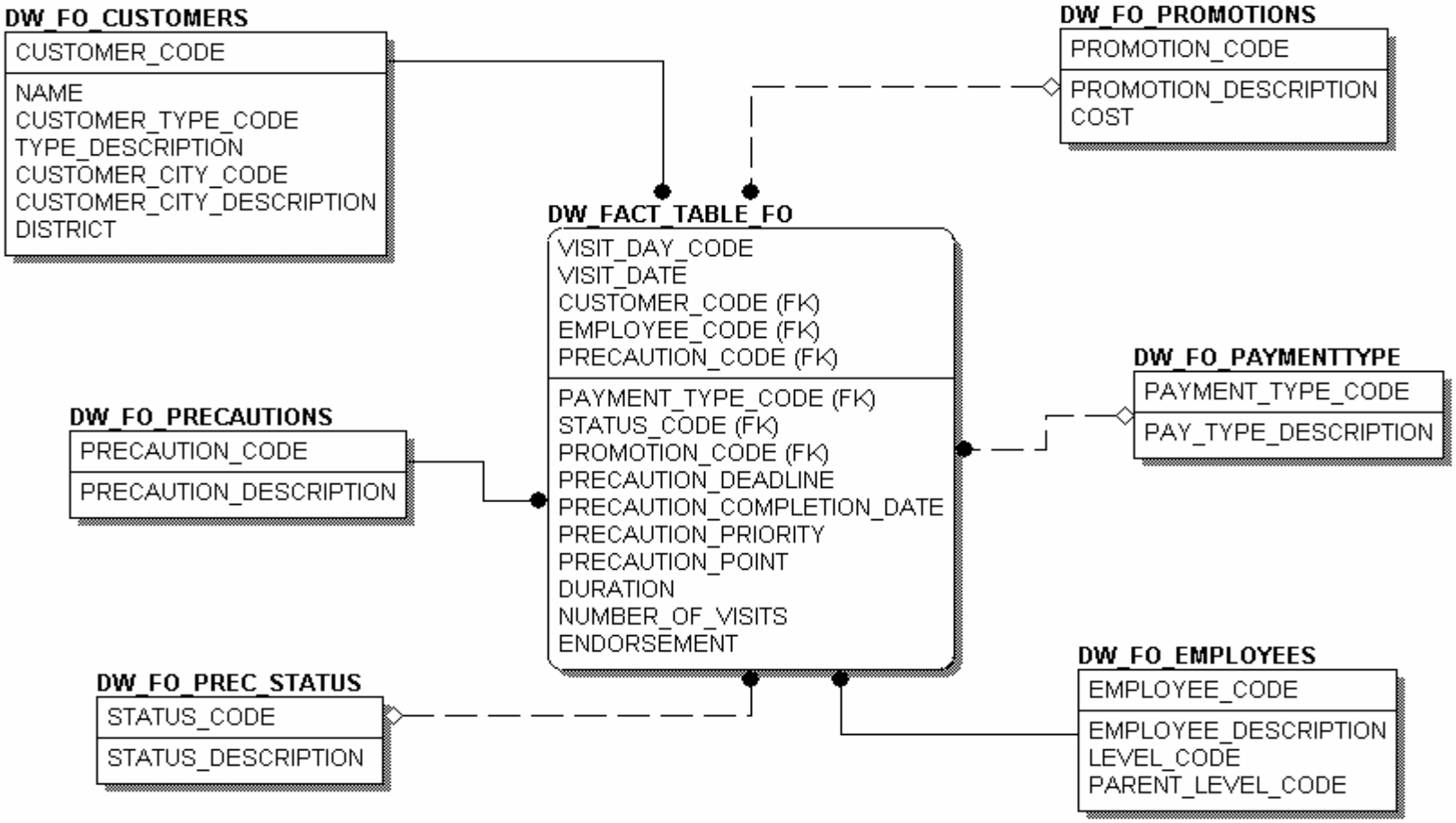


Figure 4.17. Dimensional Model for Company

The design of the dimensional model prepared for the company is shown in Figure 4.17. The model is composed of one fact table and six dimension tables, respectively DW_Fact table, customers, precautions, status, promotions, payment type and employees.

4.5.6. Creating the Data Mart Database

After the logical design of the database is completed, it should be created in DBMS. All the tables and its indexes are created in the SQL Server DBMS. These are presented in the appendices part. (Appendix B)

4.6. Data Loading and Transformation

In this step, the data that will populate to the data mart will be transformed from one operational source to the destination source. In the company, Oracle database server is used as the source database. The target source is SQL Server database and MS SQL Server Data Transformation Services (DTS) are used for ETL process.

Data Transformation Services

Data warehouse applications require the transformation of data from many sources into a cohesive, consistent set of data configured appropriately for use in data warehouse operations. SQL Server 2000 provides a powerful tool for such tasks. This tool is Data Transformation Services (DTS). DTS can provide access to data from a wide variety of sources, and transform it using built-in and custom transformation specifications.

When the data warehouse is about to start, the first data loading should be applied by using all the historical data in the operational source. To keep the data warehouse up-to-date, fact tables are updated every time according to the granularity of the data in the data warehouse.

Before the transformation process, the data source should be examined, and the tables that will be used for transforming the data should be designated.

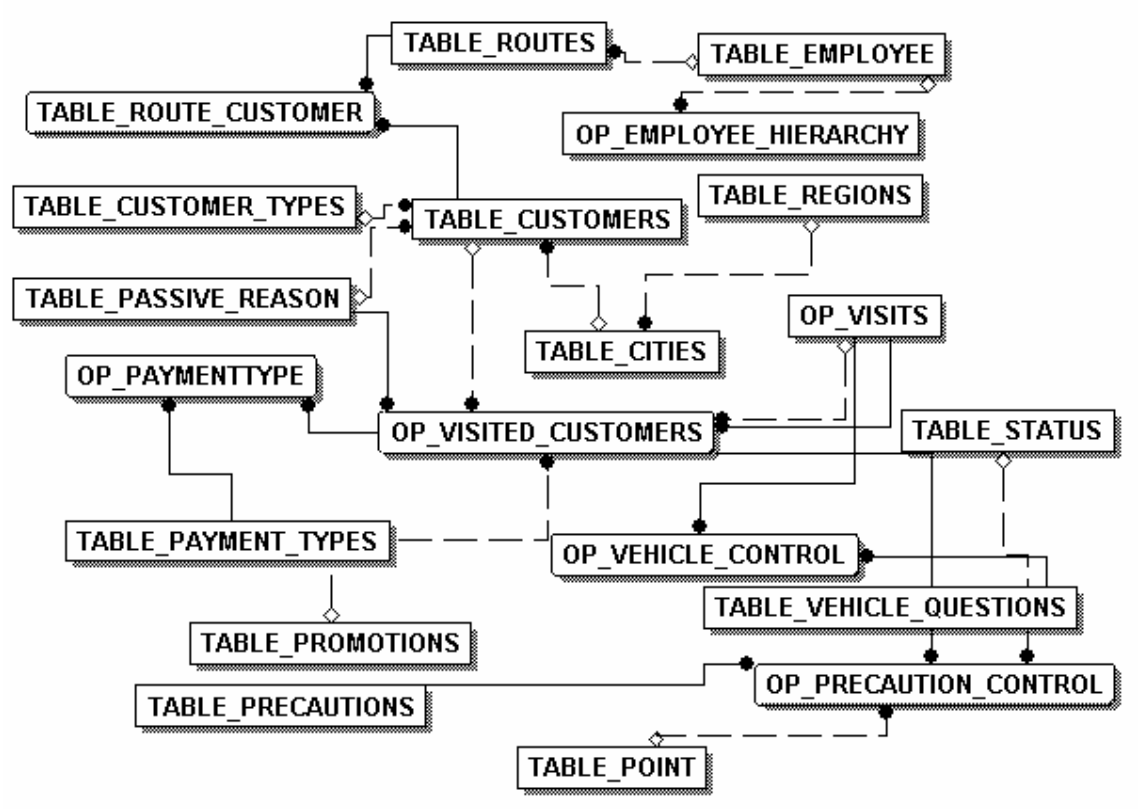


Figure 4.18. Operational Source ER Diagram

According to the designated tables in Figure 4.18, views are created to select the exact data from the operational source. These views, attached as Appendix C, are created in the operational environment, and after the creation process, they are used in the data transformation and loading processes.

4.6.1. Data Loading Process

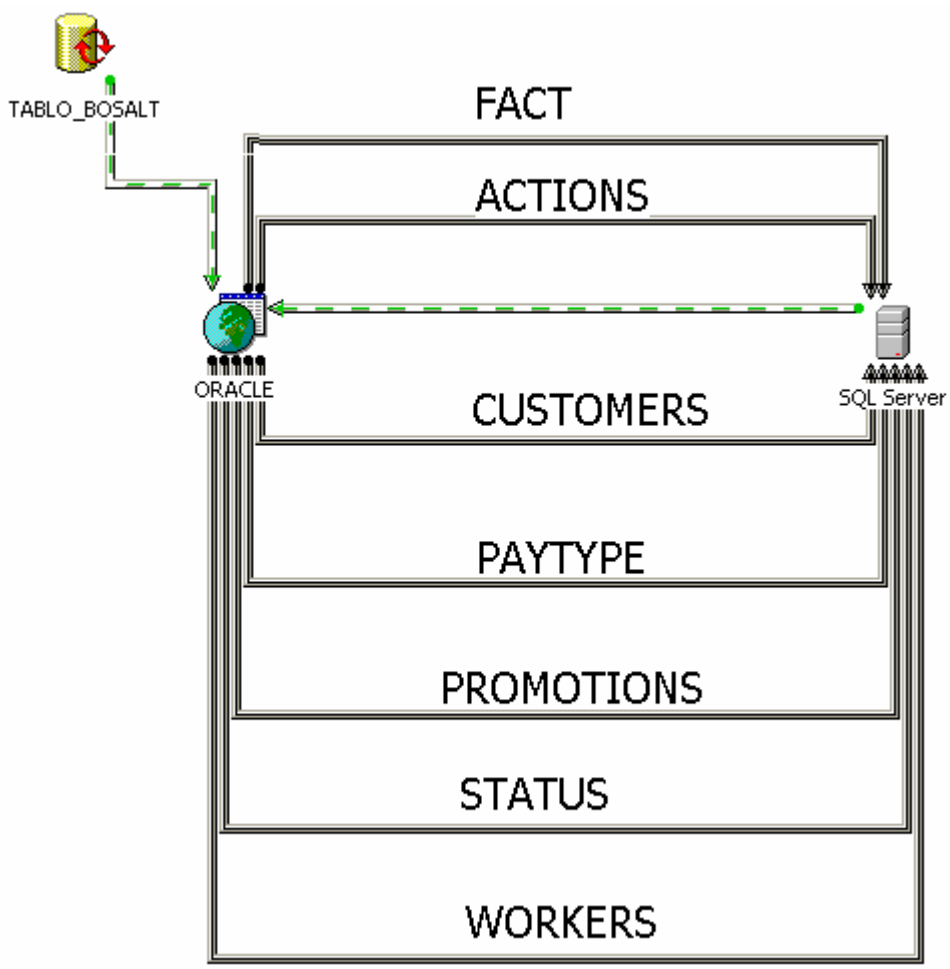


Figure 4.19. Transformation Package

Data kept in the operational source will be transformed to the data warehouse by using this transformation package (Appendix D). The direction of the data flow can be seen in Figure 4.19.

4.7. On-Line Analytical Processing (OLAP)

4.7.1. About Cube

Online analytical processing (OLAP) is a technology that uses multidimensional data representations called cubes to provide rapid access to data warehouse data. Cubes

model data in the dimension and fact tables in the data warehouse, and they provide sophisticated query and analysis possibilities to client applications.

Microsoft® SQL Server™ 2000 Analysis Services provide a powerful server and administrative tools to create and manage OLAP data and serve online client applications. Analysis Services also incorporate data mining algorithms that can analyze relational data in the data warehouse database and multidimensional data in cubes.

Cubes are the main objects in online analytic processing (OLAP), a technology that provides fast access to data in a data warehouse. A cube is a set of data usually constructed from a subset of a data warehouse, and it is organized and summarized into a multidimensional structure defined by a set of dimensions and measures.

A cube provides an easy-to-use mechanism for querying data with quick and uniform response times. End users use client applications to connect to an Analysis Server and query the cubes on the server. In most client applications, end users issue a query on a cube by manipulating the user interface controls that determine the contents of the query. This spares end users from writing language-based queries. Pre-calculated summary data called aggregations provide the mechanism with rapid and uniform response times to queries. Aggregations are created for a cube before end users gain access to it. The results of a query are retrieved from the aggregations, the cube's source data in the data warehouse, a copy of this data on the Analysis server, the client cache, or a combination of these sources.

Every cube has a schema that is the set of joined tables in the data warehouse from which the cube draws its source data. The central table in the schema is the fact table, namely the source of the cube's measures. The other tables are dimension tables, the sources of the cube's dimensions.

A cube can contain up to 128 dimensions, each with thousands or millions of members, and up to 1,024 measures. A cube with a modest number of dimensions and measures usually satisfies the requirements of end users. (Microsoft® SQL Server™ 2000 Analysis Services)

4.7.2. Creating the Cube

Creating a cube involves three steps:

4.7.2.1. Defining the Cube

The definition of a cube is based on the analytical requirements of end users. To define a cube, a fact table is selected, and measures are identified within the fact table. Then, dimensions are selected or created, each composed of one or more columns from another table.

First, a database in the 'Analysis Services' must be defined. This database is at the top of the object hierarchy in the 'Analysis Services Object Hierarchy', and it will include all the cubes, data sources, shared on non-shared dimensions, measures, calculated members, and other objects.

Second step includes the definition of the data source that is previously defined as 'Data Warehouse' structure. This data source will let us fill the cube with the appropriate data to be used in the analysis sessions using this OLAP cube. (Microsoft® SQL Server™ 2000 Analysis Services)

Our 'Field Organization Data Warehouse' application stores our fact tables, dimension tables and other required elements that will be used to build an OLAP cube. Dimension tables are used to define dimensions in the cube. Dimensions will be created regarding the structures of the tables lying in the "Data Warehouse" database by defining the dimension structure within the database.

Dimensions and Hierarchies:

Each cube dimension can contain a hierarchy of levels to specify the categorical breakdown available to end users. Dimension levels are powerful data-modeling tools, as they allow end users to ask questions at a high level, and then expand the dimension hierarchy to reveal more detail.

Customer dimension hierarchy:

Customer type description

City description

Customer district

Customer description

Time Dimension Hierarchy:

Year

Quarter

Month

Day

The following dimensions have one hierarchy level.

Worker Dimension: It is created with the quality of parent-child relationship, and it has one hierarchy level called Hierarchy Id. Hierarchy Id can be stated as a base for the Parent Hierarchy ID, as the field organization has vertical type of organization structure.

Precaution Dimension:

Precaution description

Payment Type Dimension:

Payment type description

Promotion Dimension:

Bonus name

Status Dimension:

Status description

Measures in Cube:

Finally, the measures and the calculated measures are defined. These are the quantitative (numerical) values helping us to make decisions on what we are trying to deduce from our database.

Number of visits

Endorsement

Point

Duration

Number of Actions

Calculated Members:

A calculated member is a dimension member whose value is calculated at run time using an expression specified when the calculated member is defined. Calculated members can also be defined as measures. Only the definitions for calculated members are stored; values are calculated in memory when an answer is needed for a query.

Calculated members enable addition of members and measures to a cube without increasing its size. Although calculated members must be based on the data (such as members) already existing in the cube, complex expressions by combining this data with arithmetic operators, numbers, and a variety of functions can be created. (Microsoft® SQL Server™ 2000 Analysis Services)

Action Success:

A calculated member as a measure named Action Success can be created in the cube by using the existing measure in the expression:

"([STATUS].CURRENT MEMBER, [Measures]. [Number of Actions])/ ([STATUS]. [All STATUS], [Measures].[Number of Actions])"

Visit Success:

A calculated member as a measure named Visit Success can be created in the cube by using the existing measure in the expression:

"[STATUS].CURRENT MEMBER,[Measures].[Number Of Visits]"

Number of Customers

A calculated member as a measure named number of customers can be created in the cube by using the existing measure in the expression:

"COUNT(
DISTINCT(
NONEMPTYCROSSJOIN(
DESCENDANTS([CUSTOMERS].CURRENTMEMBER,[CUSTOMERS].
[Customerdesc]),
DESCENDANTS([ACTION PRIORITY].CURRENTMEMBER,[ACTION
PRIORITY].CURRENTMEMBER.LEVEL),
DESCENDANTS([ACTIONS].CURRENTMEMBER,[ACTIONS].CURRENT
MEMBER.LEVEL),
DESCENDANTS([PAYMENT TYPE].CURRENTMEMBER,[PAYMENT
TYPE].CURRENTMEMBER.LEVEL),
DESCENDANTS([PROMOTIONS].CURRENTMEMBER,[PROMOTIONS].
CURRENTMEMBER.LEVEL),
DESCENDANTS([STATUS].CURRENTMEMBER,[STATUS].
CURRENTMEMBER.LEVEL),
DESCENDANTS([TIME].CURRENTMEMBER,[TIME].
CURRENTMEMBER.LEVEL),
DESCENDANTS([WORKERS].CURRENTMEMBER,[WORKERS].

CURRENTMEMBER.LEVEL)
)))”
The Figure 4.20 illustrates cube structure.

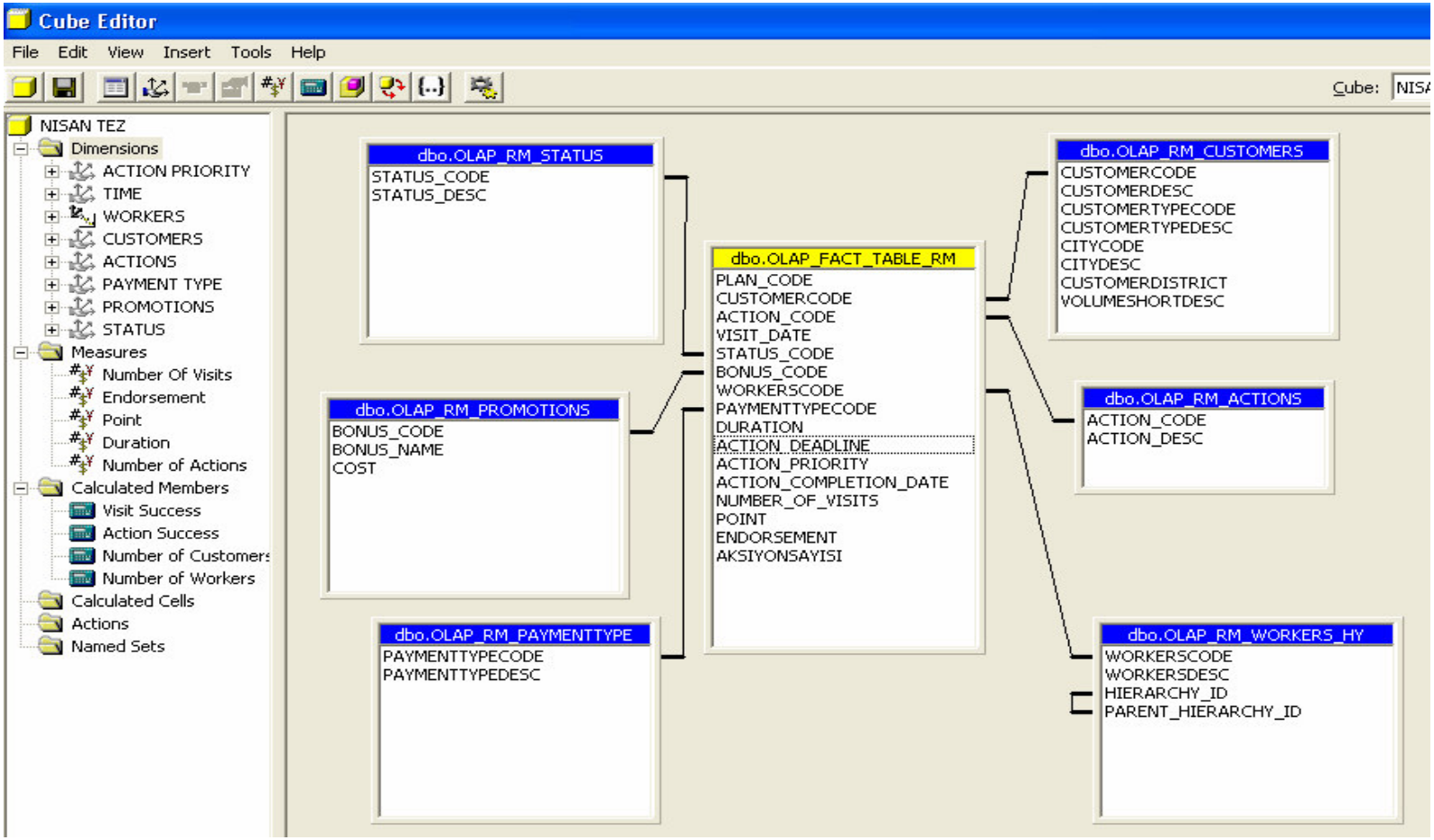


Figure 4.20. Cube Structure

4.7.2.2. Specifying Storage and Aggregation Options

After defining the cube, its aggregations that are pre-calculated summaries of cube data can be designed. Designing the aggregations specifies the summarization strategy.

The goal is to design the optimal number of aggregations. This number should not only provide satisfactory response time, but also prevent excessive partition size. A greater number of aggregations produce faster response time, but it also requires more storage space. Microsoft® SQL Server™ 2000 Analysis Services provides a wizard to specify storage and aggregation options.

At first, a storage option should be selected. The options of the OLAP storage are explained briefly in Table 4.1.

Table 4.1. OLAP Storage Options
(Source: Microsoft® SQL Server™ 2000 Analysis Services)

Storage option	Description
MOLAP	Multidimensional OLAP (MOLAP) stores aggregations and a copy of the partition's source data in a multidimensional structure on an Analysis server computer.
ROLAP	Relational OLAP (ROLAP) stores aggregations in a relational structure and leaves the partition's source data in its existing relational structure.
HOLAP	Hybrid OLAP (HOLAP) stores aggregations in a multidimensional structure on an Analysis server computer and leaves the partition's source data in its existing relational structure.

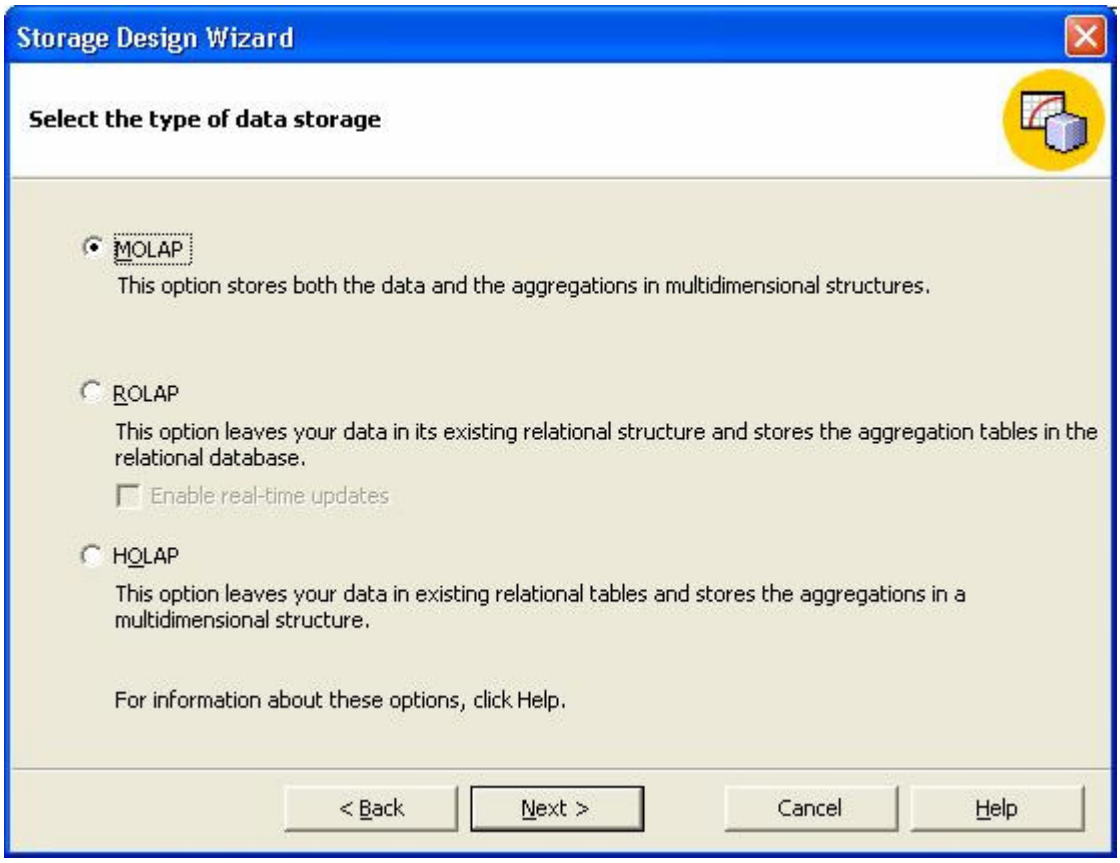


Figure 4.21. Selection of the Data Storage Type
(Source: Microsoft® SQL Server™ 2000 Analysis Services)

The print screen of the type of data storage selection is given Figure 4.21.

In this study, MOLAP was chosen due to the necessity for rapid query response. Besides, MOLAP is more appropriate for partitions in cubes with frequent use.

ROLAP is a technique used where instant data are very important for the decision-making processes over data warehouse reports. For example, data warehouse applications used in large, chain companies dealing with 'Retail Sales' need to see instant data of the sales to assess employee performance results during a workday. This kind of system should be completed with a 'Balanced Scorecard' application which is fed by data warehouse application and can be accepted as an 'Enterprise Performance-Information System'.

HOLAP is a collective method built up with some properties taken from ROLAP and some taken from MOLAP.

Data especially collected for this project do not vary between small periods. On the other hand, gaining instant data is not so important, as it is not necessary to see the employees' performance at the same time the data entered. Therefore, MOLAP is

selected to use in our data warehouse application's storage model. (Microsoft® SQL Server™ 2000 Analysis Services)

In order to control the number of aggregations, one of the following available methods in the wizard is chosen. It is shown in Figure 4.22.

- Specify a storage space limit for the aggregations
- Specify a performance gain limit
- Stop the wizard manually

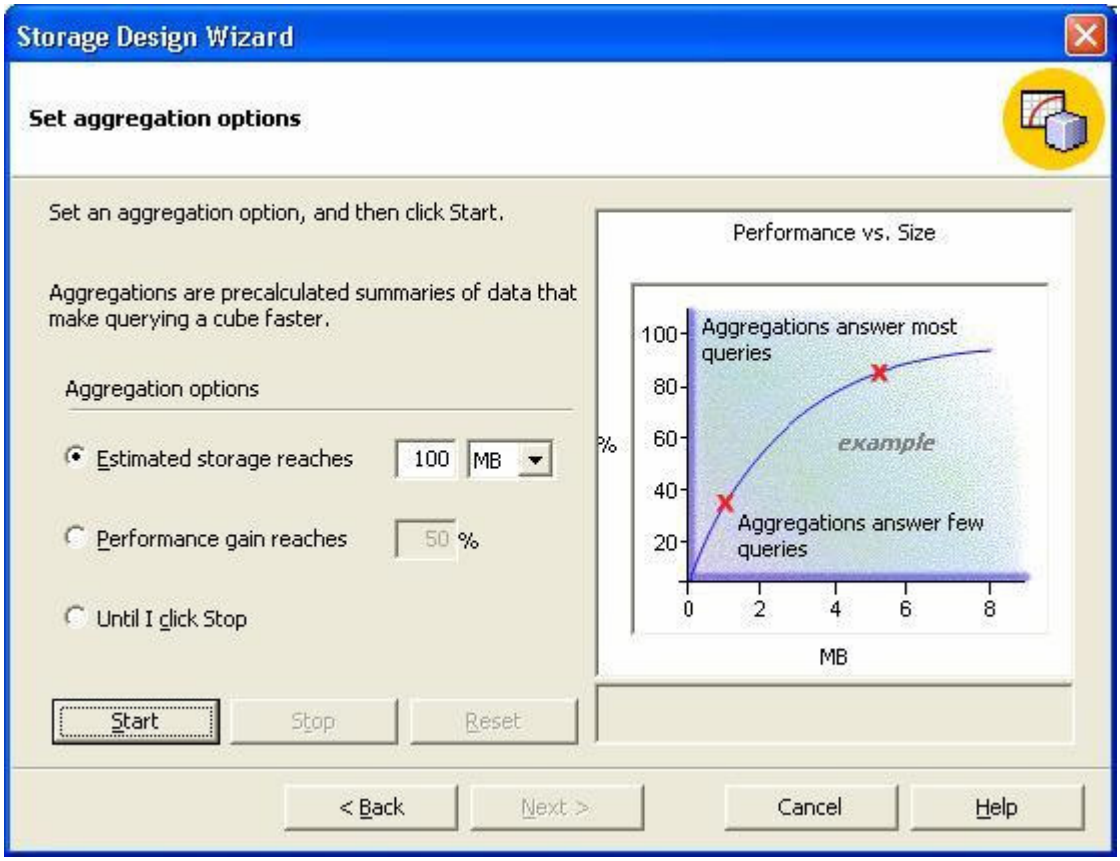


Figure 4.22. Set Aggregation Options
(Source: Microsoft® SQL Server™ 2000 Analysis Services)

Estimated storage reaches

Enter the amount of hard disk storage to allocate for storing the aggregation tables. A maximum storage size in either megabytes (MB) or gigabytes (GB) can be entered.

Performance gain reaches

Specify the percentage amount of performance gain for the queries. This amount represents the percentage improvement between the maximum and minimum query times.

Until I click Stop

Select to manually control the balance.

4.7.2.3. Processing

After designing the aggregations of a new cube, the cube should be processed. When the cube is processed, the aggregations designed for the cube are calculated, and the cube is loaded with the calculated aggregations and data. Processing the cube involves reading the dimension tables to populate the levels with members from the actual data, reading the fact table, calculating specified aggregations, and storing the results in the cube. After the cube is processed, users can query it.

4.7.3. Client Side and Report Examples

At this step, ProClarity Analytic Platform 4.0 Desktop Client is used to present the data to the end user.

ProClarity Desktop Client is a powerful multidimensional analysis tool developed specifically for analyzing business data. In order to spot trends and identify potential problem areas, large amounts of data can be analyzed by using ProClarity Desktop Client. Besides, business decision makers can communicate this information to others within the organization.

A few report examples created by using ProClarity are presented in Appendix E.

CHAPTER 5

DISCUSSION

This study was prepared to give information about the design and development of a data warehouse using sales database and requirements of a retail group company that needs a decision-making system. As a start, a survey has been done about the terminology and the implementation of the data warehousing. Application studies began after getting familiar with the data warehousing.

First, interviews with related people were organized. Employees and managers were very interested in and curious about the issue. Thus, there was no resistance against the implementation of the project. Especially the Sales Representatives had been using the Pocket PCs with pleasure for a long time, so acquiring the data continuously did not occur as a problem. All of the employees in the company helped during the project as much as they could. The contribution of the employees was one of the important factors that contributed to the successful implementation of the project. Despite the good profile, a few minor problems arouse during the application of the project due to the abundant and complex operational source data. It was hard to analyze the database, as its structure was sophisticated and incomprehensible. Although these difficulties existed, the analysis of the requirements and current state were concluded, all requirements were revealed and Key Performance Indicators (KPIs) were determined clearly and successfully.

Before the requirements validation step, an initial data model was determined. The Erwin tool was selected as a modeling tool. The purpose of the design phase was to create a blueprint for the components that comprise the data warehouse. At the end of this part, the sufficient detail level of the design was retrieved to construct the system.

After the analysis of the requirements and design phase were complete, requirements validation began. The objective of this step was to demonstrate that the model addresses the business problems and satisfies user requirements. The business problems were addressed, and the deliverables were produced during the analysis and design phases. So the requirements were validated, and data warehouse construction began.

The goal of the next stage was to create the data warehouse. First, a data warehouse database was created, and then historical data were loaded into the database. Operational source data kept in ODMS were transferred into the data storage area by using Stored Procedure (SP) written with PL/SQL. They were prepared as consistent data that were populated to the data warehouse by using MS SQL Server Data Transformation Services (DTS). MS SQL Server 2000, which also includes an integrated set of tools for data transformation between the origin and destination warehouse, was chosen as the data warehouse development tool.

After loading and inquiring performance was validated, the cube that organizes and summarizes data for efficient analytical querying was created in order to have multidimensional presentation of the data warehouse. MS Analysis Services (Online Analytical Processing (OLAP)) was selected as the OLAP development tool.

Finally, training was updated. This is an important component of data warehouse implementation to ensure the success of the project. Business users are trained to use the warehouse for meaningful analysis and decision support. While presenting the data to the users, ProClarity Analytic Platform 4.0 Desktop Client that is powerful and easy-to-use was chosen.

The tools used in this project were supplied by the company.

During the preparation of this thesis, original data were used; yet, here the data that were changed are presented. It is a crucial and a critical point for reasons regarding security. The data are kept as a secret due to the competition among the retailer companies.

Difference between Business Intelligence Software of Oracle, IBM and Microsoft

Companies have begun to make serious investments in “Business Intelligence (BI) Software”. According to some researches, in 2004, the most popular ones of these products were; Oracle, with a rate of 41%, IBM, with a rate of 31% and Microsoft, with a rate of 13%. The latest products of these three important vendors are:

IBM DB2 8.2 Data Warehouse Edition includes Cube Views, AlphaBlox, Intelligent Miner Modeling, Visualization and Scoring, Query Patroller, WebSphere Information Integrator, Office Enterprise Connect for Excel, Data Warehouse Manager (ETL and scheduling). Microsoft SQL Server BI composed of Analysis Services, Reporting Services, Integration Services, BI Intelligence Studio, Report Builder. Significantly, Microsoft also plans real-time reporting tools. Oracle Database &

Business Intelligence Server 10 has OLAP engine, Data Mining engine, Discoverer, Reports Services, Spreadsheet add-in, BI Warehouse Builder and BI Beans. The core relational database platform, which includes embedded data and text-mining algorithms, statistical functions, and an OLAP engine, making it possible for enterprises to perform powerful analytics without having to move data to expensive, stand-alone analytical servers. WEB_15 (2005), WEB_16 (2005)

It is not easily possible to differentiate the BI database packages among the major players, except perhaps in the client side developer tools. The features that all three vendors implement are the BI-based capabilities that are being added to database engines, such as materialized views, bitmap and other BI-oriented indexing, and special partitioning and clustering schemes.

Microsoft added free OLAP Services into SQL Server 7 and, thus, started a tendency toward combining BI capabilities with its database in a major way. With SQL Server 2000, Microsoft more than doubled the ante by providing new data mining features along with a more powerful DTS (data transformation services) engine. Now Microsoft is offering in both SQL Server 2000 and 2005 its newly updated Reporting Services, and it also includes a fairly sophisticated drag-and-drop report writer and notable enhancements to Analysis Services' data-mining capabilities and DTS' (now called Integration Services) ETL programmability.

In addition to the new features, the old features, such as price, performance and reliability still affect customers' choices in buying the product. Performance has always been an important factor in database systems. The Transaction Processing Performance Council (TPC)'s TPC-H benchmarks measure a decision-support/BI-oriented mix of queries. IBM here again leads the way, however among the largest databases Oracle comes to the fore. Again, these measures demonstrate that things are different now from the time when either Microsoft or Oracle dominated the low size databases, and IBM or Teradata dominated the large TPC-H database test sizes. Here is a summary of results: WEB_15 (2005)

> 100GB database - IBM DB2 takes top 4 spots and best price/performance
> 300GB database - IBM DB2 takes top 7 spots and best price/performance
> 1000GB database - IBM DB2 takes top spot, Oracle the next 3, IBM best price/perf.
> 3000GB database - Oracle takes top 7 spots and best price/performance
> 10000GB database - Oracle takes 1, 3 and best price performance; IBM takes the 2nd spot

The table makes it clear that Oracle is right in its claim that the grid-computing approach can raise their databases to new performance levels. Microsoft is seen below the top five rankings, and then in only two of the TPC-H benchmarks. However, the company is promising new benchmarks for SQL Server 2005. If new 64-bit processing and OS and grid and clustering capabilities are taken into consideration, it can be said that users should expect a new level of benchmarks from all of the vendors; but clearly, Microsoft has the most ground to make up.

It is not easy to measure reliability and customer satisfaction in the database arena; yet, Microsoft seems to be the leader in customer satisfaction. In addition, Microsoft has always been a leader in the area of price. However, it is often claimed that IBM and Oracle have taken the lead in advanced database control for such tasks as recovery, failover, backup, and ongoing system maintenance through self-healing features. Clearly with their cross platform offerings, IBM and Oracle offer more choices to customers to get the exact OS+database package with the availability, reliability, security, price/performance and other key factors they prefer. WEB_15 (2005)

Above, the opportunities that IBM, Oracle and Microsoft Business Software present are explained. As a result, a sharp differentiation is not possible. The product that the company will use or should use will change according to the structure of the company, the performance it desires, investment force and its plans for the future. A decision must be reached after a meticulous investigation by taking into consideration the advantages and disadvantages of the product. At the project that is carried out, products of Microsoft satisfied the needs of the company, and the data warehouse that is designed is being successfully used.

CHAPTER 6

CONCLUSION

6.1. Results of the Study

The data warehouse designed and developed in this study is used by Sales Department and Human Resources Department. The field organization data were stored in the data warehouse, and they were controlled by the Sales Managers as a policy of the company.

The answers of the complex queries and the weekly or monthly reports presented using ProClarity tool are assessed by Sales Department and Human Resources Department. The managers and executives are generally concerned with aggregated reports that easily reveal characteristics of the data warehouse whereas sales chief or human resources personnel want to get detailed reports. A few examples of report are given in the Appendix E.

Analysis of the historical data by using this operational source structure was difficult. This data warehouse model has the ability to organize and store the data needed for informative analytical processing over a long historical time perspective and comparing the historical data. This historical data is used for analysis that supports business decisions at many levels, from strategic planning to performance evaluation of the organization unit.

The delivery channel in the Sales-Delivery Network are optimized and improved according to deliverable of the data warehouse. The positions of the rival companies in the retail sector may be known and traced by using the data obtained from the customer. For example, it is checked whether a product belonging to a rival is available at the company's customer or not.

Controls of some properties of the products such as refreshment and availability are tracked easily. The condition of the assets like stands is inspected. The customers' complaints are learned during the customer visits. Sales Managers check the accuracy of the data, determine the problems, errors, and wrong business strategies, and then improve methods to prevent the repetition of deficiencies.

The Human Resources Department has improved a premium system for sales representatives by using the data warehouse. Thus, according to their performances, incomes of the sales representatives are increased.

There are 6 dimensions at the data warehouse model that is designed. There are 4 levels at the time dimension; year → quarter → month → date. At the customer dimension, the four levels are; region → province → customer type → customer. Employees dimension includes five levels and these levels are; General Director → Assistant of the General Director of the Sales Department → Director of a Region → Sales Director → Sales Representative. The remaining 4 dimensions are; promotions, payment type, precautions, prec_status and they all have one level. There are 8 measures in the model. This data warehouse model, if desired, presents the opportunity of taking 640 reports –with a simple calculation 4*4*5*1*1*1*1*8=640- at the smallest detail level with the exception of the filtration. Let us suppose that 1 day on average is necessary to prepare 1 report. If we consider that the employees working at Information Technology Department tried to prepare this report by the help of operational sources, there would be a serious necessity of sources regarding time and workforce. In addition, the slightest change in the report would mean an extra burden. Thanks to this project, the burden on the reporting system has been relieved. Physically, the load on hardware has also been lessened.

By using the present system of the company, 32 reports could be taken. The average working time for these reports was 8 minutes. Some reports could be taken in 1 minute whereas others took 18-20 minutes. It was observed that during busy periods, the time needed rose to 27-28 minutes. As a result of the tests conducted, within the system that is being prepared, the working time for a report that works at the bottom level and requires maximum time has been determined as 18 seconds. Employees working at the technical department at the server management tested the performance of queries and reports by performance tools and results that supported the arguments listed above were obtained. The results were pleasing for the company.

6.2. Future Work

Enterprise data warehouse: At the beginning of this project, methods of designing and developing sales data warehouse were desired to be used for whole enterprise, but this had to be skipped and left for the studies that will be carried out in the future. Because the current database structure had no good design, and the data kept in the operational source were not accurate, consistent and continuous. This problem should be solved initially, and then new departmental data marts may be created or new dimensions may be added to this system. For example, all products and their sales information should be handled in order to make sales and products analysis for customers or districts. As for the time dimension, the seasonal trends can be found by making product-sales analysis.

Route Optimization: The routes are planned before the visits of the sales representatives'. The detailed route plan can be prepared by using operations research algorithms. For example, the total time of the whole plan and the total time that is spent at the visit can be handled. The difference between them gives us the sales representative's free time. The remaining time may be reduced by optimizing the route planning.

Balance Score Card: Balance Score Card is a tool used for measuring a company's activities in terms of its vision and strategies. This card reveals whether a company and its employees achieve the results set forth by the strategy or not. It also helps the company express the necessary objectives and initiatives to support the strategies.

The balanced score card suggests that the decision-makers view the organization from four perspectives, and they develop metrics, collect data and analyze it according to each of these perspectives:

- The Learning and Growth Perspective
- The Business Process Perspective
- The Customer Perspective
- The Financial Perspective

A new data warehouse model may be developed by taking the requirements of the field organization or improvements as perspectives of the Balance Scorecard. In this way, the Sales Representatives premium system can be improved, and a new performance analysis system can be set for all employees in the company. Therefore,

the premium system can be kept up-to-date, a prize system can be formed, and competition can be created among the employees. All of these will be useful for the progress of the company.

REFERENCES

Ahmad, I. and Azhar, S., 2002. "Data Warehousing In Construction: From Conception To Application", First International Conference on Construction in the 21st Century (CITC2002) Challenges and Opportunities in Management and Technology, Miami, Florida, USA, (25-26 April, 2002).

Bain, T., Benkovich, M., Dewson, R., Ferguson, S., Graves, C., Joubert, J.T., Lee, D., Scott, M., Skoglund, R., Turley, P., Youness, S., 2000. *Professional SQL Server 2000 Data Warehousing with Analysis Services*, (Wrox Pres).

Bayoğlu, L., June,2000. *Management Information System and Data Warehousing*, A Master Thesis In Computer Engineering, Dokuz Eylül University, İzmir, June 2000.

Bocutoğlu, E., Atasoy Y., 1999. *Yükselen Süpermarket Olgusu Karşısında Bakkaliye Sektörünün Yeri ve Trabzon Örneği*, Karadeniz Teknik Üniversitesi.

Chrysafis, T., 2003. "On-Line Analytical Processing", Second International Student Spring Symposium on Contemporary Topics In IT (CTIT), Thessaloniki, Greece, (28 February-1March 2003).

Govindaraj, T., Blanco, E.E., Bodner, D.A., Goetschalckx, M., McGinnis, L.F., Sharp, G.P., 2000. "An On-Line Tutor for Warehouse Design", Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics, (8–11 October 2000), Nashville, Tennessee, USA, pp. 1158–1162.

Kendall, J.E., Kendall, K.E., 2002. *System Analysis and Design*, (5th ed.), (Pearson Education Inc., Upper Saddle River, New Jersey).

Kimball, R., Reeves, L., Ross, M., Thornthwaite, W., 1998. *The Data Warehouse Lifecycle Toolkit*, (John Wiley & Sons, Inc.).

Lane, P., Schupmann, V., 2002. *Oracle9i Data Warehousing Guide*, Oracle Pres.

Lechtenbörger, J., 2003. "Data Warehouse Schema Design", Proceedings of the 10th Conference on Database Systems for Business, Technology and WEB, Leipzig, (26-28 February 2003).

Mehler, G., 2002. “Data Warehousing for the Enterprise”, Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference (SUGI27), Orlando, Florida, (14-17 April, 2002), SAS Institute Inc., Cary North Carolina, pp. 143-27.

Microsoft Corporation, 2000. *Microsoft SQL Server 7.0 Data Warehousing Training*, (Microsoft Pres).

Microsoft Corporation, 2000. *Programming a Microsoft SQL Server 2000 Database Workbook*, (Microsoft Pres).

Molina-Garcia, H., Ullman, D.J., Widom, J., 2000. *Database System Implementation*, (Prentice-Hall Inc., New Jersey).

Moss, T.L., Adelman, S., 2000. *Data Warehouse Project Management*, (Addison-Wesley)

Murray, D., Anahory, S., 1997. *Data Warehousing in the Real World: A Practical Guide for Decision Support Systems*, (Addison Wesley, Massachusetts).

Poe, V., Klaver, P., Brobst, S., 1998. *Building a Data Warehouse for Decision Support*, (Prentice Hall, Upper Saddle River, New Jersey).

Tek, O.B., 1984. *Perakende Pazarlama Yönetimi*, (Üçel Yay., İzmir).

Zikmund, W.G., 2003. *Business Research Methods*, (5th ed.), (Thomson South-Western, Oklahoma State University).

WEB_1, (2004). DSS Lab, 10/02/2004. O'Donnell, P., Monash University, Introduction to Data Warehousing. http://www.sims.monash.edu.au/subjects/IMS5024/lectures/week10slides5024.pdf

WEB_2, (2004). Infrocentric web site, 13/01/2004. Wells, D.L., Developing the Data Warehouse Incrementally, http://www.infocentric.org/articles/developing_dw_incrementally.PDF

WEB_3, (2004). Infrocentric web site, 13/01/2004. Thomann, J.,& Wells, D.L., Evaluating Data Warehousing Methodologies: Objectives and Criteria, http://www.infocentric.org/articles/dw_methodology_1of3.PDF

WEB_4, (2004). Infrocentric web site, 13/01/2004. Thomann, J.,& Wells, D.L., Evaluating Data Warehousing Methodologies: An Evaluation Process, http://www.infocentric.org/articles/dw_methodology_2of3.PDF

WEB_5, (2004). Infrocentric web site, 13/01/2004. Thomann, J.,& Wells, D.L., Evaluating Data Warehousing Methodologies: Guidelines for Success, http://www.infocentric.org/articles/dw_methodology_3of3.PDF

WEB_6, (2004). Infrocentric web site, 13/01/2004. Wells, D.,& Thomann, J., The Keys to the Data Warehouse, http://www.infocentric.org/articles/keys_to_dw.PDF

WEB_7, (2004). Popkin Software (*White Papers*) for ITtoolbox Data Warehouse, 29/01/2004, Data Warehouse Design: Technology, Techniques and Tools, http://dw.ittoolbox.com/documents/document.asp?i=1104

WEB_8, (2004). W. H. Inmon (*White Papers*) for W. H. Inmon, 5/12/2003, Data Marts and Data Warehouse: Information Architecture for the Millenium, http://dw.ittoolbox.com/documents/document.asp?i=937

WEB_9, (2004). Online Analytical Processing (OLAP). CSWL Inc., http://www.olap.com

WEB_10, (2004). Center for Information Management, Integration and Connectivity, 10/06/2005. The State University of New Jersey Rutgers, http://cimic.rutgers.edu/~gusadi/design/

WEB_11, (2004). OLAP Report's Web Site, 5/12/2003. http://www.olapreport.com

WEB_12, (2004). Altaplana's Web Site, 5/12/2005. www.altaplana.com/olap

WEB_13, (2005). WisdomForce Technologies web site, 25/07/2005. Gornshtein, D.,& Tamarkin, B., Features, Strengths and Weaknesses Comparison between MS SQL 2005 (Yukon) and Oracle 10g Databases, http://www.wisdomforce.com/dweb/resources/docs/MSSQL2005_ORACLE10g_compare.pdf

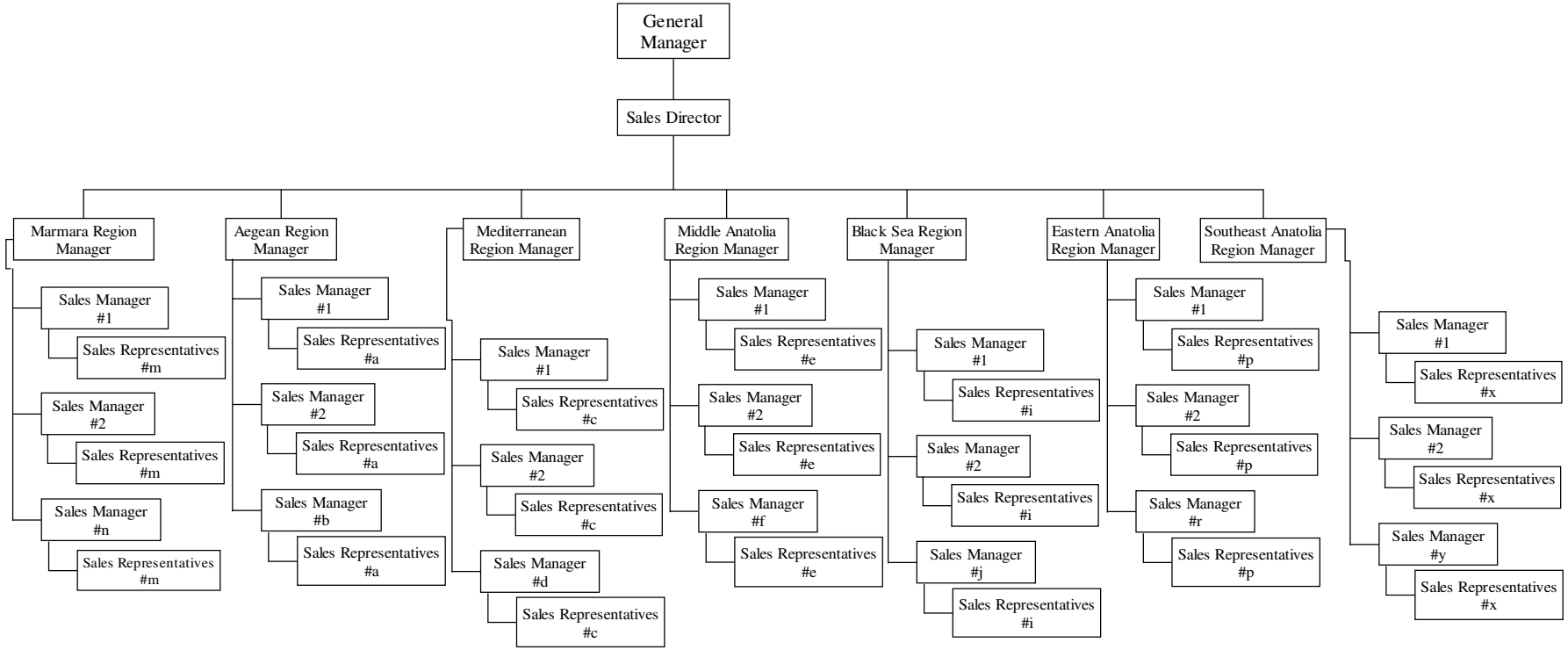
WEB_14, (2005) California State University web site, 25/07/2005. http://www.csulb.edu/~deesun/is480/OracleSQL.htm

WEB_15, (2005) Bizintelligencepipeline web site, 25/07/2005. Surveyer, J., BI: The New Database Differentiator, http://www.bizintelligencepipeline.com/howto/163105687

WEB_16, (2005) Oracle web site, (25/07/2005). Joch, A., Eye on Information, http://www.oracle.com/technology/oramag/oracle/05-jan/o15eye.html

APPENDIX A

ORGANIZATION CHART OF THE SALES FIELD ORGANIZATION



APPENDIX B

DATAMART DATABASE

Status Dimension View

```
DROP TABLE DW_FO_PREC_STATUS

CREATE TABLE DW_FO_PREC_STATUS (
    STATUS_CODE         int NOT NULL,
    STATUS_DESCRIPTION   varchar(100) NULL)

ALTER TABLE DW_FO_PREC_STATUS
    ADD PRIMARY KEY (STATUS_CODE)
```

Customer Dimension View

```
DROP TABLE DW_FO_CUSTOMERS

CREATE TABLE DW_FO_CUSTOMERS (
    CUSTOMER_CODE       int NOT NULL,
    NAME                varchar(50) NOT NULL,
    CUSTOMER_TYPE       real NOT NULL,
    TYPE_DESCRIPTION    varchar(50) NOT NULL,
    CUSTOMER_CITY_CODE  varchar(10) NOT NULL,
    CUSTOMER_CITY_DESCRIPTION varchar(50) NOT NULL,
    DISTRICT            varchar(20) NULL)

ALTER TABLE DW_FO_CUSTOMERS
    ADD PRIMARY KEY (CUSTOMER_CODE)
```

Precaution Dimension View

```
DROP TABLE DW_FO_PRECAUTION

CREATE TABLE DW_FO_PRECAUTION (
    PRECAUTION_CODE     varchar(5) NOT NULL,
    PRECAUTION_DESCRIPTION varchar(150) NOT NULL
)

ALTER TABLE DW_FO_PRECAUTION
    ADD PRIMARY KEY (PRECAUTION_CODE)
```

Payment Type Dimension View

```
DROP TABLE DW_FO_PAYMENTTYPE

CREATE TABLE DW_FO_PAYMENTTYPE (
    PAYMENT_TYPE_CODE   int NOT NULL,
    PAY_TYPE_DESCRIPTION varchar(50) NOT NULL
)

ALTER TABLE DW_FO_PAYMENTTYPE
    ADD PRIMARY KEY (PAYMENT_TYPE_CODE)
```

Promotion Dimension View

```
DROP TABLE DW_FO_PROMOTIONS

CREATE TABLE DW_FO_PROMOTIONS (
```

```
    PROMOTION_CODE       int NOT NULL,
    PROMOTION_DESCRIPTION varchar(100) NOT NULL,
    COST                 float NOT NULL
)

ALTER TABLE DW_FO_PROMOTIONS
    ADD PRIMARY KEY (PROMOTION_CODE)
```

Employee Dimension View

```
DROP TABLE DW_FO_EMPLOYEES

CREATE TABLE DW_FO_EMPLOYEES (
    EMPLOYEE_CODE        int NOT NULL,
    EMPLOYEE_DESCRIPTION varchar(50) NOT NULL,
    LEVEL_CODE           varchar(10) NULL,
    PARENT_LEVEL_CODE    varchar(10) NULL
)

ALTER TABLE DW_FO_EMPLOYEES
    ADD PRIMARY KEY (EMPLOYEE_CODE)
```

Fact Table View

```
DROP TABLE DW_FACT_TABLE_FO

CREATE TABLE DW_FACT_TABLE_FO (
    VISIT_DAY_CODE       float NOT NULL,
    VISIT_DATE           datetime NOT NULL,
    CUSTOMER_CODE        int NOT NULL,
    EMPLOYEE_CODE        int NOT NULL,
    PRECAUTION_CODE      varchar(5) NOT NULL,
    PAYMENT_TYPE_CODE    int NULL,
    STATUS_CODE          int NULL,
    PROMOTION_CODE       int NULL,
    PRECAUTION_DEADLINE  datetime NULL,
    PRECAUTION_COMPLETION_DATE datetime NULL,
    PRECAUTION_PRIORITY  float NULL,
    PRECAUTION_POINT     float NULL,
    DURATION             float NULL,
    NUMBER_OF_VISITS     float NULL,
    ENDORSEMENT          float NULL
)

ALTER TABLE DW_FACT_TABLE_FO
    ADD PRIMARY KEY (VISIT_DAY_CODE, VISIT_DATE, CUSTOMER_CODE,
        EMPLOYEE_CODE, PRECAUTION_CODE)
```

APPENDIX C

VIEWS

```
CREATE OR REPLACE PROCEDURE FILL_TABLES AS

        sSQL                VARCHAR2(10000);
        sTableName          VARCHAR2(30);
        iMultiplier         NUMBER;
        iPlanCode           TBLPPCHHD_ROUTE_PLAN.PLAN_CODE%TYPE;
        iCustCode           TBLPPCHHD_ROUTE_PLAN_CUS.CUSTOMERCODE%TYPE;
        iChiefCode          TBLPPCHHD_ROUTE_PLAN.ROUTE_CHIEF%TYPE;

        sErrDesc            VARCHAR2(255);
        sErrNo              NUMBER;
        i                   NUMBER;
        dDate               TBLPPCHHD_ROUTE_PLAN.PLAN_DATE%TYPE;

        TYPE GENERIC_CURSOR     IS REF CURSOR;
        curPlans            GENERIC_CURSOR;
        curTables           GENERIC_CURSOR;
        curNumberOfs        GENERIC_CURSOR;
        curPlanChiefs       GENERIC_CURSOR;
BEGIN

    i := 0;
    --Hali haz¿rda bulunan tablolar¿ silelim
    sSQL := 'SELECT TABLE_NAME FROM USER_TABLES WHERE TABLE_NAME IN (';
    sSQL := sSQL || '       "DW_FACT_TABLE_FO"';
    sSQL := sSQL || '     , "DW_FO_EMPLOYEES"';
    sSQL := sSQL || '     , "DW_FO_CUSTOMERS"';
    sSQL := sSQL || '     , "DW_FO_PAYMENTTYPE"';
    sSQL := sSQL || '     , "DW_FO_PREC_STATUS"';
    sSQL := sSQL || '     , "DW_FO_PRECAUTION"';
    sSQL := sSQL || '     , "DW_FO_PROMOTIONS"';
    sSQL := sSQL || ')';
    OPEN curTables FOR sSQL;
    LOOP
        FETCH curTables INTO sTableName;
        EXIT WHEN curTables - NVL(OP_VISITED_CUSTOMERS.START_TIME, TO_DATE("01/01/1900",
"DD/MM/YYYY")) AS DURATION, ';
        sSQL := sSQL || '   NVL(OP_VISITED_CUSTOMERS.PROMOTION,         TO_DATE("01/01/1900",
"DD/MM/YYYY")) AS PROMOTION_CODE, ';
        sSQL := sSQL || '   OP_PAYMENTTYPE.PAYMENT_TYPE AS PAYMENT_TYPE_CODE, ';
        sSQL := sSQL || '   OP_VISITED_CUSTOMERS.ENDORSEMENT ENDORSEMENT, ';
        sSQL := sSQL || '   NVL(OP_VISITED_CUSTOMERS.PRECAUTION,        TO_DATE("01/01/1900",
"DD/MM/YYYY")) AS PRECAUTION_CODE, ';
        sSQL := sSQL || '   OP_PRECAUTION_CONTROL.STATUS, AS PRECAUTION_STATUS, ';
        sSQL := sSQL || '   OP_PRECAUTION_CONTROL.PRIORITY AS PRECAUTION_PRIORITY, ';
        sSQL := sSQL || '   OP_PRECAUTION_CONTROL.DEADLINE AS PRECAUTION_DEADLINE, ';
        sSQL := sSQL || '   OP_PRECAUTION_CONTROL.COMPLETION_DATE                    AS
PRECAUTION_COMPLETION_DATE, ';
        sSQL := sSQL || '   TABLE_POINT.WEIGHT AS PRECAUTION_POINT, ';
        sSQL := sSQL || '   0 AS NUMBER_OF_VISITS ';
        sSQL := sSQL || ' /*Tables*/ ';
        sSQL := sSQL || ' FROM      TABLE_CUSTOMERS, ';
        sSQL := sSQL || '   TABLE_EMPLOYEE, ';
        sSQL := sSQL || '   TABLE_PAYMENT_TYPES, ';
        sSQL := sSQL || '   TABLE_PRECAUTIONS, ';
```

```
        sSQL := sSQL || '   TABLE_PROMOTIONS,  ';
        sSQL := sSQL || '   TABLE_STATUS,  ';
        sSQL := sSQL || '   OP_PAYMENTTYPE,  ';
        sSQL := sSQL || '   OP_PRECAUTION_CONTROL,  ';
        sSQL := sSQL || '   TABLE_POINT,  ';
        sSQL := sSQL || '   OP_VEHICLE_CONTROL,  ';
        sSQL := sSQL || '   OP_VISITS,  ';
        sSQL := sSQL || '   OP_VISITED_CUSTOMERS ';
        sSQL := sSQL || ' /*Join Operations*/ ';
        sSQL := sSQL || ' WHERE 1 = 1 ';
        sSQL := sSQL || '   AND OP_VISITED_CUSTOMERS.VISIT_CODE          =
OP_VISITS.VISIT_CODE ';
        sSQL := sSQL || '   AND TABLE_CUSTOMERS.CODE              = TABLE_EMPLOYEE.CODE  ';
        sSQL := sSQL || '   AND TABLE_CUSTOMERS.CODE              =
TABLE_PAYMENT_TYPES.CODE  ';
        sSQL := sSQL || '   AND TABLE_CUSTOMERS.CODE              = TABLE_PROMOTIONS.CODE  ';
        sSQL := sSQL || '   AND TABLE_CUSTOMERS.CODE              = TABLE_STATUS.CODE  ';
        sSQL := sSQL || '   AND TABLE_CUSTOMERS.CODE              =
OP_PAYMENTTYPE.CUSTOMER  ';
        sSQL := sSQL || '   AND TABLE_CUSTOMERS.CODE              =
OP_VISITED_CUSTOMERS.CUSTOMER  ';
        sSQL := sSQL || '   AND TABLE_CUSTOMERS.CODE              =
OP_PRECAUTION_CONTROL.CUSTOMER  ';
        sSQL := sSQL || '   AND TABLE_PAYMENT_TYPES.CODE                  =
OP_PAYMENTTYPE.PAYMENT_TYPE  ';
        sSQL := sSQL || '   AND OP_PAYMENTTYPE.VISIT_CODE                 =
OP_VISITS.VISIT_CODE  ';
        sSQL := sSQL || '   AND TABLE_STATUS.CODE                         =
OP_PRECAUTION_CONTROL.STATUS  ';
        sSQL := sSQL || '   AND TABLE_PRECAUTIONS.CODE                    =
OP_PRECAUTION_CONTROL.PRECAUTION  ';
        sSQL := sSQL || '   AND OP_PRECAUTION_CONTROL.PRIORITY          =
TABLE_POINT.PRIORITY  ';
        sSQL := sSQL || '   AND OP_PRECAUTION_CONTROL.VISIT_CODE       =
OP_VISITS.VISIT_CODE  ';
        sSQL := sSQL || '   AND TABLE_PROMOTIONS.CODE             =
OP_VISITED_CUSTOMERS.PROMOTION  ';
        sSQL := sSQL || ' /*Criteria*/ ';
        sSQL := sSQL || '   AND   (OP_VISITED_CUSTOMERS.START_TIME   IS   NOT   NULL   AND
OP_VISITED_CUSTOMERS.FINISH_TIME IS NOT NULL) ';
        sSQL := sSQL || '   AND (OP_VISITS.START_TIME IS NOT NULL AND OP_VISITS.FINISH_TIME
IS NOT NULL) ';


        --CUSTOMERS
        sSQL := '';
        sSQL := sSQL || ' CREATE TABLE DW_FO_CUSTOMERS AS ';
        sSQL  :=  sSQL  ||  '  SELECT  CODE  AS  CUSTOMER_CODE,  DESCRIPTION  AS  NAME,  TYPE  AS
CUSTOMER_TYPE, ';
        sSQL  :=  sSQL  ||  '  DESCRIPTION  AS  TYPE_DESCRIPTION,  CITY  AS  CUSTOMER_CITY_CODE,
DESCRIPTION AS CUSTOMER_CITY_DESCRIPTION, REGION AS DISTRICT ';
        sSQL  :=  sSQL  ||  '  FROM  TABLE_CUSTOMERS  CUST,  TABLE_CUSTOMER_TYPES  CUS_TY,
TABLE_CITIES CTY ';
        sSQL := sSQL || ' WHERE CUS_TY.CODE = CUST.TYPE ';
        sSQL := sSQL || ' AND CTY.CODE =  CUST.CUSTOMER_CITY_CODE ';
        EXECUTE IMMEDIATE sSQL;

        --EMPLOYEES
        sSQL := '';
        sSQL := sSQL || ' CREATE TABLE DW_FO_EMPLOYEES AS  ';
        sSQL  :=  sSQL  ||  '  SELECT  CODE  AS  EMPLOYEE_CODE,  DESCRIPTION  AS
EMPLOYEE_DESCRIPTION, LEVEL_CODE, PARENT_LEVEL_CODE ';
        sSQL := sSQL || ' FROM TABLE_EMPLOYEES EMP, TABLE_EMPLOYEE_HIERARCHY HY ';
        sSQL := sSQL || ' WHERE EMP.CODE = HY.EMPLOYEE ';
        EXECUTE IMMEDIATE sSQL;

        --PROMOTIONS
```

```
	sSQL := '';
	sSQL := sSQL || ' CREATE TABLE DW_FO_PROMOTIONS AS ';
	sSQL := sSQL || ' SELECT CODE AS PROMOTION_CODE, DESCRIPTION AS
PROMOTION_DESCRIPTION, COST ';
	sSQL := sSQL || ' FROM TABLE_PROMOTIONS ';
	EXECUTE IMMEDIATE sSQL;


	--PROMOTIONS
	sSQL := '';
	sSQL := sSQL || ' CREATE TABLE DW_FO_PAYMENTTYPE AS ';
	sSQL := sSQL || ' SELECT CODE AS PAYMENT_TYPE_CODE, DESCRIPTION AS
PAY_TYPE_DESCRIPTION ';
	sSQL := sSQL || ' FROM TABLE_PAYMENT_TYPES ';
	EXECUTE IMMEDIATE sSQL;


	--PAYMENTTYPES
	sSQL := '';
	sSQL := sSQL || ' CREATE TABLE DW_FO_PAYMENTTYPE AS ';
	sSQL := sSQL || ' SELECT CODE AS PAYMENT_TYPE_CODE, DESCRIPTION AS
PAY_TYPE_DESCRIPTION ';
	sSQL := sSQL || ' FROM TABLE_PAYMENT_TYPES ';
	EXECUTE IMMEDIATE sSQL;


	--PRECUATION STATUS
	sSQL := '';
	sSQL := sSQL || ' CREATE TABLE DW_FO_PREC_STATUS AS ';
	sSQL := sSQL || ' SELECT -1 AS STATUS_CODE, "<TAMAMLANDI>" AS STATUS_DESCRIPTION
FROM DUAL ';
	sSQL := sSQL || ' UNION ALL ';
	sSQL := sSQL || ' SELECT 0 AS STATUS_CODE, "<TAMAMLANMADI>" AS
STATUS_DESCRIPTION FROM DUAL ';
	EXECUTE IMMEDIATE sSQL;


	--PRECUATIONS
	sSQL := '';
	sSQL := sSQL || ' CREATE TABLE DW_FO_PRECAUTION AS ';
	sSQL := sSQL || ' SELECT CODE AS PRECAUTION_CODE, DESCRIPTION AS
PRECAUTION_DESCRIPTION ';
	sSQL := sSQL || ' FROM TABLE_PRECAUTIONS ';
	EXECUTE IMMEDIATE sSQL;
END;/
```

APPENDIX D

DATA TRANSFORMATION PACKAGE

```
'**********************************************************************
'Microsoft SQL Server 2000
'Visual Basic file generated for DTS Package
'Package Name: TEZ_AKTARIMI
'**********************************************************************
Option Explicit
Public goPackageOld As New DTS.Package
Public goPackage As DTS.Package2
Private Sub Main()
        set goPackage = goPackageOld

        goPackage.Name = "ISIL_TEZ_AKTARIMI2"
        goPackage.WriteCompletionStatusToNTEventLog = False
        goPackage.FailOnError = False
        goPackage.PackagePriorityClass = 2
        goPackage.MaxConcurrentSteps = 4
        goPackage.LineageOptions = 0
        goPackage.UseTransaction = True
        goPackage.TransactionIsolationLevel = 4096
        goPackage.AutoCommitTransaction = True
        goPackage.RepositoryMetadataOptions = 0
        goPackage.UseOLEDBServiceComponents = True
        goPackage.LogToSQLServer = False
        goPackage.LogServerFlags = 0
        goPackage.FailPackageOnLogFailure = False
        goPackage.ExplicitGlobalVariables = False
        goPackage.PackageType = 0

'-------------------------------------------------------------------------
' create package connection information
'-------------------------------------------------------------------------

Dim oConnection as DTS.Connection2

'------------ a new connection defined below.
'For security purposes, the password is never scripted

Set oConnection = goPackage.Connections.New("SQLOLEDB")

        oConnection.ConnectionProperties("Persist Security Info") = True
        oConnection.ConnectionProperties("User ID") = "SA"
        oConnection.ConnectionProperties("Initial Catalog") = "ISIL_TEZ"
        oConnection.ConnectionProperties("Data Source") = "(local)"
        oConnection.ConnectionProperties("Application Name") = "DTS Designer"

        oConnection.Name = "SQL Server"
        oConnection.ID = 2
        oConnection.Reusable = True
        oConnection.ConnectImmediate = False
        oConnection.DataSource = "(local)"
        oConnection.UserID = "SA"
        oConnection.ConnectionTimeout = 60
        oConnection.Catalog = "ISIL_TEZ"
        oConnection.UseTrustedConnection = False
        oConnection.UseDSL = False
```

```
        'If you have a password for this connection, please uncomment and add your password below.
        'oConnection.Password = "<put the password here>"

goPackage.Connections.Add oConnection
Set oConnection = Nothing

'------------ a new connection defined below.
'For security purposes, the password is never scripted

Set oConnection = goPackage.Connections.New("MSDASQL")

        oConnection.ConnectionProperties("Persist Security Info") = True
        oConnection.ConnectionProperties("User ID") = "SA"
        oConnection.ConnectionProperties("Data Source") = "INASE"

        oConnection.Name = "ORACLE"
        oConnection.ID = 1
        oConnection.Reusable = True
        oConnection.ConnectImmediate = False
        oConnection.DataSource = "INASE"
        oConnection.UserID = "SA"
        oConnection.ConnectionTimeout = 60
        oConnection.UseTrustedConnection = False
        oConnection.UseDSL = False

        'If you have a password for this connection, please uncomment and add your password below.
        'oConnection.Password = "<put the password here>"

goPackage.Connections.Add oConnection
Set oConnection = Nothing

'-----------------------------------------------------------------------
' create package steps information
'-----------------------------------------------------------------------

Dim oStep as DTS.Step2
Dim oPrecConstraint as DTS.PrecedenceConstraint

'------------ a new step defined below

Set oStep = goPackage.Steps.New

        oStep.Name = "DTSStep_DTSDataPumpTask_1"
        oStep.Description = "FACT"
        oStep.ExecutionStatus = 1
        oStep.TaskName = "DTSTask_DTSDataPumpTask_1"
        oStep.CommitSuccess = False
        oStep.RollbackFailure = False
        oStep.ScriptLanguage = "VBScript"
        oStep.AddGlobalVariables = True
        oStep.RelativePriority = 3
        oStep.CloseConnection = False
        oStep.ExecuteInMainThread = False
        oStep.IsPackageDSORowset = False
        oStep.JoinTransactionIfPresent = False
        oStep.DisableStep = False
        oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below

Set oStep = goPackage.Steps.New

        oStep.Name = "DTSStep_DTSDataPumpTask_2"
        oStep.Description = "ACTIONS"
```

```
		oStep.ExecutionStatus = 1
		oStep.TaskName = "DTSTask_DTSDataPumpTask_2"
		oStep.CommitSuccess = False
		oStep.RollbackFailure = False
		oStep.ScriptLanguage = "VBScript"
		oStep.AddGlobalVariables = True
		oStep.RelativePriority = 3
		oStep.CloseConnection = False
		oStep.ExecuteInMainThread = False
		oStep.IsPackageDSORowset = False
		oStep.JoinTransactionIfPresent = False
		oStep.DisableStep = False
		oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below

Set oStep = goPackage.Steps.New

		oStep.Name = "DTSStep_DTSDataPumpTask_3"
		oStep.Description = "CUSTOMERS"
		oStep.ExecutionStatus = 1
		oStep.TaskName = "DTSTask_DTSDataPumpTask_3"
		oStep.CommitSuccess = False
		oStep.RollbackFailure = False
		oStep.ScriptLanguage = "VBScript"
		oStep.AddGlobalVariables = True
		oStep.RelativePriority = 3
		oStep.CloseConnection = False
		oStep.ExecuteInMainThread = False
		oStep.IsPackageDSORowset = False
		oStep.JoinTransactionIfPresent = False
		oStep.DisableStep = False
		oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below

Set oStep = goPackage.Steps.New

		oStep.Name = "DTSStep_DTSDataPumpTask_4"
		oStep.Description = "PAYTYPE"
		oStep.ExecutionStatus = 1
		oStep.TaskName = "DTSTask_DTSDataPumpTask_4"
		oStep.CommitSuccess = False
		oStep.RollbackFailure = False
		oStep.ScriptLanguage = "VBScript"
		oStep.AddGlobalVariables = True
		oStep.RelativePriority = 3
		oStep.CloseConnection = False
		oStep.ExecuteInMainThread = False
		oStep.IsPackageDSORowset = False
		oStep.JoinTransactionIfPresent = False
		oStep.DisableStep = False
		oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below

Set oStep = goPackage.Steps.New
```

```
		oStep.Name = "DTSStep_DTSDataPumpTask_5"
		oStep.Description = "PROMOTIONS"
		oStep.ExecutionStatus = 1
		oStep.TaskName = "DTSTask_DTSDataPumpTask_5"
		oStep.CommitSuccess = False
		oStep.RollbackFailure = False
		oStep.ScriptLanguage = "VBScript"
		oStep.AddGlobalVariables = True
		oStep.RelativePriority = 3
		oStep.CloseConnection = False
		oStep.ExecuteInMainThread = False
		oStep.IsPackageDSORowset = False
		oStep.JoinTransactionIfPresent = False
		oStep.DisableStep = False
		oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below

Set oStep = goPackage.Steps.New

		oStep.Name = "DTSStep_DTSDataPumpTask_6"
		oStep.Description = "STATUS"
		oStep.ExecutionStatus = 1
		oStep.TaskName = "DTSTask_DTSDataPumpTask_6"
		oStep.CommitSuccess = False
		oStep.RollbackFailure = False
		oStep.ScriptLanguage = "VBScript"
		oStep.AddGlobalVariables = True
		oStep.RelativePriority = 3
		oStep.CloseConnection = False
		oStep.ExecuteInMainThread = False
		oStep.IsPackageDSORowset = False
		oStep.JoinTransactionIfPresent = False
		oStep.DisableStep = False
		oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below

Set oStep = goPackage.Steps.New

		oStep.Name = "DTSStep_DTSDataPumpTask_7"
		oStep.Description = "WORKERS"
		oStep.ExecutionStatus = 1
		oStep.TaskName = "DTSTask_DTSDataPumpTask_7"
		oStep.CommitSuccess = False
		oStep.RollbackFailure = False
		oStep.ScriptLanguage = "VBScript"
		oStep.AddGlobalVariables = True
		oStep.RelativePriority = 3
		oStep.CloseConnection = False
		oStep.ExecuteInMainThread = False
		oStep.IsPackageDSORowset = False
		oStep.JoinTransactionIfPresent = False
		oStep.DisableStep = False
		oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below
```

```
Set oStep = goPackage.Steps.New

        oStep.Name = "DTSStep_DTSExecuteSQLTask_1"
        oStep.Description = "TABLO_BOSALT"
        oStep.ExecutionStatus = 1
        oStep.TaskName = "DTSTask_DTSExecuteSQLTask_1"
        oStep.CommitSuccess = False
        oStep.RollbackFailure = False
        oStep.ScriptLanguage = "VBScript"
        oStep.AddGlobalVariables = True
        oStep.RelativePriority = 3
        oStep.CloseConnection = False
        oStep.ExecuteInMainThread = False
        oStep.IsPackageDSORowset = False
        oStep.JoinTransactionIfPresent = False
        oStep.DisableStep = False
        oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a new step defined below

Set oStep = goPackage.Steps.New

        oStep.Name = "DTSStep_DTSOlapProcess.Task_1"
        oStep.Description = "CUBE_PROCESS"
        oStep.ExecutionStatus = 1
        oStep.TaskName = "DTSTask_DTSOlapProcess.Task_1"
        oStep.CommitSuccess = False
        oStep.RollbackFailure = False
        oStep.ScriptLanguage = "VBScript"
        oStep.AddGlobalVariables = True
        oStep.RelativePriority = 3
        oStep.CloseConnection = False
        oStep.ExecuteInMainThread = True
        oStep.IsPackageDSORowset = False
        oStep.JoinTransactionIfPresent = False
        oStep.DisableStep = False
        oStep.FailPackageOnError = False

goPackage.Steps.Add oStep
Set oStep = Nothing

'------------ a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSExecuteSQLTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSExecuteSQLTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------ a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_2")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_2"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------ a precedence constraint for steps defined below
```

```
Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_3")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_3"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_4")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_4"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_5")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_5"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_6")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_6"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_7")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_7"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_2")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSExecuteSQLTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSExecuteSQLTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below
```

```
Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_3")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSExecuteSQLTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSExecuteSQLTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_4")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSExecuteSQLTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSExecuteSQLTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_5")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSExecuteSQLTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSExecuteSQLTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_6")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSExecuteSQLTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSExecuteSQLTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSDataPumpTask_7")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSExecuteSQLTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSExecuteSQLTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSOlapProcess.Task_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_1")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_1"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSOlapProcess.Task_1")
```

```
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_2")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_2"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSOlapProcess.Task_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_3")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_3"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSOlapProcess.Task_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_4")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_4"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSOlapProcess.Task_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_5")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_5"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSOlapProcess.Task_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_6")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_6"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'------------- a precedence constraint for steps defined below

Set oStep = goPackage.Steps("DTSStep_DTSOlapProcess.Task_1")
Set oPrecConstraint = oStep.PrecedenceConstraints.New("DTSStep_DTSDataPumpTask_7")
        oPrecConstraint.StepName = "DTSStep_DTSDataPumpTask_7"
        oPrecConstraint.PrecedenceBasis = 1
        oPrecConstraint.Value = 0

oStep.precedenceConstraints.Add oPrecConstraint
Set oPrecConstraint = Nothing

'-----------------------------------------------------------------------
' create package tasks information
'-----------------------------------------------------------------------
```

```
'------------- call Task_Sub1 for task DTSTask_DTSDataPumpTask_1 (FACT)
Call Task_Sub1( goPackage   )

'------------- call Task_Sub2 for task DTSTask_DTSDataPumpTask_2 (ACTIONS)
Call Task_Sub2( goPackage   )

'------------- call Task_Sub3 for task DTSTask_DTSDataPumpTask_3 (CUSTOMERS)
Call Task_Sub3( goPackage   )

'------------- call Task_Sub4 for task DTSTask_DTSDataPumpTask_4 (PAYTYPE)
Call Task_Sub4( goPackage   )

'------------- call Task_Sub5 for task DTSTask_DTSDataPumpTask_5 (PROMOTIONS)
Call Task_Sub5( goPackage   )

'------------- call Task_Sub6 for task DTSTask_DTSDataPumpTask_6 (STATUS)
Call Task_Sub6( goPackage   )

'------------- call Task_Sub7 for task DTSTask_DTSDataPumpTask_7 (WORKERS)
Call Task_Sub7( goPackage   )

'------------- call Task_Sub8 for task DTSTask_DTSExecuteSQLTask_1 (TABLO_BOSALT)
Call Task_Sub8( goPackage   )

'------------- call Task_Sub9 for task DTSTask_DTSOlapProcess.Task_1 (CUBE_PROCESS)
Call Task_Sub9( goPackage   )

'----------------------------------------------------------------------------
' Save or execute package
'----------------------------------------------------------------------------

'goPackage.SaveToSQLServer "(local)", "sa", ""
goPackage.Execute
goPackage.Uninitialize
'to save a package instead of executing it, comment out the executing package line above and uncomment the saving
package line
set goPackage = Nothing

set goPackageOld = Nothing

End Sub


'------------- define Task_Sub1 for task DTSTask_DTSDataPumpTask_1 (FACT)
Public Sub Task_Sub1(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup

Dim oCustomTask1 As DTS.DataPumpTask2
Set oTask = goPackage.Tasks.New("DTSDataPumpTask")
Set oCustomTask1 = oTask.CustomTask

        oCustomTask1.Name = "DTSTask_DTSDataPumpTask_1"
        oCustomTask1.Description = "FACT"
        oCustomTask1.SourceConnectionID = 1
        oCustomTask1.SourceSQLStatement = "SELECT       PLAN_CODE, WORKER_CODE, VISIT_DATE,
CUSTOMER_CODE, " & vbCrLf
        oCustomTask1.SourceSQLStatement = oCustomTask1.SourceSQLStatement & "PAYMENTTYPECODE,
BONUS_CODE, DURATION, ACTION_CODE, ACTION_STATUS, " & vbCrLf
        oCustomTask1.SourceSQLStatement         =         oCustomTask1.SourceSQLStatement         &         "
ACTION_DEADLINE, ACTION_COMPLETION_DATE, " & vbCrLf
        oCustomTask1.SourceSQLStatement = oCustomTask1.SourceSQLStatement & "ACTION_PRIORITY,
NUMBER_OF_VISITS, POINT, CIRO" & vbCrLf
        oCustomTask1.SourceSQLStatement = oCustomTask1.SourceSQLStatement & "FROM       " & vbCrLf
        oCustomTask1.SourceSQLStatement         =         oCustomTask1.SourceSQLStatement         &         "
OLAP_FACT_TABLE_RM " & vbCrLf
```

```
        oCustomTask1.SourceSQLStatement = oCustomTask1.SourceSQLStatement & "WHERE
VISIT_DATE>=DATEADD(WEEK,-1,VISIT_DATE)"
        oCustomTask1.DestinationConnectionID = 2
        oCustomTask1.DestinationObjectName = "[ISIL_TEZ].[dbo].[OLAP_FACT_TABLE_RM]"
        oCustomTask1.ProgressRowCount = 1000
        oCustomTask1.MaximumErrorCount = 0
        oCustomTask1.FetchBufferSize = 1
        oCustomTask1.UseFastLoad = True
        oCustomTask1.InsertCommitSize = 0
        oCustomTask1.ExceptionFileColumnDelimiter = "|"
        oCustomTask1.ExceptionFileRowDelimiter = vbCrLf
        oCustomTask1.AllowIdentityInserts = False
        oCustomTask1.FirstRow = 0
        oCustomTask1.LastRow = 0
        oCustomTask1.FastLoadOptions = 2
        oCustomTask1.ExceptionFileOptions = 1
        oCustomTask1.DataPumpOptions = 0

Call oCustomTask1_Trans_Sub1( oCustomTask1 )
Call oCustomTask1_Trans_Sub2( oCustomTask1 )
Call oCustomTask1_Trans_Sub3( oCustomTask1 )
Call oCustomTask1_Trans_Sub4( oCustomTask1 )
Call oCustomTask1_Trans_Sub5( oCustomTask1 )
Call oCustomTask1_Trans_Sub6( oCustomTask1 )
Call oCustomTask1_Trans_Sub7( oCustomTask1 )
Call oCustomTask1_Trans_Sub8( oCustomTask1 )
Call oCustomTask1_Trans_Sub9( oCustomTask1 )
Call oCustomTask1_Trans_Sub10( oCustomTask1 )
Call oCustomTask1_Trans_Sub11( oCustomTask1 )
Call oCustomTask1_Trans_Sub12( oCustomTask1 )
Call oCustomTask1_Trans_Sub13( oCustomTask1 )
Call oCustomTask1_Trans_Sub14( oCustomTask1 )
Call oCustomTask1_Trans_Sub15( oCustomTask1 )
goPackage.Tasks.Add oTask
Set oCustomTask1 = Nothing
Set oTask = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub1(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__1"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("PLAN_CODE" , 1)
                        oColumn.Name = "PLAN_CODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("PLAN_CODE" , 1)
                        oColumn.Name = "PLAN_CODE"
```

```
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 5
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub2(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
            oTransformation.Name = "DTSTransformation__2"
            oTransformation.TransformFlags = 63
            oTransformation.ForceSourceBlobsBuffered = 0
            oTransformation.ForceBlobsInMemory = False
            oTransformation.InMemoryBlobSize = 1048576
            oTransformation.TransformPhases = 4

            Set oColumn = oTransformation.SourceColumns.New("WORKER_CODE" , 1)
                oColumn.Name = "WORKER_CODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 3
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("WORKERSCODE" , 1)
                oColumn.Name = "WORKERSCODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 3
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing
```

```
End Sub

Public Sub oCustomTask1_Trans_Sub3(ByVal oCustomTask1 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__3"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("CUSTOMER_CODE" , 1)
			oColumn.Name = "CUSTOMER_CODE"
			oColumn.Ordinal = 1
			oColumn.Flags = 24
			oColumn.Size = 0
			oColumn.DataType = 3
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("CUSTOMERCODE" , 1)
			oColumn.Name = "CUSTOMERCODE"
			oColumn.Ordinal = 1
			oColumn.Flags = 24
			oColumn.Size = 0
			oColumn.DataType = 3
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask1.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub4(ByVal oCustomTask1 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__4"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("BONUS_CODE" , 1)
			oColumn.Name = "BONUS_CODE"
```

```
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 5
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("BONUS_CODE" , 1)
                oColumn.Name = "BONUS_CODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 5
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub5(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
            oTransformation.Name = "DTSTransformation__5"
            oTransformation.TransformFlags = 63
            oTransformation.ForceSourceBlobsBuffered = 0
            oTransformation.ForceBlobsInMemory = False
            oTransformation.InMemoryBlobSize = 1048576
            oTransformation.TransformPhases = 4

            Set oColumn = oTransformation.SourceColumns.New("ACTION_CODE" , 1)
                oColumn.Name = "ACTION_CODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 5
                oColumn.DataType = 129
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("ACTION_CODE" , 1)
                oColumn.Name = "ACTION_CODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 5
                oColumn.DataType = 129
                oColumn.Precision = 0
```

```
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub6(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
            oTransformation.Name = "DTSTransformation__6"
            oTransformation.TransformFlags = 63
            oTransformation.ForceSourceBlobsBuffered = 0
            oTransformation.ForceBlobsInMemory = False
            oTransformation.InMemoryBlobSize = 1048576
            oTransformation.TransformPhases = 4

            Set oColumn = oTransformation.SourceColumns.New("VISIT_DATE" , 1)
                oColumn.Name = "VISIT_DATE"
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 135
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("VISIT_DATE" , 1)
                oColumn.Name = "VISIT_DATE"
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 135
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub7(ByVal oCustomTask1 As Object)
```

```
        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__7"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("ACTION_STATUS" , 1)
                        oColumn.Name = "ACTION_STATUS"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("STATUS_CODE" , 1)
                        oColumn.Name = "STATUS_CODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.DestinationColumns.Add oColumn
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub8(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__8"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("PAYMENTTYPECODE" , 1)
                        oColumn.Name = "PAYMENTTYPECODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 120
                        oColumn.Size = 0
                        oColumn.DataType = 4
                        oColumn.Precision = 0
```

```
                oColumn.NumericScale = 0
                oColumn.Nullable = True

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("PAYMENTTYPECODE" , 1)
                oColumn.Name = "PAYMENTTYPECODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 120
                oColumn.Size = 0
                oColumn.DataType = 4
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = True

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub9(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
            oTransformation.Name = "DTSTransformation__9"
            oTransformation.TransformFlags = 63
            oTransformation.ForceSourceBlobsBuffered = 0
            oTransformation.ForceBlobsInMemory = False
            oTransformation.InMemoryBlobSize = 1048576
            oTransformation.TransformPhases = 4

            Set oColumn = oTransformation.SourceColumns.New("DURATION" , 1)
                oColumn.Name = "DURATION"
                oColumn.Ordinal = 1
                oColumn.Flags = 120
                oColumn.Size = 0
                oColumn.DataType = 5
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = True

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("DURATION" , 1)
                oColumn.Name = "DURATION"
                oColumn.Ordinal = 1
                oColumn.Flags = 120
                oColumn.Size = 0
                oColumn.DataType = 5
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = True

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing
```

```
        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub10(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__10"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("ACTION_DEADLINE" , 1)
                        oColumn.Name = "ACTION_DEADLINE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 120
                        oColumn.Size = 0
                        oColumn.DataType = 135
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = True

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("ACTION_DEADLINE" , 1)
                        oColumn.Name = "ACTION_DEADLINE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 120
                        oColumn.Size = 0
                        oColumn.DataType = 135
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = True

                oTransformation.DestinationColumns.Add oColumn
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask1.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub11(ByVal oCustomTask1 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__11"
```

```
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("ACTION_PRIORITY" , 1)
			oColumn.Name = "ACTION_PRIORITY"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("ACTION_PRIORITY" , 1)
			oColumn.Name = "ACTION_PRIORITY"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask1.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub12(ByVal oCustomTask1 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__12"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("ACTION_COMPLETION_DATE" , 1)
			oColumn.Name = "ACTION_COMPLETION_DATE"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 135
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing
```

```
		Set oColumn = oTransformation.DestinationColumns.New("ACTION_COMPLETION_DATE" ,
1)
			oColumn.Name = "ACTION_COMPLETION_DATE"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 131
			oColumn.Precision = 18
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask1.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub13(ByVal oCustomTask1 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__13"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("NUMBER_OF_VISITS" , 1)
			oColumn.Name = "NUMBER_OF_VISITS"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("NUMBER_OF_VISITS" , 1)
			oColumn.Name = "NUMBER_OF_VISITS"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties
```

```
	Set oTransProps = Nothing

	oCustomTask1.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub14(ByVal oCustomTask1 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__14"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("POINT" , 1)
			oColumn.Name = "POINT"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("POINT" , 1)
			oColumn.Name = "POINT"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask1.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask1_Trans_Sub15(ByVal oCustomTask1 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask1.Transformations.New("DTSPump.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__15"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
```

```
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("CIRO" , 1)
			oColumn.Name = "CIRO"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("ENDORSEMENT" , 1)
			oColumn.Name = "ENDORSEMENT"
			oColumn.Ordinal = 1
			oColumn.Flags = 120
			oColumn.Size = 0
			oColumn.DataType = 5
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask1.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

'------------- define Task_Sub2 for task DTSTask_DTSDataPumpTask_2 (ACTIONS)
Public Sub Task_Sub2(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup

Dim oCustomTask2 As DTS.DataPumpTask2
Set oTask = goPackage.Tasks.New("DTSDataPumpTask")
Set oCustomTask2 = oTask.CustomTask

	oCustomTask2.Name = "DTSTask_DTSDataPumpTask_2"
	oCustomTask2.Description = "ACTIONS"
	oCustomTask2.SourceConnectionID = 1
	oCustomTask2.SourceObjectName = "[INASE].[dbo].[OLAP_RM_ACTIONS]"
	oCustomTask2.DestinationConnectionID = 2
	oCustomTask2.DestinationObjectName = "[ISIL_TEZ].[dbo].[OLAP_RM_ACTIONS]"
	oCustomTask2.ProgressRowCount = 1000
	oCustomTask2.MaximumErrorCount = 0
	oCustomTask2.FetchBufferSize = 1
	oCustomTask2.UseFastLoad = True
	oCustomTask2.InsertCommitSize = 0
	oCustomTask2.ExceptionFileColumnDelimiter = "|"
	oCustomTask2.ExceptionFileRowDelimiter = vbCrLf
	oCustomTask2.AllowIdentityInserts = False
	oCustomTask2.FirstRow = 0
	oCustomTask2.LastRow = 0
	oCustomTask2.FastLoadOptions = 2
	oCustomTask2.ExceptionFileOptions = 1
	oCustomTask2.DataPumpOptions = 0
```

```
Call oCustomTask2_Trans_Sub1( oCustomTask2)
Call oCustomTask2_Trans_Sub2( oCustomTask2)
goPackage.Tasks.Add oTask
Set oCustomTask2 = Nothing
Set oTask = Nothing

End Sub

Public Sub oCustomTask2_Trans_Sub1(ByVal oCustomTask2 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask2.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__1"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("ACTION_CODE" , 1)
			oColumn.Name = "ACTION_CODE"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 5
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("ACTION_CODE" , 1)
			oColumn.Name = "ACTION_CODE"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 5
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask2.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask2_Trans_Sub2(ByVal oCustomTask2 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask2.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__2"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
```

```
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("ACTION_DESC" , 1)
			oColumn.Name = "ACTION_DESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 150
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("ACTION_DESC" , 1)
			oColumn.Name = "ACTION_DESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 150
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask2.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

'------------- define Task_Sub3 for task DTSTask_DTSDataPumpTask_3 (CUSTOMERS)
Public Sub Task_Sub3(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup

Dim oCustomTask3 As DTS.DataPumpTask2
Set oTask = goPackage.Tasks.New("DTSDataPumpTask")
Set oCustomTask3 = oTask.CustomTask

	oCustomTask3.Name = "DTSTask_DTSDataPumpTask_3"
	oCustomTask3.Description = "CUSTOMERS"
	oCustomTask3.SourceConnectionID = 1
	oCustomTask3.SourceObjectName = "[INASE].[dbo].[OLAP_RM_CUSTOMERS]"
	oCustomTask3.DestinationConnectionID = 2
	oCustomTask3.DestinationObjectName = "[ISIL_TEZ].[dbo].[OLAP_RM_CUSTOMERS]"
	oCustomTask3.ProgressRowCount = 1000
	oCustomTask3.MaximumErrorCount = 0
	oCustomTask3.FetchBufferSize = 1
	oCustomTask3.UseFastLoad = True
	oCustomTask3.InsertCommitSize = 0
	oCustomTask3.ExceptionFileColumnDelimiter = "|"
	oCustomTask3.ExceptionFileRowDelimiter = vbCrLf
	oCustomTask3.AllowIdentityInserts = False
	oCustomTask3.FirstRow = 0
	oCustomTask3.LastRow = 0
	oCustomTask3.FastLoadOptions = 2
```

```
        oCustomTask3.ExceptionFileOptions = 1
        oCustomTask3.DataPumpOptions = 0

Call oCustomTask3_Trans_Sub1( oCustomTask3)
Call oCustomTask3_Trans_Sub2( oCustomTask3)
Call oCustomTask3_Trans_Sub3( oCustomTask3)
Call oCustomTask3_Trans_Sub4( oCustomTask3)
Call oCustomTask3_Trans_Sub5( oCustomTask3)
Call oCustomTask3_Trans_Sub6( oCustomTask3)
Call oCustomTask3_Trans_Sub7( oCustomTask3)
Call oCustomTask3_Trans_Sub8( oCustomTask3)
goPackage.Tasks.Add oTask
Set oCustomTask3 = Nothing
Set oTask = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub1(ByVal oCustomTask3 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__1"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("CUSTOMERCODE" , 1)
                        oColumn.Name = "CUSTOMERCODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 3
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("CUSTOMERCODE" , 1)
                        oColumn.Name = "CUSTOMERCODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 3
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.DestinationColumns.Add oColumn
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask3.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub2(ByVal oCustomTask3 As Object)
```

```
	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__2"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("CUSTOMERDESC" , 1)
			oColumn.Name = "CUSTOMERDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 50
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("CUSTOMERDESC" , 1)
			oColumn.Name = "CUSTOMERDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 50
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask3.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub3(ByVal oCustomTask3 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__3"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("CUSTOMERTYPECODE" , 1)
			oColumn.Name = "CUSTOMERTYPECODE"
			oColumn.Ordinal = 1
			oColumn.Flags = 24
			oColumn.Size = 0
			oColumn.DataType = 4
```

```
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("CUSTOMERTYPECODE" , 1)
                oColumn.Name = "CUSTOMERTYPECODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 4
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask3.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub4(ByVal oCustomTask3 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
            oTransformation.Name = "DTSTransformation__4"
            oTransformation.TransformFlags = 63
            oTransformation.ForceSourceBlobsBuffered = 0
            oTransformation.ForceBlobsInMemory = False
            oTransformation.InMemoryBlobSize = 1048576
            oTransformation.TransformPhases = 4

            Set oColumn = oTransformation.SourceColumns.New("CUSTOMERTYPEDESC" , 1)
                oColumn.Name = "CUSTOMERTYPEDESC"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 50
                oColumn.DataType = 129
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("CUSTOMERTYPEDESC" , 1)
                oColumn.Name = "CUSTOMERTYPEDESC"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 50
                oColumn.DataType = 129
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
```

```
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask3.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub5(ByVal oCustomTask3 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__5"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("CITYCODE" , 1)
                        oColumn.Name = "CITYCODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 8
                        oColumn.Size = 10
                        oColumn.DataType = 129
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("CITYCODE" , 1)
                        oColumn.Name = "CITYCODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 8
                        oColumn.Size = 10
                        oColumn.DataType = 129
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.DestinationColumns.Add oColumn
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask3.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub6(ByVal oCustomTask3 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
```

```
		oTransformation.Name = "DTSTransformation__6"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("CITYDESC" , 1)
			oColumn.Name = "CITYDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 50
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("CITYDESC" , 1)
			oColumn.Name = "CITYDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 50
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask3.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub7(ByVal oCustomTask3 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__7"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("CUSTOMERDISTRICT" , 1)
			oColumn.Name = "CUSTOMERDISTRICT"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 20
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
```

```
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("CUSTOMERDISTRICT" , 1)
			oColumn.Name = "CUSTOMERDISTRICT"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 20
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask3.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask3_Trans_Sub8(ByVal oCustomTask3 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask3.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__8"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("VOLUMESHORTDESC" , 1)
			oColumn.Name = "VOLUMESHORTDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 5
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("VOLUMESHORTDESC" , 1)
			oColumn.Name = "VOLUMESHORTDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 5
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties
```

```
        Set oTransProps = Nothing

        oCustomTask3.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

'------------- define Task_Sub4 for task DTSTask_DTSDataPumpTask_4 (PAYTYPE)
Public Sub Task_Sub4(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup

Dim oCustomTask4 As DTS.DataPumpTask2
Set oTask = goPackage.Tasks.New("DTSDataPumpTask")
Set oCustomTask4 = oTask.CustomTask

        oCustomTask4.Name = "DTSTask_DTSDataPumpTask_4"
        oCustomTask4.Description = "PAYTYPE"
        oCustomTask4.SourceConnectionID = 1
        oCustomTask4.SourceObjectName = "[INASE].[dbo].[OLAP_RM_PAYMENTTYPE]"
        oCustomTask4.DestinationConnectionID = 2
        oCustomTask4.DestinationObjectName = "[ISIL_TEZ].[dbo].[OLAP_RM_PAYMENTTYPE]"
        oCustomTask4.ProgressRowCount = 1000
        oCustomTask4.MaximumErrorCount = 0
        oCustomTask4.FetchBufferSize = 1
        oCustomTask4.UseFastLoad = True
        oCustomTask4.InsertCommitSize = 0
        oCustomTask4.ExceptionFileColumnDelimiter = "|"
        oCustomTask4.ExceptionFileRowDelimiter = vbCrLf
        oCustomTask4.AllowIdentityInserts = False
        oCustomTask4.FirstRow = 0
        oCustomTask4.LastRow = 0
        oCustomTask4.FastLoadOptions = 2
        oCustomTask4.ExceptionFileOptions = 1
        oCustomTask4.DataPumpOptions = 0

Call oCustomTask4_Trans_Sub1( oCustomTask4)
Call oCustomTask4_Trans_Sub2( oCustomTask4)
goPackage.Tasks.Add oTask
Set oCustomTask4 = Nothing
Set oTask = Nothing

End Sub

Public Sub oCustomTask4_Trans_Sub1(ByVal oCustomTask4 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask4.Transformations.New("DTS.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__1"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("PAYMENTTYPECODE" , 1)
                        oColumn.Name = "PAYMENTTYPECODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 4
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False
```

```
		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("PAYMENTTYPECODE" , 1)
			oColumn.Name = "PAYMENTTYPECODE"
			oColumn.Ordinal = 1
			oColumn.Flags = 24
			oColumn.Size = 0
			oColumn.DataType = 4
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask4.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask4_Trans_Sub2(ByVal oCustomTask4 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask4.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__2"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("PAYMENTTYPEDESC" , 1)
			oColumn.Name = "PAYMENTTYPEDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 50
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("PAYMENTTYPEDESC" , 1)
			oColumn.Name = "PAYMENTTYPEDESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 8
			oColumn.Size = 50
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = False

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties
```

```
        Set oTransProps = Nothing

        oCustomTask4.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

'------------- define Task_Sub5 for task DTSTask_DTSDataPumpTask_5 (PROMOTIONS)
Public Sub Task_Sub5(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup

Dim oCustomTask5 As DTS.DataPumpTask2
Set oTask = goPackage.Tasks.New("DTSDataPumpTask")
Set oCustomTask5 = oTask.CustomTask

        oCustomTask5.Name = "DTSTask_DTSDataPumpTask_5"
        oCustomTask5.Description = "PROMOTIONS"
        oCustomTask5.SourceConnectionID = 1
        oCustomTask5.SourceObjectName = "[INASE].[dbo].[OLAP_RM_PROMOTIONS]"
        oCustomTask5.DestinationConnectionID = 2
        oCustomTask5.DestinationObjectName = "[ISIL_TEZ].[dbo].[OLAP_RM_PROMOTIONS]"
        oCustomTask5.ProgressRowCount = 1000
        oCustomTask5.MaximumErrorCount = 0
        oCustomTask5.FetchBufferSize = 1
        oCustomTask5.UseFastLoad = True
        oCustomTask5.InsertCommitSize = 0
        oCustomTask5.ExceptionFileColumnDelimiter = "|"
        oCustomTask5.ExceptionFileRowDelimiter = vbCrLf
        oCustomTask5.AllowIdentityInserts = False
        oCustomTask5.FirstRow = 0
        oCustomTask5.LastRow = 0
        oCustomTask5.FastLoadOptions = 2
        oCustomTask5.ExceptionFileOptions = 1
        oCustomTask5.DataPumpOptions = 0

Call oCustomTask5_Trans_Sub1( oCustomTask5)
Call oCustomTask5_Trans_Sub2( oCustomTask5)
Call oCustomTask5_Trans_Sub3( oCustomTask5)
goPackage.Tasks.Add oTask
Set oCustomTask5 = Nothing
Set oTask = Nothing

End Sub

Public Sub oCustomTask5_Trans_Sub1(ByVal oCustomTask5 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask5.Transformations.New("DTS.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__1"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("BONUS_CODE" , 1)
                        oColumn.Name = "BONUS_CODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
```

```
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("BONUS_CODE" , 1)
                oColumn.Name = "BONUS_CODE"
                oColumn.Ordinal = 1
                oColumn.Flags = 24
                oColumn.Size = 0
                oColumn.DataType = 5
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask5.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask5_Trans_Sub2(ByVal oCustomTask5 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask5.Transformations.New("DTS.DataPumpTransformCopy")
            oTransformation.Name = "DTSTransformation__2"
            oTransformation.TransformFlags = 63
            oTransformation.ForceSourceBlobsBuffered = 0
            oTransformation.ForceBlobsInMemory = False
            oTransformation.InMemoryBlobSize = 1048576
            oTransformation.TransformPhases = 4

            Set oColumn = oTransformation.SourceColumns.New("BONUS_NAME" , 1)
                oColumn.Name = "BONUS_NAME"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 100
                oColumn.DataType = 129
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("BONUS_NAME" , 1)
                oColumn.Name = "BONUS_NAME"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 100
                oColumn.DataType = 129
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
```

```
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask5.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask5_Trans_Sub3(ByVal oCustomTask5 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask5.Transformations.New("DTS.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__3"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("COST" , 1)
                        oColumn.Name = "COST"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("COST" , 1)
                        oColumn.Name = "COST"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.DestinationColumns.Add oColumn
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask5.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

'------------- define Task_Sub6 for task DTSTask_DTSDataPumpTask_6 (STATUS)
Public Sub Task_Sub6(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup
```

```
Dim oCustomTask6 As DTS.DataPumpTask2
Set oTask = goPackage.Tasks.New("DTSDataPumpTask")
Set oCustomTask6 = oTask.CustomTask

        oCustomTask6.Name = "DTSTask_DTSDataPumpTask_6"
        oCustomTask6.Description = "STATUS"
        oCustomTask6.SourceConnectionID = 1
        oCustomTask6.SourceObjectName = "[INASE].[dbo].[OLAP_RM_STATUS]"
        oCustomTask6.DestinationConnectionID = 2
        oCustomTask6.DestinationObjectName = "[ISIL_TEZ].[dbo].[OLAP_RM_STATUS]"
        oCustomTask6.ProgressRowCount = 1000
        oCustomTask6.MaximumErrorCount = 0
        oCustomTask6.FetchBufferSize = 1
        oCustomTask6.UseFastLoad = True
        oCustomTask6.InsertCommitSize = 0
        oCustomTask6.ExceptionFileColumnDelimiter = "|"
        oCustomTask6.ExceptionFileRowDelimiter = vbCrLf
        oCustomTask6.AllowIdentityInserts = False
        oCustomTask6.FirstRow = 0
        oCustomTask6.LastRow = 0
        oCustomTask6.FastLoadOptions = 2
        oCustomTask6.ExceptionFileOptions = 1
        oCustomTask6.DataPumpOptions = 0

Call oCustomTask6_Trans_Sub1( oCustomTask6)
Call oCustomTask6_Trans_Sub2( oCustomTask6)
goPackage.Tasks.Add oTask
Set oCustomTask6 = Nothing
Set oTask = Nothing

End Sub

Public Sub oCustomTask6_Trans_Sub1(ByVal oCustomTask6 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask6.Transformations.New("DTS.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__1"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("STATUS_CODE" , 1)
                        oColumn.Name = "STATUS_CODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("STATUS_CODE" , 1)
                        oColumn.Name = "STATUS_CODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 5
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False
```

```
		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask6.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask6_Trans_Sub2(ByVal oCustomTask6 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask6.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__2"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("STATUS_DESC" , 1)
			oColumn.Name = "STATUS_DESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 4000
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("STATUS_DESC" , 1)
			oColumn.Name = "STATUS_DESC"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 100
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask6.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

'------------- define Task_Sub7 for task DTSTask_DTSDataPumpTask_7 (WORKERS)
Public Sub Task_Sub7(ByVal goPackage As Object)

Dim oTask As DTS.Task
```

```
Dim oLookup As DTS.Lookup

Dim oCustomTask7 As DTS.DataPumpTask2
Set oTask = goPackage.Tasks.New("DTSDataPumpTask")
Set oCustomTask7 = oTask.CustomTask

        oCustomTask7.Name = "DTSTask_DTSDataPumpTask_7"
        oCustomTask7.Description = "WORKERS"
        oCustomTask7.SourceConnectionID = 1
        oCustomTask7.SourceObjectName = "[INASE].[dbo].[OLAP_RM_WORKERS_HY]"
        oCustomTask7.DestinationConnectionID = 2
        oCustomTask7.DestinationObjectName = "[ISIL_TEZ].[dbo].[OLAP_RM_WORKERS_HY]"
        oCustomTask7.ProgressRowCount = 1000
        oCustomTask7.MaximumErrorCount = 0
        oCustomTask7.FetchBufferSize = 1
        oCustomTask7.UseFastLoad = True
        oCustomTask7.InsertCommitSize = 0
        oCustomTask7.ExceptionFileColumnDelimiter = "|"
        oCustomTask7.ExceptionFileRowDelimiter = vbCrLf
        oCustomTask7.AllowIdentityInserts = False
        oCustomTask7.FirstRow = 0
        oCustomTask7.LastRow = 0
        oCustomTask7.FastLoadOptions = 2
        oCustomTask7.ExceptionFileOptions = 1
        oCustomTask7.DataPumpOptions = 0

Call oCustomTask7_Trans_Sub1( oCustomTask7)
Call oCustomTask7_Trans_Sub2( oCustomTask7)
Call oCustomTask7_Trans_Sub3( oCustomTask7)
Call oCustomTask7_Trans_Sub4( oCustomTask7)
goPackage.Tasks.Add oTask
Set oCustomTask7 = Nothing
Set oTask = Nothing

End Sub

Public Sub oCustomTask7_Trans_Sub1(ByVal oCustomTask7 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask7.Transformations.New("DTS.DataPumpTransformCopy")
                oTransformation.Name = "DTSTransformation__1"
                oTransformation.TransformFlags = 63
                oTransformation.ForceSourceBlobsBuffered = 0
                oTransformation.ForceBlobsInMemory = False
                oTransformation.InMemoryBlobSize = 1048576
                oTransformation.TransformPhases = 4

                Set oColumn = oTransformation.SourceColumns.New("WORKERSCODE" , 1)
                        oColumn.Name = "WORKERSCODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
                        oColumn.DataType = 3
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = False

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("WORKERSCODE" , 1)
                        oColumn.Name = "WORKERSCODE"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 24
                        oColumn.Size = 0
```

```
                oColumn.DataType = 3
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask7.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

Public Sub oCustomTask7_Trans_Sub2(ByVal oCustomTask7 As Object)

        Dim oTransformation As DTS.Transformation2
        Dim oTransProps as DTS.Properties
        Dim oColumn As DTS.Column
        Set oTransformation = oCustomTask7.Transformations.New("DTS.DataPumpTransformCopy")
            oTransformation.Name = "DTSTransformation__2"
            oTransformation.TransformFlags = 63
            oTransformation.ForceSourceBlobsBuffered = 0
            oTransformation.ForceBlobsInMemory = False
            oTransformation.InMemoryBlobSize = 1048576
            oTransformation.TransformPhases = 4

            Set oColumn = oTransformation.SourceColumns.New("WORKERSDESC" , 1)
                oColumn.Name = "WORKERSDESC"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 50
                oColumn.DataType = 129
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.SourceColumns.Add oColumn
            Set oColumn = Nothing

            Set oColumn = oTransformation.DestinationColumns.New("WORKERSDESC" , 1)
                oColumn.Name = "WORKERSDESC"
                oColumn.Ordinal = 1
                oColumn.Flags = 8
                oColumn.Size = 50
                oColumn.DataType = 129
                oColumn.Precision = 0
                oColumn.NumericScale = 0
                oColumn.Nullable = False

            oTransformation.DestinationColumns.Add oColumn
            Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask7.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub
```

```
Public Sub oCustomTask7_Trans_Sub3(ByVal oCustomTask7 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask7.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__3"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("HIERARCHY_ID" , 1)
			oColumn.Name = "HIERARCHY_ID"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 10
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.SourceColumns.Add oColumn
		Set oColumn = Nothing

		Set oColumn = oTransformation.DestinationColumns.New("HIERARCHY_ID" , 1)
			oColumn.Name = "HIERARCHY_ID"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 10
			oColumn.DataType = 129
			oColumn.Precision = 0
			oColumn.NumericScale = 0
			oColumn.Nullable = True

		oTransformation.DestinationColumns.Add oColumn
		Set oColumn = Nothing

	Set oTransProps = oTransformation.TransformServerProperties


	Set oTransProps = Nothing

	oCustomTask7.Transformations.Add oTransformation
	Set oTransformation = Nothing

End Sub

Public Sub oCustomTask7_Trans_Sub4(ByVal oCustomTask7 As Object)

	Dim oTransformation As DTS.Transformation2
	Dim oTransProps as DTS.Properties
	Dim oColumn As DTS.Column
	Set oTransformation = oCustomTask7.Transformations.New("DTS.DataPumpTransformCopy")
		oTransformation.Name = "DTSTransformation__4"
		oTransformation.TransformFlags = 63
		oTransformation.ForceSourceBlobsBuffered = 0
		oTransformation.ForceBlobsInMemory = False
		oTransformation.InMemoryBlobSize = 1048576
		oTransformation.TransformPhases = 4

		Set oColumn = oTransformation.SourceColumns.New("PARENT_HIERARCHY_ID" , 1)
			oColumn.Name = "PARENT_HIERARCHY_ID"
			oColumn.Ordinal = 1
			oColumn.Flags = 104
			oColumn.Size = 10
```

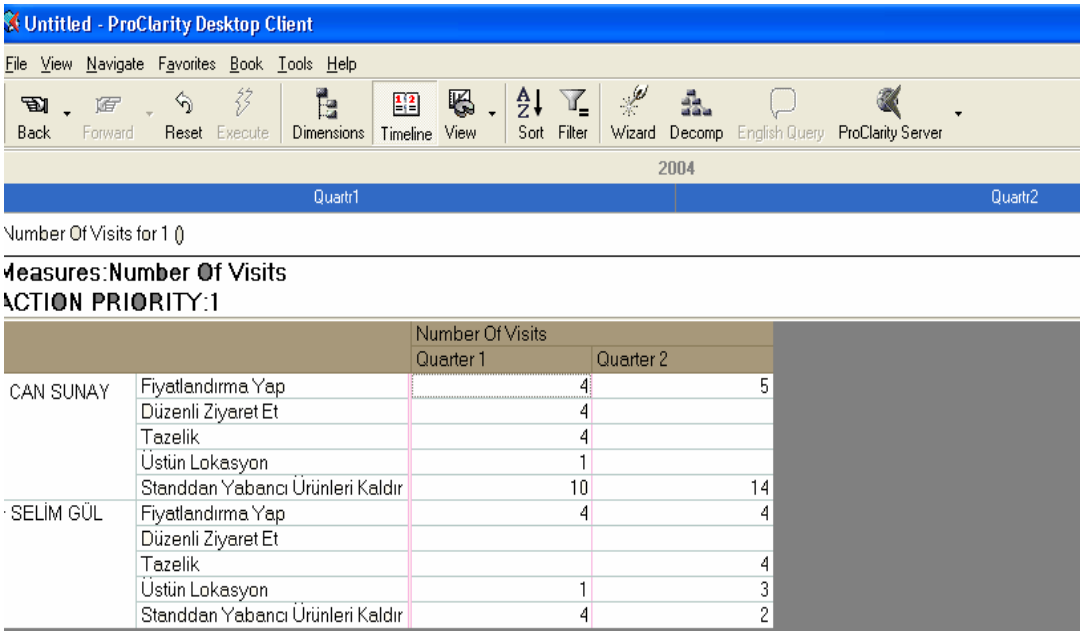
```
                        oColumn.DataType = 129
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = True

                oTransformation.SourceColumns.Add oColumn
                Set oColumn = Nothing

                Set oColumn = oTransformation.DestinationColumns.New("PARENT_HIERARCHY_ID" , 1)
                        oColumn.Name = "PARENT_HIERARCHY_ID"
                        oColumn.Ordinal = 1
                        oColumn.Flags = 104
                        oColumn.Size = 10
                        oColumn.DataType = 129
                        oColumn.Precision = 0
                        oColumn.NumericScale = 0
                        oColumn.Nullable = True

                oTransformation.DestinationColumns.Add oColumn
                Set oColumn = Nothing

        Set oTransProps = oTransformation.TransformServerProperties


        Set oTransProps = Nothing

        oCustomTask7.Transformations.Add oTransformation
        Set oTransformation = Nothing

End Sub

'------------- define Task_Sub8 for task DTSTask_DTSExecuteSQLTask_1 (TABLO_BOSALT)
Public Sub Task_Sub8(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup

Dim oCustomTask8 As DTS.ExecuteSQLTask2
Set oTask = goPackage.Tasks.New("DTSExecuteSQLTask")
Set oCustomTask8 = oTask.CustomTask

        oCustomTask8.Name = "DTSTask_DTSExecuteSQLTask_1"
        oCustomTask8.Description = "TABLO_BOSALT"
        oCustomTask8.SQLStatement = "DELETE FROM OLAP_FACT_TABLE_RM WHERE
VISIT_DATE>=DATEADD(WEEK,-1,VISIT_DATE)" & vbCrLf
        oCustomTask8.SQLStatement = oCustomTask8.SQLStatement & "        TRUNCATE TABLE
OLAP_RM_ACTIONS " & vbCrLf
        oCustomTask8.SQLStatement = oCustomTask8.SQLStatement & "        TRUNCATE TABLE
OLAP_RM_CUSTOMERS " & vbCrLf
        oCustomTask8.SQLStatement = oCustomTask8.SQLStatement & "        TRUNCATE TABLE
OLAP_RM_PAYMENTTYPE " & vbCrLf
        oCustomTask8.SQLStatement = oCustomTask8.SQLStatement & "        TRUNCATE TABLE
OLAP_RM_PROMOTIONS " & vbCrLf
        oCustomTask8.SQLStatement = oCustomTask8.SQLStatement & "        TRUNCATE TABLE
OLAP_RM_STATUS " & vbCrLf
        oCustomTask8.SQLStatement = oCustomTask8.SQLStatement & "        TRUNCATE TABLE
OLAP_RM_WORKERS_HY"
        oCustomTask8.ConnectionID = 2
        oCustomTask8.CommandTimeout = 0
        oCustomTask8.OutputAsRecordset = False

goPackage.Tasks.Add oTask
Set oCustomTask8 = Nothing
Set oTask = Nothing

End Sub
```

```
'------------- define Task_Sub9 for task DTSTask_DTSOlapProcess.Task_1 (CUBE_PROCESS)
Public Sub Task_Sub9(ByVal goPackage As Object)

Dim oTask As DTS.Task
Dim oLookup As DTS.Lookup

Dim oCustomTask9 As DTSOlapProcess.Task
Set oTask = goPackage.Tasks.New("DTSOlapProcess.Task")
Set oCustomTask9 = oTask.CustomTask

        oCustomTask9.Name = "DTSTask_DTSOlapProcess.Task_1"
        oCustomTask9.Description = "CUBE_PROCESS"
        oCustomTask9.TreeKey = "NADUS\NISAN_TEZ"
        oCustomTask9.ItemType = 1
        oCustomTask9.ProcessOption = 0
        oCustomTask9.IncrementallyUpdateDimensions = False

goPackage.Tasks.Add oTask
Set oCustomTask9 = Nothing
Set oTask = Nothing

End Sub
```

APPENDIX E

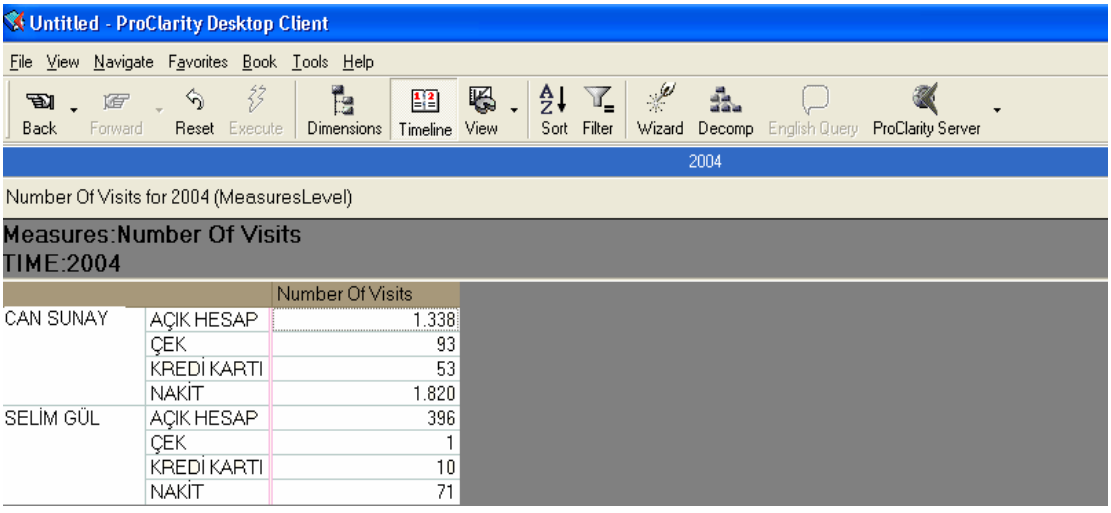
REPORT EXAMPLES

Report 1: Below is the information regarding how many times two staffs have given priority to the selected pre-cautions during two different periods of time.



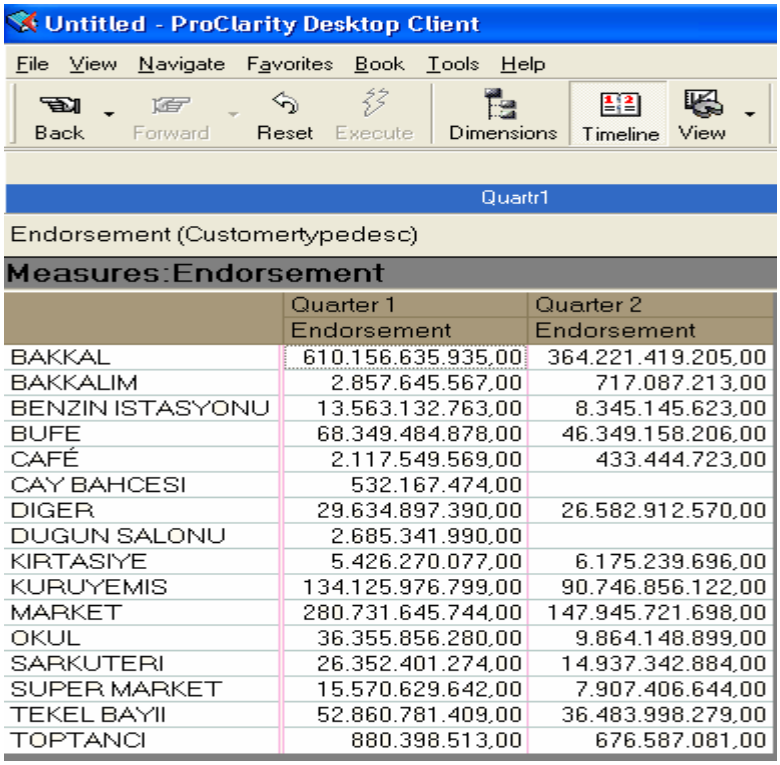
		Number Of Visits	
		Quarter 1	Quarter 2
CAN SUNAY	Fiyatlandırma Yap	4	5
	Düzenli Ziyaret Et	4	
	Tazelik	4	
	Üstün Lokasyon	1	
	Standdan Yabancı Ürünleri Kaldır	10	14
SELİM GÜL	Fiyatlandırma Yap	4	4
	Düzenli Ziyaret Et		
	Tazelik		4
	Üstün Lokasyon	1	3
	Standdan Yabancı Ürünleri Kaldır	4	2

Report 2: Customers have been classified according to payment type in the report below. This report illustrates the number of visits paid by Can Sunay and Selim Gul to different types of customers during the year 2004. It is seen that Can Sunay's customers mostly prefer "cash" payment type.



		Number Of Visits
CAN SUNAY	AÇIK HESAP	1.338
	ÇEK	93
	KREDİ KARTI	53
	NAKİT	1.820
SELİM GÜL	AÇIK HESAP	396
	ÇEK	1
	KREDİ KARTI	10
	NAKİT	71

Report 3: This report shows the endorsement of client groups (regarding to customer type description) in Quarter 1 and Quarter 2.



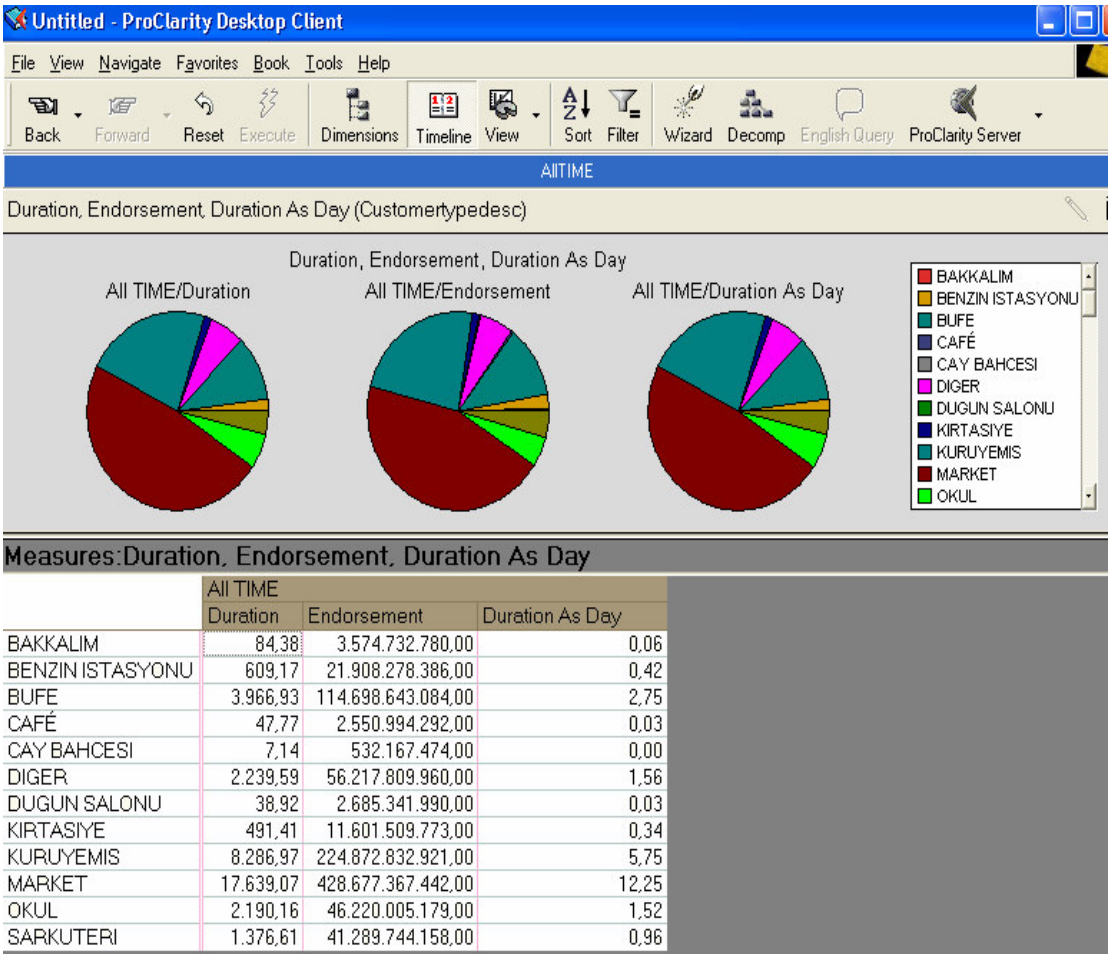
Quartr1

Endorsement (Customertypedesc)

Measures:Endorsement

	Quarter 1 Endorsement	Quarter 2 Endorsement
BAKKAL	610.156.635.935,00	364.221.419.205,00
BAKKALIM	2.857.645.567,00	717.087.213,00
BENZIN ISTASYONU	13.563.132.763,00	8.345.145.623,00
BUFE	68.349.484.878,00	46.349.158.206,00
CAFÉ	2.117.549.569,00	433.444.723,00
CAY BAHCESI	532.167.474,00	
DIGER	29.634.897.390,00	26.582.912.570,00
DUGUN SALONU	2.685.341.990,00	
KIRTASIYE	5.426.270.077,00	6.175.239.696,00
KURUYEMIS	134.125.976.799,00	90.746.856.122,00
MARKET	280.731.645.744,00	147.945.721.698,00
OKUL	36.355.856.280,00	9.864.148.899,00
SARKUTERI	26.352.401.274,00	14.937.342.884,00
SUPER MARKET	15.570.629.642,00	7.907.406.644,00
TEKEL BAYII	52.860.781.409,00	36.483.998.279,00
TOPTANCI	880.398.513,00	676.587.081,00

Report 4: What is the time in terms of minute or day spent at visits of customer groups? What is the endorsement according to customer groups? Answers of these two questions can be found by using graphics. It can be seen from the sample that market group visits have taken most of the time, and the endorsement is higher than others.



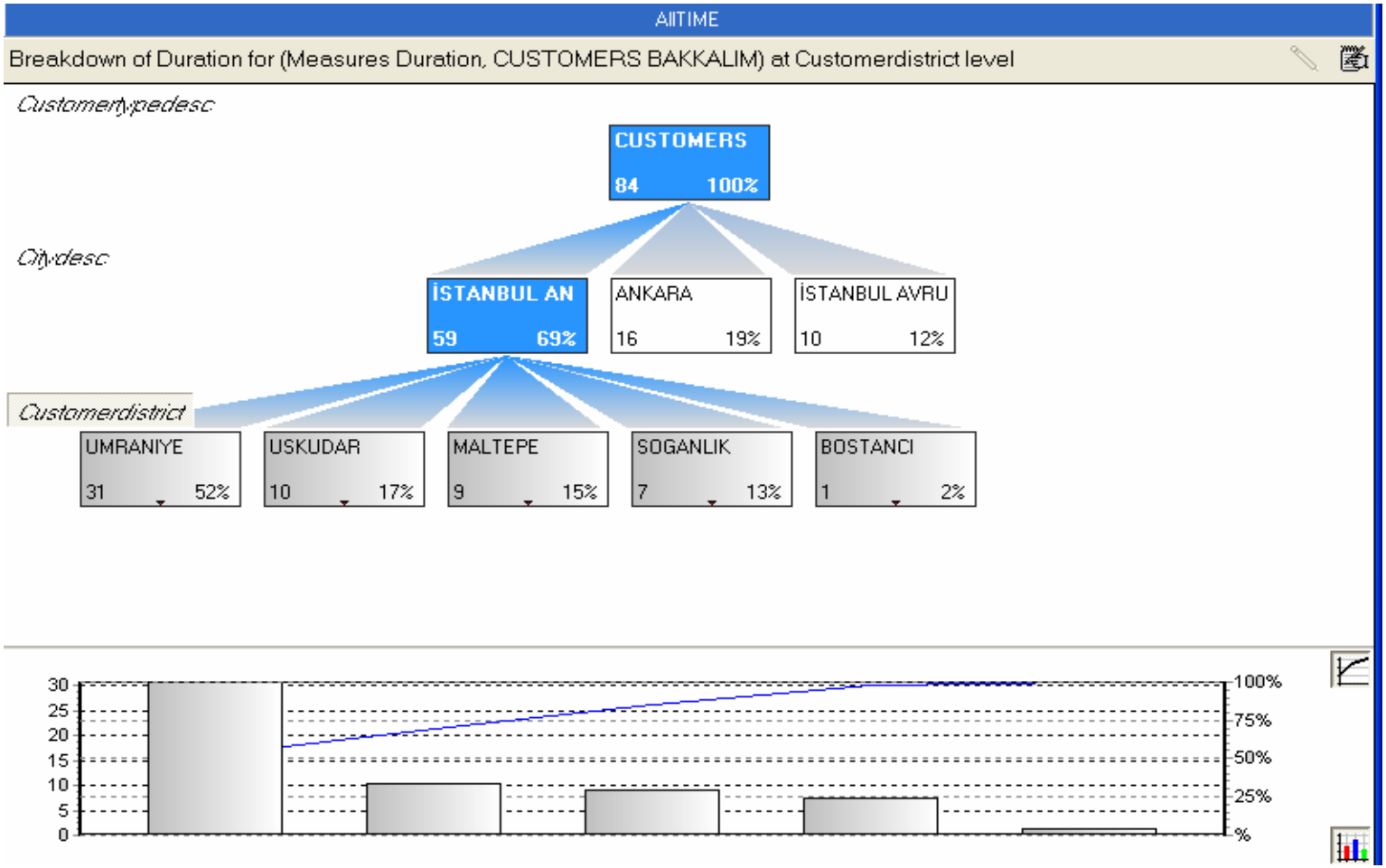
Measures:Duration, Endorsement, Duration As Day

	All TIME		
	Duration	Endorsement	Duration As Day
BAKKALIM	84,38	3.574.732.780,00	0,06
BENZIN ISTASYONU	609,17	21.908.278.386,00	0,42
BUFE	3.966,93	114.698.643.084,00	2,75
CAFÉ	47,77	2.550.994.292,00	0,03
CAY BAHCESI	7,14	532.167.474,00	0,00
DIGER	2.239,59	56.217.809.960,00	1,56
DUGUN SALONU	38,92	2.685.341.990,00	0,03
KIRTASIYE	491,41	11.601.509.773,00	0,34
KURUYEMIS	8.286,97	224.872.832.921,00	5,75
MARKET	17.639,07	428.677.367.442,00	12,25
OKUL	2.190,16	46.220.005.179,00	1,52
SARKUTERI	1.376,61	41.289.744.158,00	0,96

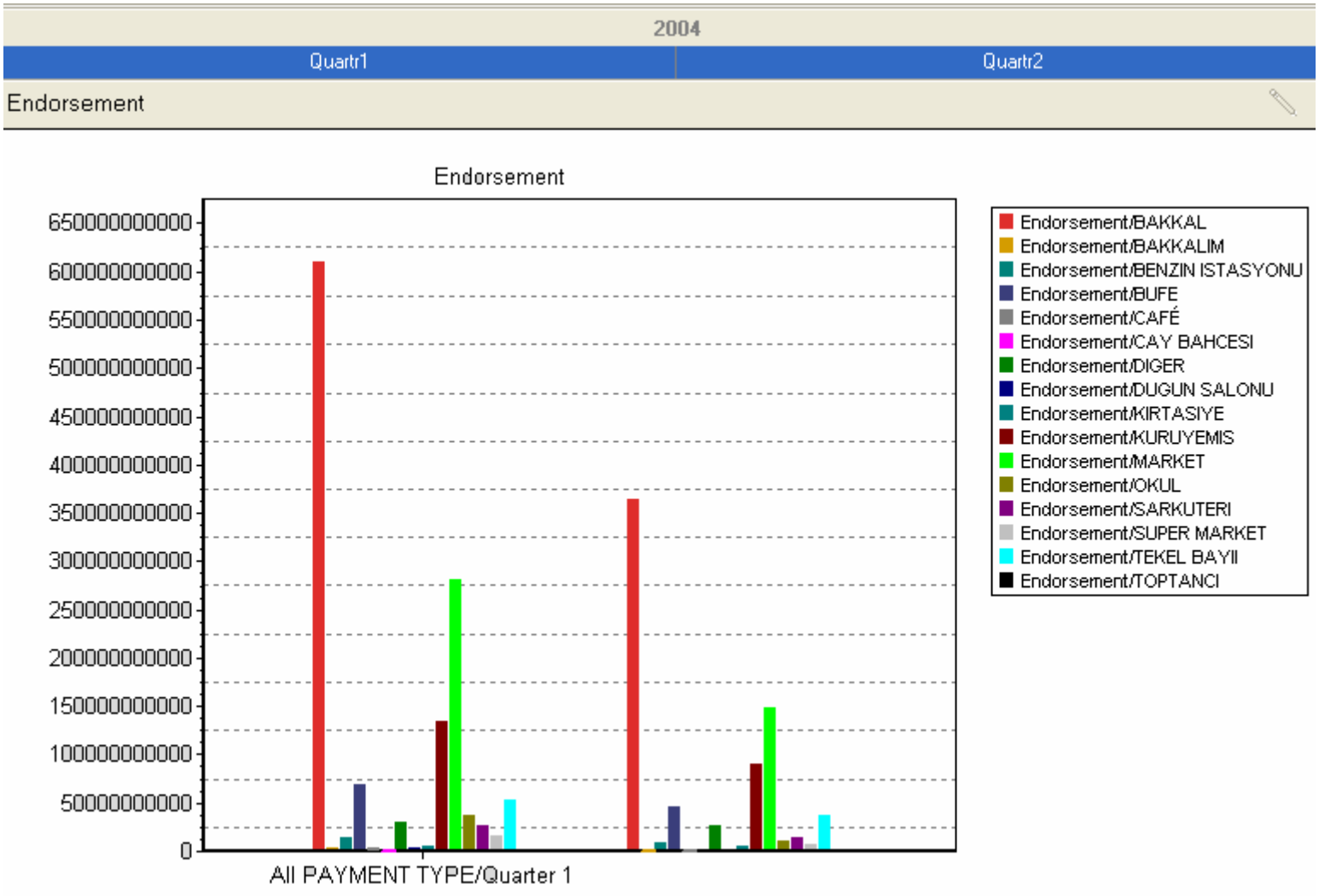
Report 5: The "Bakkalım" customer group in Ankara, Istanbul the Anatolian side and Istanbul European side has been studied in this report. How much time has been spent for which district in İstanbul Anadolu side can be observed by using drill down method. It can be seen that the time spent for visits to Umraniye area is more than others.

(cont. on next page)

Report 5 (cont.) :



Report 6 :


2004
Quartr1
Quartr2
Endorsement
Endorsement
650000000000
600000000000
550000000000
500000000000
450000000000
400000000000
350000000000
300000000000
250000000000
200000000000
150000000000
100000000000
50000000000
0
All PAYMENT TYPE/Quarter 1
Endorsement/BAKKAL
Endorsement/BAKKALIM
Endorsement/BENZIN ISTASYONU
Endorsement/BUFE
Endorsement/CAFÉ
Endorsement/CAY BAHCESI
Endorsement/DIGER
Endorsement/DUGUN SALONU
Endorsement/KIRTASIYE
Endorsement/KURUYEMIS
Endorsement/MARKET
Endorsement/OKUL
Endorsement/SARKUTERI
Endorsement/SUPER MARKET
Endorsement/TEKEL BAYII
Endorsement/TOPTANCI

Report 6 (cont.): Above graphic shows the Quarter 1 and Quarter 2 endorsement of all customers classified according to customer type description. The highest amount of endorsement belongs to “Bakkalım” group in both periods.