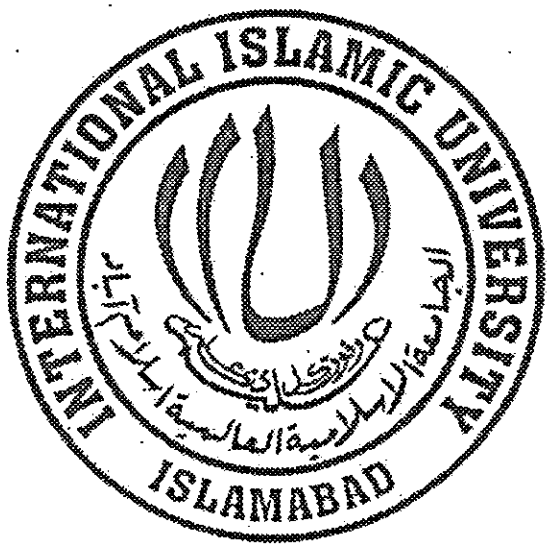
Schema Transformation OLTP Systems to OLAP Systems



Developed By:

Muhammad Ilyas

(237N-FAS/MSDCS/S05)

Supervised By

Dr. Sikandar Hayat Khiyal

Co Supervised By

Muhammad Imran Saeed

Department of Computer Science

Faculty of Basic Applied Science

International Islamic University Islamabad

2009

In the Name of ONE

Who has all the Names

A thesis submitted to the

Department of Computer Science

International Islamic University Islamabad
As a partial fulfillment of requirements for the award of
The degree of

MS in Computer Science

Dedicated
To

The Holiest Man Ever Born
Prophet Muhammad (P.B.U.H)
&
To

Our parents and Families
We are most indebted to our parents and families, whose affection has always been the Source of encouragement for us, and whose prayers have always been a key to our Success
&
To

Those Holy Seekers
Who give away their lives to make the stream of life flow Smoothly and with justice
&
To

Our Honorable Teachers
Who have been a beacon of knowledge and a constant source of inspiration, For our whole life span.

Declaration

We hereby declare that "Schema Transformation OLTP Systems to OLAP Systems", neither as a whole nor as a part has been copied out from any source. It is further declared that I have developed the algorithm and the accompanied report entirely on the basis of our personal efforts and under the sincere guidance of my supervisors. No portion of the work presented in this dissertation has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Ilyas
(237N-FAS/MSDCS/S05)

Acknowledgment

All praise to Almighty Allah who has guided us in undertaking work on the project "Schema Transformation OLTP Systems to OLAP Systems"and has helped us through at each step when I felt there was no hope of pulling through.

I am very obliged to our supervisor Dr. Sikandar Hayat Khyal to provide us technical help for each and every problem faced during my research work and provide me all information about research methodology and always guided me what to do next. I also wish to express my appreciation to my co-supervisor Sir Imran Saeed who help me a lot and provided his full cooperation.

I would also like to thank our colleagues and faculty members of the University.

Finally I owe a lot to our beloved parents and my family for their love, guidance, moral and financial support.

Project in Brief

Project Title:	**Schema Transformation OLTP Systems to OLAP Systems**
Undertaken By:	**Muhammad Ilyas**
Supervised By:	**Dr. Sikandar Hayat Khiyal**
Start Date:	**May 2008**
Completion Date:	**January 2009**
Tools and Technologies:	**Microsoft Visual Basic, SQL Server 2000**
Documentation Tools:	**MS word, MS Excel**
Operating System:	**MS Windows Server 2003**
System Used:	**Pentium 4, 2 GHz**

ABSTRACT

Data warehouse plays an important role in automated decision making for any organization. We propose an algorithm for automatic generation of OLAP logical schema from OLTP logical schema.

Our proposed algorithm uses top-down source driven approach to minimize the number of iterations for finding exact fact tables. Our proposed algorithm uses more specifications for finding fact entities and measures thus minimizing the manual refinement done by the user. It also considers many-to-many relationship problem that breaks the hierarchy.

Our proposed algorithm is five fold. In first fold it captures all the Meta data information from the source OLTP systems. In second fold it segregates all the starting points of the enterprise source schema. In third fold it finds out the hierarchy. In fourth fold entities are classified and at the end generation of target OLAP schema. That is star schema.

Department of Computer Science

International Islamic University, Islamabad

Date: _____

Final Approval

It is certified that we have read the project "Schema Transformation OLTP Systems to OLAP Systems" submitted by **Mr. Muhammad Ilyas Reg. # 237N-FAS/MSDCS/S05.** It is our judgment that this project is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree in MS computer Science.

COMMITTEE

External Examiner

Dr Nazir A. Sangi

Chairman DCS,
Allama Iqbal Open University, Islamabad

Internal Examiner

Muhammad Nadeem

Asst. Professor, Department of computer Science
International Islamic University, Islamabad

Supervisor:

Dr. Sikandar Hayat Khiyal
Prof. FJWU, the Mall, Rawalpindi

Co-Supervisor:

Mr Imran Saeed
Asst. Professor, Department of computer Science
International Islamic University, Islamabad

1. Introduction

1.1 Introduction

The Data Warehouse systems have become bread and butter for decision support systems to provide platform for knowledge discovery. The concept of Data Warehouse (DW) system is very simple but its implementation is not [1]. For the design of data warehouse systems information form user as well as the structure of underlying source systems play and important role. In simple words a DW is just a database of databases but the characteristics of this type of database are different. The data warehouse in such type of database which is subject-oriented, integrated, non-volatile and time-variant. Moreover these kind of databases are used to support the management's decisions [2]. The following diagram describes the basic architecture of DW system.

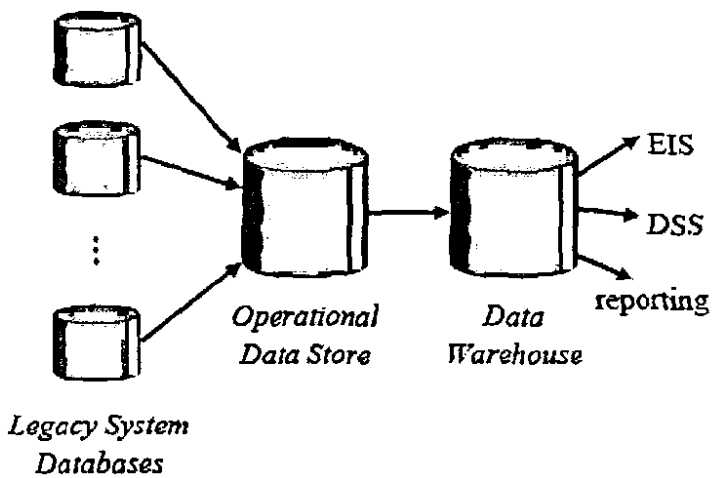


Fig 1.1 Data Warehouse Architecture

The main components of the architecture are as follows

- Legacy System Databases: these are the systems which record details of the business transactions [3].
- Operational Data Store: an intermediate between DW and legacy Systems. The data from legacy systems is first loaded into this and then finally aggregated, reconciled and loaded into DW in a consistent format [3].
- Data Warehouse: a central source of decision support data across enterprise [3].

- End Users: write queries and perform analysis against data stored in Data Warehouse [3].

From last decade big organizations in Pakistan are interested in implementing data warehousing system. Information is most important asset for any organization. Mostly big organizations kept this asset in two forms, operational system and data warehouse. Simply speaking about Operational system where we store the data and data warehouse systems are used to get the data out for MIS. In OLTP systems day to day transactions are stored where as OLAP systems have historical data. [2]

In this chapter we discuss the motivation and challenges of our area of research, then background information and research domain. Then we continued with proposed approach of the problem. At the end of the chapter thesis outline is put forth.

1.1 Motivation and Challenges

Pakistan is developing country. There is no doubt; in our country most of the organizations are late to implement new technology. Data warehouse is quite old technology but only few of the organizations benefited from this technology in Pakistan. Most of them are multinational. Most of the organizations in our country have OLTP systems called legacy systems. The challenge is to design OLAP (data warehouse) system over these legacy systems as second layer. We are facing this dilemma as our country lack human expert human resources to implement this technology.

By keeping in view these problems it is decided to design a robust algorithm to convert legacy OLTP systems design the OLAP design. By this algorithm time required to design data warehouse will enormously reduced as compared to other algorithms.

1.2 Background

Logical schema is dependent on conceptual schema. Conceptual schema can be created by translating user requirements into pictorial representation. This is understandable to

the user and independent of implementation issues, but is formal and complete, so that it can be transformed into the next logical schema without ambiguities [4].

It is required to make models for solving any kind of problems. For the design of data ware house dimensional modeling plays an important role. Basically dimensional models are understandable for both users and system analysts. Therefore dimensional modeling is used to bridge the gap of understanding between users, designers and system analysts during the requirement gathering and data warehouse design phase. These models are used as a conceptual model as an understanding between users and analysts. And useful to avoid any kind of technical jargons during the remaining phases of data warehouse development. The modeling phase is very important building block, as errors could be detect in this phase. Moreover it is also useful during the further extensions in the data warehouse project [5, 6]. It is largely accepted that the dimensional modeling is the basic building block for the design of data warehouse.

.

Srivastava and Chen at el [7] propose three step formulae. These three steps are DW architecture selection, DW Pre-Conversion Phase and data warehouse maintenance. These three steps are elaborated in section 1.2.1, 1.2.2 and 1.2.3 subsequently. These three steps are discussed by keeping view the all levels of schema i.e. conceptual, physical and logical.

1.2.1 Pre-Conversion Phase

Data warehouse development requirements are typically defined in general statements but project objectives need to be specific. They should answer such questions as, "What do I want to analyze and why do I want to analyze it?" By answering these types of questions, we get an understanding of the requirements [8, 9, 10, 11, 12, 13, 14, 15, 16]. Data warehouse requirements contrast with typical application requirements. There are many methods for deriving business requirements. In general these methods can be placed in one of two categories: Source driven and user driven requirement gatherings. [10].

In Source-driven requirements gathering we take requirements directly from OLTP systems in other words OLTP systems are staring points to gather Source driven requirements. Automation in modeling of data warehouse generally uses a source-driven requirement method [17, 18]. In this thesis for automated data warehouse design, source driven approach is used. . There are lots of benefits of using this approach as minimum time is required to complete the design and ultimately complexity of ETL process reduced as the design is extracted directly from source schema. The main disadvantage of this approach is that it didn't cater fall all the user requirements because of the limited dimensions of OLTP design structure and limited data. In this approach, user involvement is very less. Therefore there are chances that the output structure may not full fill all the requirement of the users. To overcome these disadvantage the user given the liberty to refine the schema manually.

The main advantage of users driven approach is that it requires less time to complete the projects. Because designer conceders only what is required. May be there is some other information available missed. In this approach all the information in collected by having meetings with deferent level of users. In other words system analyst only considers the function performed by the users. That is why time required to complete the project is shorter as compared to other approaches. [8, 12, 15, 16]. Thus this is a big advantage as the scope of the project is focused on user requirements. The main disadvantage of the above mentioned approach is that it is impossible to cope up with the user requirement with the passage of time. It means the user satisfactory time is less in this approach. In this thesis we follow source-driven requirements approach. Source-driven approach is useful to gather all data which exits somewhere in the organization. We describe the proposed algorithms in our thesis by experimenting on three sample schemas in SQL Server 2000.

Capacity planning is the sub phase of the pre-development phase of data warehouse. Before the development of any kind of data warehouse, it is required to perform capacity planning. In capacity planning of any kind of data warehouse number of measures are analyzed. For example daily data arrival size, daily growth of data, monthly and yearly

growth of data. And for how long the data will be available for analysis, number of users and their expect growth with the passage of time. Although capacity planning is not required for our proposed algorithm, it may be required for the decision to select hardware or software for our algorithm.

1.2.2 Selection of Architecture

Many factors required consideration to select the best architecture of data warehouse design. It depends upon the existing infrastructure of the organization, the policies of the management and the technical expertise of the staff of the organization and their availability [8]. Architecture selection in useful to determine the scope of the project typical functions performed by an organization. Architecture selection process mainly focus on the structure of out put schema and the desired translation processes required to translate data from the source OLTP schemas to data warehouse.

The two popular and common architectures for data warehouse design are Bottom-up and top down. There are tot of research on these architectures to derive some extended architecture which are called derivatives of the above two. In top-down approach a central data warehouse is created which may have several data marts. These data marts are subsets of the central data warehouse. Data marts are a mini data warehouse, which represent a department of the whole organization, which can perform function independently. On the other hand in bottom up architecture data marts are designed first. Then these data marts are interconnected with each other to form central data warehouse [8, 19, 20]. In bottom-up architecture the process of data warehouse creation is incremental. Firestone at el [19] gives lot of discussion about data warehouse architectures.

The necessary resources are required for determining the factors for architecture selection of any kind of data warehouse. The cost of initial planning and design is significant when top-down architecture approach is followed. This approach is very time consuming and definitely having impact on ROI [8]. The incremental approach (bottom-up) requires less

time for the initial implementation of data warehouse. As data marts are smaller as compare to data warehouse, less time is required to build them.

Bottom-up architecture is very popular in industry as it has significant advantages over the top-down approach. However top-down architecture has a single edge on bottom up architecture. That advantage is of consistency between data marts and data warehouse, a data marts are derived from the central data warehouse. Moreover it is easier to implement organizational policies in central data warehouse. On the other hand in bottom up architecture the main disadvantages are consistency and redundancy. Lots of efforts are required during combing different data marts to make central data warehouse to reduce redundancy and increase consistency. The proposed algorithm which automatically design data warehouse schema is independent of the architecture. This automation algorithm is totally dependent on input OLTP schema. If the source OLTP schema is the subset schema of organization then data marts will be created. On the other hand if entire enterprise OLTP schema is given as in input to proposed algorithm then whole OLAP schema will be created for the organization. Our algorithm is robust enough to give output for both types of inputs. Our algorithm is totally dependent on the input logical schema of the OLTP systems.

1.3 Problem Domain

The objective to automate the data warehouse design is to minimize the time and cost. There are different goals and objectives to create any kind of data ware house. Important objectives and goals are given below [11]:

1. Reconcile different views of data
2. Consistency,
3. It should be flexible enough to cover all perspectives of analysis (slice and dice),
4. All kind of business practices should be covered.
5. Provide a consolidated picture of enterprise data
6. The data should be qualitative for business re-engineering.

The Schema creation for OLTP systems or for OLAP systems is designed by keeping in view the needs of business users and executive decision makers. Commonly ER models are used for the design of OLTP systems on the other hand dimensional modeling is used for OLAP systems. Actually these models are pictorial representation of organizational business trends. During the design of data warehouse all the above mentioned goals and objectives are considered. There are three levels of schema i.e. conceptual, logical and physical. The conceptual schema is a high level of representation of business units and data flow of the business entities. This level of schema is more users friendly. Users can give feed back if there is any change in the design well before implementation. The logical design of the schema could be inferred from the conceptual schema. Logical schema represents the tabular arrangement and relationships between the tables. At the physical level storage and indexes are analyzed. These three levels are discussed in various literatures which include ME/R [21, 22], Star [8], StarER [23], and dimensional fact model [24]. In this thesis we are covering only logical design of the data warehouse. Our algorithm generated logical schema for OLAP as an output.

1.3.1 Decision Support Terminology and Functionality

OLTP system didn't provide support for decision support systems. The advent of OLAP systems provides this interesting capability. OLAP systems are designed in provide support for decision support systems. This functionality is not available in OLTP systems. An OLAP system provides additional functionality for query processing by the concept of Cubes. Cubes are used to represent the multidimensional data for analysis. General understanding of the cube is that it has three sides. But there is possibility of cube having more than three sides. These types of cubes are called hyper cubes. Both dimensional and ER based models could be used to represent the cubical data. Both modeling techniques are different in pictorial representation, and each of them has distinct notations for modeling representation. Both of techniques are popular in database modeling. But dimensional modeling is popular for data warehouse modeling. We will discuss both of them in coming sections.

1.3.2 ER Models

These models are designed during requirement gathering phase in for predevelopment rectification of design bugs. ER models are used to represent the relationship between entities. ER stands for Entity and relation. As ER models are high level pictorial representation of the design of the system, therefore it is quire easy for user understanding. ER model designed for data consistency and reduction in data redundancy. Entities and relationships are two basic concepts of ER models. The detailed representation of ER model may include attributes of the entities also.

A sample ER-based model is shown in the figure below. An entity represents a real world object, shown as a rectangle. The ER based model is explained with the help of an example. Product and supplier are two entities are represented by a rectangle. One entity could participate to make relation with one or more entities. The diamond symbol is used to represent the relationship between two or more entities. As mentioned in the picture below the relation supplies acts a bridge between entities Product and suppliers. In ER based models relationship cardinality is represented by ordered pairs (m,n), m and n are variables. Cardinality of a relation could be of four types

1. One to One (1,1)
2. one to Many (1,m)
3. Many to One (n,1)
4. Many to Many (n,m)

If we explain these four type of cardinality with the help of figure below. It is possibility that one product could be supplies by only one suppler and called one to one (1, 1). A suppler could supply many products and called one to many. Similarly there is many to many relationship (n,m) between author and books entity. Actually entities are basically used to group data with one primary key and other attribute to represent data. ER models are the conceptual design of relational database. ER based models are designed by using top down approach. In detail discussion could be found about ER based models in [8].

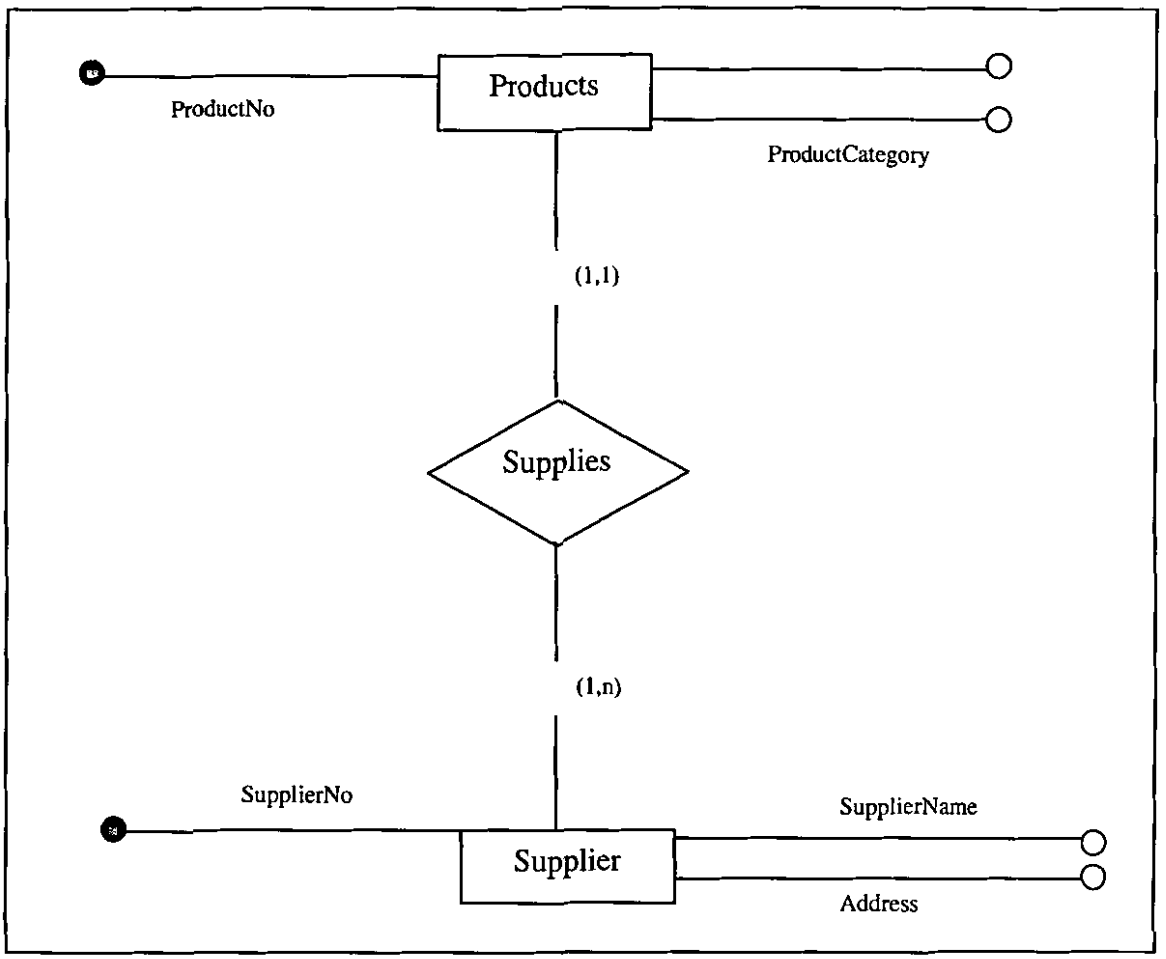


Fig 1.2: Sample ER Schema

ER modeling has been proved to be good for conceptual representation of OLTP systems. And users are well familiar with this kind of modeling. The extensions of ER models could be used for data warehouse design also.

1.3.3 Dimensional Models

Dimensional modeling is a technique used to design database of data warehouse. Dimensional modeling is a significant contribution in the data warehouse research. Dimensional modeling is very useful for query processing and query optimization. ER based models could easily be converted into dimensional models by having some demoralization process. Dimensional modeling is very user friendly for practitioners and users. It is a great break through in data warehouse world to solve the dilemma of

understanding of users. . A sample dimensional model (star schema) is shown in figure below.

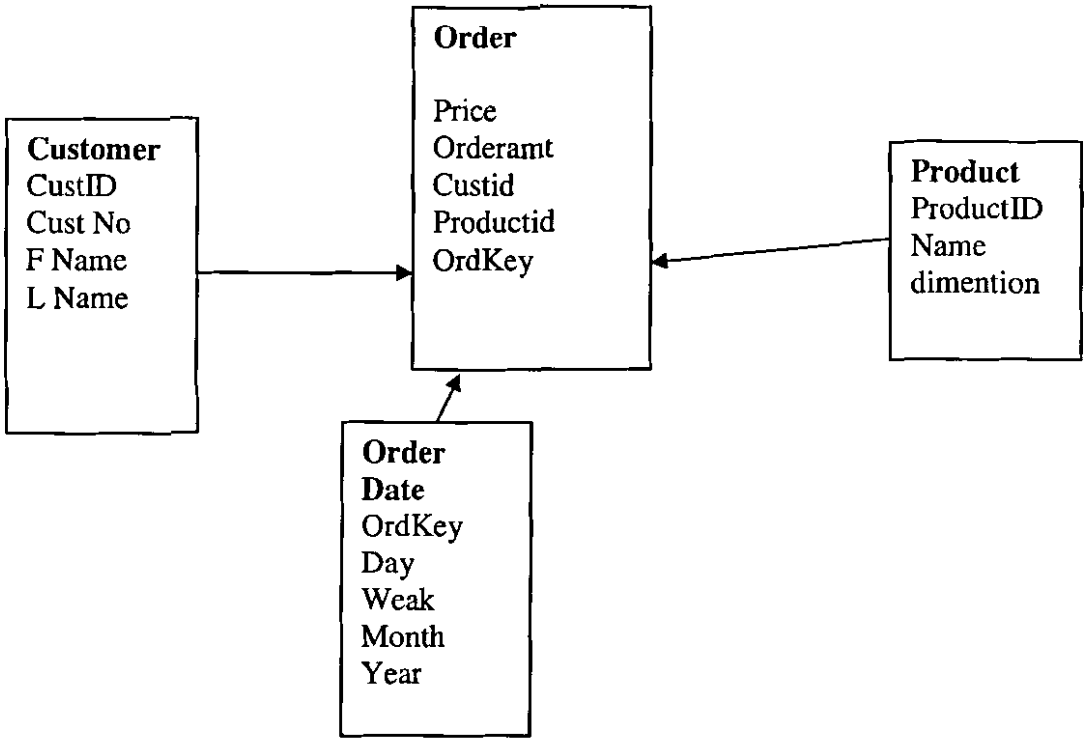


Fig 1.3: Dimensional Star Schema

There are different models to represent dimensional model i.e Star schema, Snow Flack etc. A Dimensional Modeling is combination of Fact and Dimension tables as shown in the figure above. Fact tables and dimensional tables are basic building blocks of the dimensional modeling. Fact table contain all the measure which are generally numeric. The tables contacted directly to the fact tables are called dimensional table. The date time dimension is the default dominion, which is part of every dimensional model. Basically dimension plays and import role for the level of aggregation. They are direct indictors for the summarization of measures stored in the fact table. In the above diagram Order table is the fact table and remaining are the dimensions. An order table has all the foreign keys which are primary key for every dimension. Every event occurred will be the part to the fact table. For the efficient query processing in dimensional modeling the concept of

cubes are used. Every dimension will made a side of cube and fact table data on the other hand will be shared on each side of the cube. Every record of fact table has reference with the attached dimension. In other words the fact table is the heart of dimensional modeling. Analysts are interested to see the same data from different angles. The dimensions are capable to summarize data of fact table according to the information available in the dimensions. In dimensional models the fact table acts as an intersecting point for every dimension. In figure above order table is the intersecting point for every dimension attached directly with it. Order date is the dimension having attributes day, weak, month and year. The data in the fact table could be summarized or aggregated on the basis of weak, month and year. More information about the fact and dimension relation could be found in [8, 17, 15].

There are many types of dimensional modeling. But the concept of all these types is almost same. In all these models all the organizational data surround the central fact table. The most important types are Star model and Snowflake model. Snowflake model is the derivation of Star model [8]. In Snowflake dimensions have hierarchies. But in Star model these hierarchies are clasped in the master dimension.

In Snowflake model the dimensional tables are normalized like ER models. In Start model dimensions are collapsed upward in the hierarchy. That is why Star model is simple as compare to Snowflake model. In star model there is duplication of data while in snowflake data is less duplicated due to the normalization of dimensions. Both the models are equally practicable in the industry. Any how snowflake model have some advantage due to its robustness. Star schema is good in conceptual design by snowflake is best for logical representation of schema. Because in Snowflake model dimensions have the concept of master detail [8]. Snow Flake schema is shown in the figure below. Region is the detail of city dimension. Dimensional modeling is best for data warehouse high level as well as logical design.

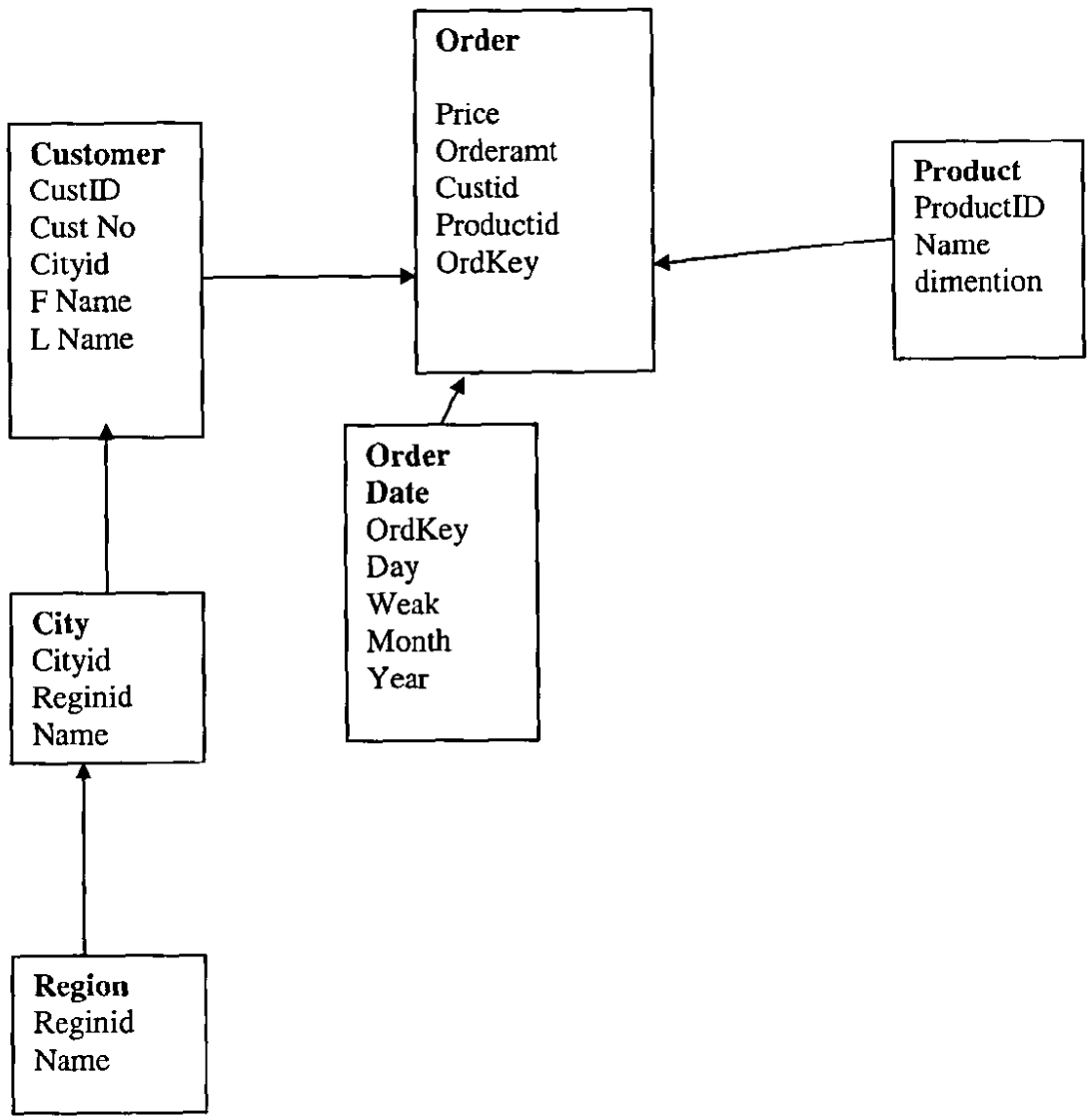


Fig 1.4: Snowflake Schema

1.3.4 Model Used for Schema Creation

Ballard at el [8] concludes that dimensional modeling is better than that of ER modeling. There is very less difference between ER and dimensional modeling. The logical design of dimensional model is more efficient for query processing but not for data pupation. On the other hand ER design is robust for data pupation and less efficient for reporting purpose. Because in ER modeling the data is normalized. ER models is optimized for

data updating on the other hand Dimensional modeling is optimized for data retrieval. The ER model is transaction processing model on the other hand dimensional model is data warehouse model.

Organizational business rules are focal point in dimensional modeling. In user point of view the automation of business roles are most important. This functionality could be provided through dimensional modeling in the form of fact table as it is center for all dimensions. This is one of most popular model for user understandability pint of view. The dimensional model is capable to answer following queries. "How much deposit is collected in the month of December in Pakistan," the query contains the summation of all deposits in the month of December in Pakistan. Moreover ER based model are not capable for how the business is measured [25, 15, 26].

Dimensional modeling technique is the most suitable technique for data warehouse design. Data warehouse is a specific remedy for decision support systems. In decision support more emphasis is given on data retrieval rather than data insertion and updating. Dimensional modeling provides this kind provision. Updating and insertion took place after some specific interval, weakly, monthly and quarterly basis. Dimensional modeling structure is not ideal for OLTP systems. Because OLTP systems are designed for day to day transactions. Where data insertion and updating is important. To optimize the query different query optimization algorithms are used. The major focus of dimensional modeling is understandability of users.

Dimensional model is scalable as compared to ER model. It is very easy to add attributes in the dimension or any new dimension in the model. Similarly facts can be change easily without affecting the existing structure. Dimensions could be broken in more than one table [25]. Dimensional modeling also handles slowly changing dimensions. These slowly changing dimensions handled according to the organizational structure. Dimensional models are structured in such a way that if structure changed in running data warehouse. It could be easily managed without having any impact of the existing data. All these advantages are discussed in more detail in the literature [8, 11, 27].

Dimensional modeling is one of the best technique in situation where changes occurred frequently in structure to give efficient query performance and user satisfaction. User is satisfied because there is more provision to slice and dice the data, different level of aggregation and summarization. On the other hand ER model deprived of such kind of functionality. More detail about the comparison of ER and dimension modeling could be found in [28, 16].

In this thesis the models used for schema creation are dimensional in nature. OLTP systems which follow ER modeling provide data for data warehousing. Both Kimball [25] and McGuff [14] devise mechanism of derivation of dimensional models from ER based models. Golfarelli and Rizzi at el [24] describe mechanism of using dimensional model for the design of data warehouse. Ballard et al. [8] have great contribution for the popularity of dimensional modeling. Researcher also used combination fo ER based and dimension models for the design of data warehouse. In [29,17,14,26] ER base models are used for conceptual design of schema where as dimensional model is used for logical design of schema.

.

Logical schema has tables, relationship between the table and columns in tables. Logical schema is responsible to handle data populated or retrieved from production data bases. This level of schema could be design directly or could be generated from conceptual design also. The standardized approach is to create logical schema from the conceptual schema. In logical design conceptual schema is realized in form of tables and relations. Data can be retrieved by simple relational queries. Golfarelli et al. [18] did not create any conceptual schema for logical schema. He directly design logical schema. The last phase of schema design is the physical schema design. In this phase the storage mechanism for the structure of data warehouse and indexes. An approach is also given for automatic physical design by Blaschka [21]. Underlying structure of data warehouse is managed in this physical design phase. Conceptual level, Logical level and physical level all the three stages of data warehouse schema design is necessary for the efficient and robust design of the data warehouse of any organization. In automatic design of data warehouse only

conceptual schema design is considered. No consideration yet given for automatic logical design of schema.

ER and Dimensional model could be used for the design of logical and conceptual schemas in the process of data warehouse creation. Different modeling technique could be used for conceptual schema and logical schema creation. Snowflake, Star, ER, DFM, EVER and ME/R models have been proposed for specification of schema. All of them vary form each other with little difference. How ever Star and Snowflake schema is very popular among the industry. There are different opinion of using Star for conceptual and Snowflake for logical schema design and vise versa. How ever Star it is common opinion to use Star model for both conceptual and logical design of schema.

In these theses we designed our algorithm to generated Star Schema as it is simple and understandable to users. Our algorithm is source driven; there is no input from the user in the process of schema generation. Therefore it very important the out should be simple enough for understandability. So that user should able made amendments are refinement according to their desire. More over in star model query processing speed in fast as compared to Snowflake schema.

1.4 Proposed Approach

Keeping view disadvantages as well enhancements in Cassandra Phipps and Karen C. Davis at el [30] algorithm, we propose an algorithm which uses top-down source driven approach to minimize the number of iterations for finding exact fact entities. Our proposed algorithm uses more specifications for finding fact entities and measures thus minimizing the manual refinement done by the user. It also considers many-to-many relationship problem that breaks the hierarchy.

Our proposed algorithm will comprise of the following steps.

1. Schema Population

2. Finding out starting points
3. Finding Hierarchies
4. Classification of Entities
5. Schema Generation

1.5 Thesis Outline

The thesis is divided into three phases. In the first phase, we review the general architecture of data warehouse design. In depth analysis of different approaches for architecture selection. In second phase we also do an in-depth literature review on the existing semi automated approaches for data warehouse creation. The in third phase the system design and its implementation with a possible solution are proposed to give algorithm for OLAP schema generation from the source OLTP schema with more valuable measures. In the final phase we evaluate our algorithm on sample database to check its consistency and productivity.

2. Literature Survey

candidate fact schemas using user driven approach. Their algorithm still ignores some important aspects of dimensional modeling for example handling of Many-to-Many relationships that breaks hierarchy b/w entities. Also the algorithm performs a large number of iterations for finding out candidate Fact schemas as it works in bottom up fashion. Only one specification is used for measures that are Numeric data type. More specifications are needed to find out measures as well fact entities so that user refinement can be minimized.

According to Ralph Kimball ER based model could be converted to dimensional models. It could be done by some demoralization and aggregation process. Moreover this conversion from ER models to Dimensional model provides substantially useful information to the user about the systems and helps to bridge the gape between these two. More significantly we can say that the structure and data where house is the repacking of the OLTP structures. In three step formula of Ralph Kimball, business units are defined in first step then finding candidate fact table having more numeric values then demoralizing to remaining table to make relation with the fact table [25]. Kimball's approach of translating schema from OLTP systems to OLAP is manual approach. This manual approach uses many to many relations to find table. . Our algorithm also use some specification used by Kimball but we did not consider many-to-many relation to find the fact table. The main preference of our algorithm to find fact table is numeric data and hierarchy level of entity.

Golfarelli, Maio and Rizzi [18] develop a approach for semi-automation to generate a DFM (Dimensional Fact Model) conceptual schema from an OLTP ER schema. Their methodology has following steps: 1) Defining Faction then for each fact 2) following steps should be performed.

1. Construction of attribute tree for fact table.
2. Pruning Attribute tree of fact table.
3. Dimensions are defined for fact table.

4. Attributes for facts.
5. Hierarchies defination.

In this methodology the frequently updated entities determines the facts. Only the creation of attribute tree is automated in this methodology rest of the steps are manual. In this methodology attribute tree is build by placing dimensions and their attributes around the pre selected fact table.

The only automated step in his work is that he made dimension table as a leaf for the fact table. All other steps are manual work, decided by the user on the basis of knowledge one have of particular OLTP system. Fact tables are determined manually. Then dimensions are generated automatically on the basis of relationship from fact table. This work is partly automated but still lot of manual work is required to complete the design of the work. In dimensional modeling fact table is most important entity and this entity is found by the user. According to the method of author most important measures are that which update frequently. The approach in this thesis suggests that the assumption is not correct to determine the measures of interest of user. There is no specific role of traversing entities to generate attribute tree. Such a way every entity will be the part of attribute tree. In an automation point of view it is computationally expensive because additional recourses are required for pruning the generated tree. . In the proposed algorithm hierarchy levels are defined bases on the relationships between the OLTP schema entities. This hierarchy level is very important measure for finding fact table [18].

Another semi automated approach is proposed by Boehnien and Ende at el [17]. This approach uses SERM (Structured Entity Relationship Model) to create Star schema from OLTP schema. In this methodology following stages are observed.

1. Business measures. Are determined
2. Dimension tables and their hierarchies are determined.

3. There should be integrity constraints between dimensions and their hierarchies.

In step 1 Businesses measures are determined by analyzing user requirements as well as business objectives of any organization. In step 2 hierarchies are determined for each dimension, this step is partially automated but further manual refinements are required to achieve the desired results. One drawback to this methodology is that user understanding will be significantly reduces by the loss of aggregation levels representation. But conceptual representation in not the focal point in this research as In this approach the author deal with only logical schema the aggregation paths and hierarchies are not so important.

2.3 Limitations

The main focus of this thesis is to an approach and algorithm for automatic conversion of OLTP logical schema to OLAP logical schema. Our proposed approach is independent of conceptual design. There is no need to design conceptual schema for logical schema generation. However conceptual schema could be obtained from the logical schema generated by our proposed algorithm by reverse engineering. Here is the boundary of our theses.

1. Give an approach for fully automation to find fact table
2. Remove the requirement of conceptual schema for logical design of data ware house
3. Direct finding the fact table from the source OLTP schema instead for traversing each candidate fact table which reduces the processing cost of algorithm.

2.4 Summary

It is very tedious job to find facts and measures during the design of data warehouse. Ant its automation is also a challenging job. We have discussed four approaches in this chapter to partially automate the concept of data warehouse creation. Actually all these

approaches lack one or from some other way. All of these approaches are manual or partly automated. One of them select fact table based on many to many relation and some other non key attributes [25]. In the second approach the queries are analyzed which are performed in data items to measure the business performance [8]. In third approach the writer give important to numeric attributes to find out fact table [18]. In the last approach mentioned in the literature survey he proposed that the entities which are frequently updated in production environment will qualify for fact table. None of the discussed approach claimed to find the hundred percent fact entities. We claimed that our proposed algorithm not only give the hundred percent fact table but also generate whole data warehouse logical schema by having single click of the users.

3. Requirement Analysis

Requirement Analysis

3.1 Introduction

Data warehouse systems are different from conventional operational systems by their design, development and architecture. During the design of OLTP system designer consider only the specific needs of the business. On the other hand the design of data warehouse is generic. The designer has to consider the specific as well as generic needs of the whole organization. For this it is important to meet the user needs and study the underlying source systems. In this thesis we introduce innovative five step algorithm for the data warehouse generation as discussed in first chapter. These five steps are Schema population, finding starting points, finding hierarchies and schema generation.. In this thesis we only consider the logical schema creation and its automation form underlying rational OLTP schemas.

3.2 Problem Scenarios

Most of approaches for data warehouse creation are manual and time consuming. Usually the involvement of the resource person who knows the underlying conventional OLTP system plays and important role in data warehouse creation. It is very time consuming to analyze the underlying resource systems to find the fact tables and their attributes, dimensions and their attributes. So algorithmic (automatic) conversion of OLTP schema to OLAP is challenging.

Very little work is done in this direction, and that work gives algorithm solution of some of part of the conceptual schema as discussed in chapter 2.

3.3 Focus of Research

Keeping in view of the critical scenarios discussed in section 3.2 and to give robust algorithm for logical schema generation from the OLTP systems following goals and targets were set, which ultimately guides the design the robust algorithm.

1. Determine the modeling techniques used to represent logical and conceptual schema.

2. Determine how logical and conceptual schemas could be generated from traditional legacy OLTP schemas.

3. Evaluation and refinement of auto generated logical schema to full fill the user needs and organizational needs.

4. To give a automated approach for logical schema creation of data warehouse by analyzing the l logical schema of underling logical schema of OLTP systems.

5. To design an algorithm for step 4

3.4 Summary

Automation for the logical schema from source traditional operational logical schemas is a challenging task. But algorithm develop in this research will make it quite easy. For this purpose different modeling techniques have been presented and criticized in previous chapter. The deep analysis our goal and objective of the research help us to develop the proposed algorithm. The design will be discussed in next chapter.

4. System Design

System Design

4.1 Introduction

In the design phase all the aspects of the project are going to be covered, according to which the coding and the implementation can be done.
In this chapter the design of algorithm is given which generate a logical schema of data warehouse from source OLTP schema. We use SQL server 2000 sample database to explain the algorithm. In Section 4.2, the design requirements are discussed. Section 4.3 contains the reference architecture for the design of algorithm. Section 4.4 presents an algorithm for conversion of OLTP logical schema to OLAP logical schema.

4.2 Design Requirements

Any relational OLTP logical schema could be the input of the proposed algorithm. But SQL Server 2000 sample database Northwind is used as input to this algorithm. It is difficult to analyze OLTP schema for automatic generation of OLAP schema. Meta data information is very useful for automatic analysis. ER diagram is shown in figure below.

The figure given below is the ER diagram cum tabular representation of the Northwind data base. Each box represents the table and its name on the top. Primary keys are always on the top. The fields having the symbol of keys are primary key fields. For example the table employee having the primary key EmployeeID. The remaining fields are attributes of the table which may contain foreign key. Foreign key attribute is that attribute which are primary key of some other table. For example in the figure given below the EmploeeID in Orders table is foreign key. The line represents relationships between entities. In relationship the line with arrow having cardinality 1 and without and arrow is many. For example relation between Employees and Orders, an employee can place more than one orders but an order could not be placed by more than one employee.

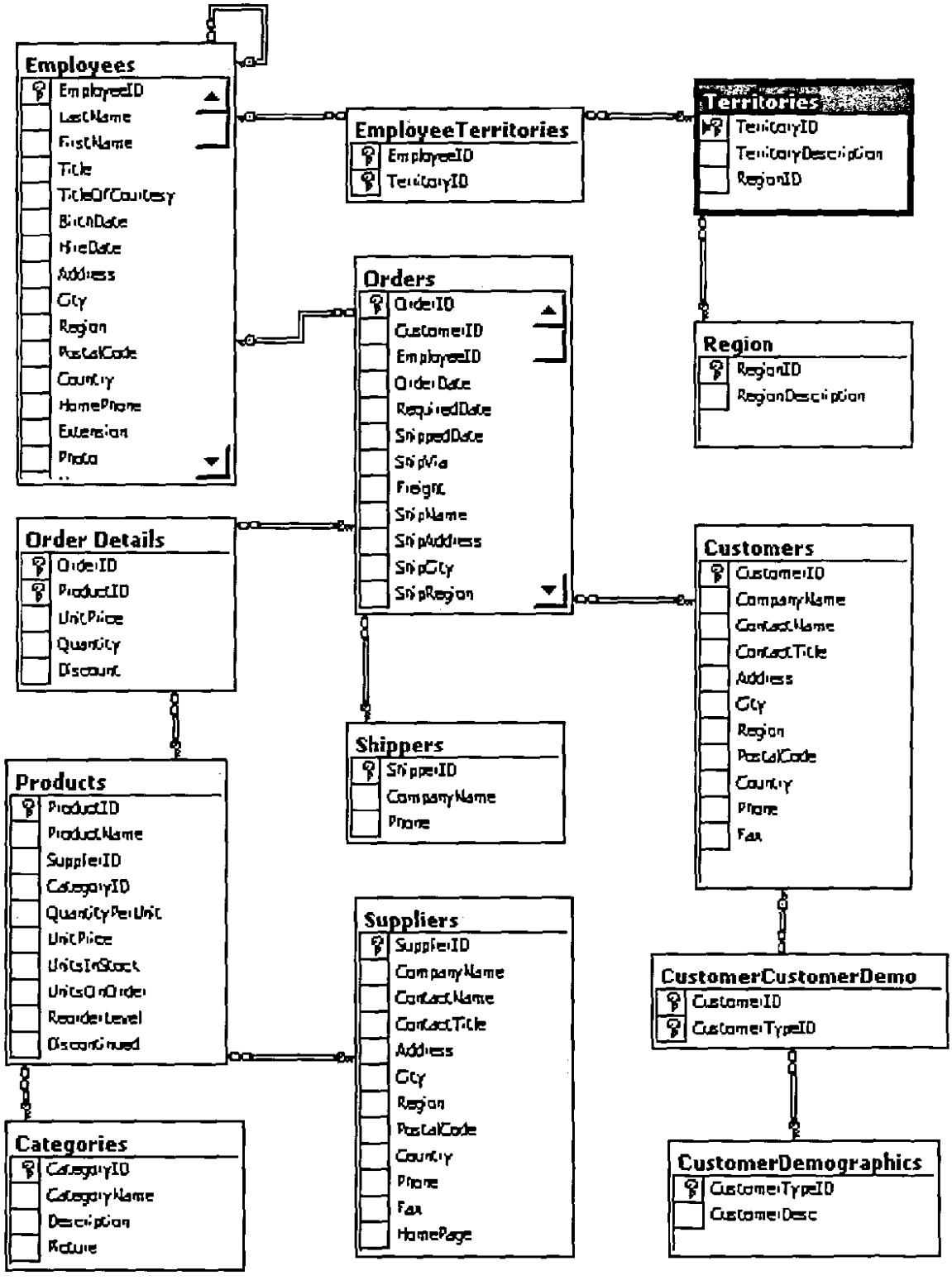


Fig 4.1: SQL Server sample database Northwind

4.3 Reference Architecture

Most common architectures are bottom-up and top-down. Some more architectures exist which are derived from the two. In bottom up architectures departmental data warehouses are designed. These departmental data warehouses are called data marts. Then these data marts are assembled into a single data warehouse, which represent the data warehouse of an organization. On the other hand in top down approach the whole organizational structure is analyzed to generate central data warehouse. Bottom up architecture is discussed in more detail in [8, 19, 20]. It follows incremental approach for design of data warehouse. Firestone at el [19] gives lots of detail for top down architecture for data warehouse design and its implementation.

The necessary resources are required for determining the factors for architecture selection of any kind of data warehouse. The cost of initial planning and design is significant when top-down architecture approach is followed. This approach is very time consuming and definitely having impact on ROI [8]. The incremental approach (bottom-up) requires less time for the initial implementation of data warehouse. As data marts are smaller as compare to data warehouse, less time is required to build them.

Bottom-up architecture is very popular in industry as it has significant advantages over the top-down approach. However top-down architecture has a single edge on bottom up architecture. That advantage is of consistency between data marts and data warehouse, a data marts are derived from the central data warehouse. Moreover it is easier to implement organizational policies in central data warehouse. On the other hand in bottom up architecture the main disadvantages are consistency and redundancy. Lots of efforts are required during combing different data marts to make central data warehouse to reduce redundancy and increase consistency. The proposed algorithm which automatically design data warehouse schema is independent of the architecture. This automation algorithm is totally dependent on input OLTP schema. If the source OLTP schema is the subset schema of organization then data marts will be created. On the other hand if entire enterprise OLTP schema is given as in input to proposed algorithm then whole OLAP schema will be created for the organization. Our algorithm is robust enough

to give output for both types of inputs. Our algorithm is totally dependent on the input logical schema of the OLTP systems. If the input is the OLTP system for the department of an organization, out put will be a data mart. If the input is the OLTP schema of whole organization then the output will be central data warehouse.

.4.4 Methodology / Algorithm

We have surveyed lot of literature in area of data warehouse design and it automation. After a comprehensive study we develop a comprehensive methodology to design this algorithm. Methodology is given by.

In this study we analyze different architectures data ware house to select the best properties of architectures by evaluation. More over the study have been conducted on different modeling techniques such as ER. Star,StarER and Snowfalke to select the best out logical schema of our proposed algorithm. Techniques which are semi automated or manual are studied to introduce best features in our algorithm. We also performed survey and evaluate techniques to design robust algorithm for automatic creation of logical data warehouse schemas.

This section focuses on the methodology of automated creation of candidate logical schemas. Our algorithm for creating candidate logical schema has five steps. The target candidate schema is star schema. The automated schema creation based upon the architecture of the existing OLTP schema e.g. table structure and their relationships. Our algorithm required that relational database schema(s).This schema must be relational schema. Having tables and relationship between each other. These are the basic requirement to for generating candidate logical schema by using five step algorithms. These steps are given below.

1. Capturing metadata information of input OLTP logical schema
2. Finding out starting points.
3. Finding out hierarchies.

4. Entities classification of OLTP logical schema.
5. OLAP logical schema generation.

4.4.1 Capturing Metadata information

In this step of algorithm all metadata information of input schema is captured in tables, used to support this algorithm. These tables are Res_entities, Res_hierarchy, Res_columnNames, Res_Factdimensions. All user created tables are stored automatically in the table Res_entities. Column Name, data types, primary keys and foreign keys are stored in the table Res_ColumnNames.

4.4.2 Finding out Starting Points

Starting points are those entities that didn't have any foreign keys. In this step algorithm check all the entities entered into table "RES_ENTITIES" to qualify for starting point. All the entities qualifying for starting point marked as "S" in the table name RES_ENTITIES.

4.4.3 Finding out Hierarchies

In this step a recursive function is designed to find the hierarchy level and reference entity at next level of each entity. All this information is kept in the table "Res_Hierarchy. By doing this stelp we are able to from top to the bottom of the relation. This step provides vital information to find our fact table.

4.4.4 Entities classification

This step classifies the OLTP source schema entities and updates the information in file "Res_Entities". F denotes the face table, D for dimension and M for many to many relations and H for hierarchy. The entity at maximum level treat as fact table and is call

hundred percent fact table. Table which are directly attached to the hundred percent fact tables and having date attribute is also qualify for fact table.

When fact tables are identified than the remaining tables which are directly attached to fact table are called dimensions. All the remaining tables are hierarchies.

4.4.5 OLAP logical schema Generation

Based on the previous steps fact tables and dimension tables are created automatically. In this step relation between data warehouse logical schema is also preserved.

Our algorithm generate OLAP logical schema by following predefined steps. All the information is captured from underlying source systems. The algorithm is given below.

4.4.6 Algorithm

AutoSchemaCreation (OLTP logical schema)

{

CaptureMetadataInfo ()

{

TableInitialization()

{

All the reserve tables are created in the source schema which are used to kept Meta data information.

}

TablePopulation()

{

All the tables of source OLTP schema are placed on one of the reserve table, created in above step.

}

ColumnPopulation()

{

All entities of source OLTP along with their attributes and data types are also stored in one of the reserve entities.

}

}

FindStartingPoints ()

{

All the entities which are not referred by any other entities or the entities didn't have any foreign key are called starting point entities.

}

FindingHierarchies (H_level)

{

//Initially hierarchy level is set to 0

This recursive function find hierarchy at each level and put them in the reserved table.

If the table referred to any other table then

FindingHierarchies (H_level+1)

Otherwise exit the procedure

}

EntityClassification ()

{

Entities are classified as fact entities, dimension entities, entities forming many to many relationship and the entities forming hierarchical dimensions.

}

SchemaGenration ()

{

This procedure generate star schema automatically from the information available in system reserve tables. The relationships between OLAP logical schema entities are automatically established.

}

}

4.5 Summary

We discuss the methodology and design architectures required for our algorithm. The main architectures are top down and bottom up. Our algorithm is source driven and is independent of these architectures. OLTP logical schema is used as input to the algorithm to generate candidate logical schemas for OLAP. We illustrate out algorithm in this chapter.

5. Implementation

5. Implementation

5.1 Introduction

We implement our robust algorithm by using Visual Basic 6 and SQL Server 2000. This chapter includes pseudo code of the algorithm and its implementation.

Moreover in this chapter full flow of the project is given by showing pictorial diagrams. To test of implementation we use different OLTP schemas. But in this thesis we sill illustrate our implementation by using SQL Server 2000 sample database NORTHWIND.

5.2 Deployment Environment

Our algorithm could be tested on simple machines having SQL Server 2000 and any sample OLTP schema. Algorithm is test on four different sample OLTP schemas to get consistency in results.

5.2.1 Tool/Language Selection

The choice of software is very important factor to be considered during the development phase of a new system. This decision depends upon many factors including the requirement of the system, current environment (i.e. existing software), amount of data to handle and the cost of programming. After deeply studying the nature of the problem and considering the need, we choose visual basic 6 for front end and SQL Server 2000 as back end.

5.3 System Flow Diagram

Our proposed algorithm comprises of five steps. In five steps this algorithm generate OLAP logical schema from OLTP logical schema.

5.3.1 First Step (Capturing Metadata information)

1. In this step reserved tables are created in the source OLTP schema which is used to capture metadata information from source OLTP logical schema. These reserve tables are shown in the diagram below.

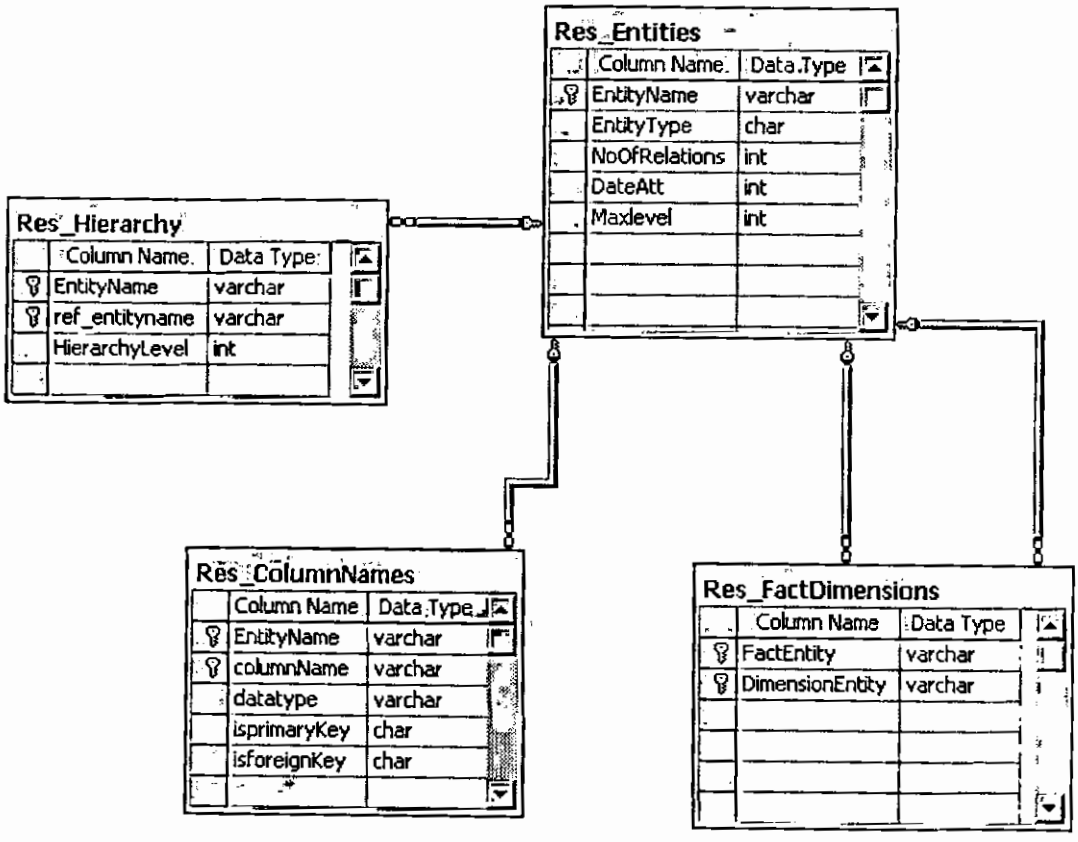


Fig 5.1 Reserve Tables

There are four reserve tables Res_Entities, Res_Hierarchy, Res_ColumnName and Res_FactDimensions. We also create a view called res_manytomany We will explain the reserve entities and view one by one.

- Entity Res_Entities structure in shown in the above diagram. Field EntityName is used to store the entity names of the source OLTP logical schema. Entitytype attribute store the information about the entity e.g. Is

entity is starting point or fact table or dimension etc. NoofRelations attribute kept the information about the individual entities relationship depth with other entities. Maxlevel attribute keeps information the depth of relation of each entity.

- Entity Res_Hierarchy used to keep the information about the hierarchy of each entity and its level of hierarchy. Hierarchyleval attribute keep the information of hierarchy level and ref_entityname keeps the information of referenced entity of their master tables.
- Reserve Entity Res_Columnnames is very important as it keeps the information of all attributes of all entity. It also preserver information it the attribute is primary key or foreign key.
- Entity Res_FACTDIMENSIONS keeps the information of fact tables and their dimensions.
- A reserved view "RES_MANYTOMANY" is created to find entities which form many to many relations in an input OLTP schema. The design of input schema is given in figure below.
-

```
CREATE VIEW RES_MANYTOMANY
AS
SELECT  REF_ENTITYNAME
FROM    (SELECT  REF_ENTITYNAME, COUNT(REF_ENTITYNAME) AS T1
        FROM (SELECT REF_ENTITYNAME, HIERARCHYLEVEL, COUNT(REF_ENTITYNAME) NOOFRELATION
        FROM RES_HIERARCHY
        WHERE HIERARCHYLEVEL <> 0
        GROUP BY REF_ENTITYNAME, HIERARCHYLEVEL) AS A
      GROUP BY REF_ENTITYNAME) D
WHERE     (T1 > 1)
```

Fig 5.2 RES_MANYTOMANY View

	ref_entityname
1	EMPLOYEETERRITORIES

Fig 5.3 RES_MANYTOMANY View Result

2. All user created tables in source OLTP schema populated into table name "RES_ENTITIES" as shown below.

	EntityName	EntityType	NoOfRelations	DateAtt	Maxlevel
1	CATEGORIES		0	0	0
2	CUSTOMERCUSTOMERDEMO		0	0	0
3	CUSTOMERDEMOGRAPHICS		0	0	0
4	CUSTOMERS		0	0	0
5	EMPLOYEES		0	0	0
6	EMPLOYEETERRITORIES		0	0	0
7	ORDER DETAILS		0	0	0
8	ORDERS		0	0	0
9	PRODUCTS		0	0	0
10	REGION		0	0	0
11	SHIPPERS		0	0	0
12	SUPPLIERS		0	0	0
13	TERRITORIES		0	0	0

Fig 5.4 Entity RES_ENTITIES

3. Field names and their data types are populated in the table "res_columnnames". It is also marked if some field is primary key or foreign key. This very important entity which plays and important role in automatic logical schema generation.

	EntityName	columnName	datatype	isprimaryKey	isforeignKey
1	CATEGORIES	CATEGORYID	NUMERIC	Y	N
2	CATEGORIES	CATEGORYNAME	VARCHAR	N	N
3	CATEGORIES	DESCRIPTION	VARCHAR	N	N
4	CATEGORIES	PICTURE	BINARY	N	N
5	CUSTOMERCUSTOMERDEMO	CUSTOMERID	VARCHAR	Y	Y
6	CUSTOMERCUSTOMERDEMO	CUSTOMERTYPEID	VARCHAR	Y	Y
7	CUSTOMERDEMOGRAPHICS	CUSTOMERDESC	VARCHAR	N	N
8	CUSTOMERDEMOGRAPHICS	CUSTOMERTYPEID	VARCHAR	Y	N
9	CUSTOMERS	ADDRESS	VARCHAR	N	N
10	CUSTOMERS	CITY	VARCHAR	N	N
11	CUSTOMERS	COMPANYNAME	VARCHAR	N	N
12	CUSTOMERS	CONTACTNAME	VARCHAR	N	N
13	CUSTOMERS	CONTACTTITLE	VARCHAR	N	N
14	CUSTOMERS	COUNTRY	VARCHAR	N	N
15	CUSTOMERS	CUSTOMERID	VARCHAR	Y	N
16	CUSTOMERS	FAX	VARCHAR	N	N
17	CUSTOMERS	PHONE	VARCHAR	N	N
18	CUSTOMERS	POSTALCODE	VARCHAR	N	N
19	CUSTOMERS	REGION	VARCHAR	N	N
20	EMPLOYEES	ADDRESS	VARCHAR	N	N
21	EMPLOYEES	BIRTHDATE	DATE	N	N
22	EMPLOYEES	CITY	VARCHAR	N	N
23	EMPLOYEES	COUNTRY	VARCHAR	N	N
24	EMPLOYEES	EMPLOYEEID	NUMERIC	Y	N
25	EMPLOYEES	EXTENSION	VARCHAR	N	N

Fig 5.6 data populated in entity res_columnnames

5.3.2 Second Step (Finding out starting points)

In this step algorithm check all the entities entered into table "RES_ENTITIES" to qualify for starting point. Starting points are those entities that didn't have any foreign key. All the entities qualifying for starting point marked as "S" in the table name "RES_ENTITIES" as shown below in the diagram.

	EntityName	EntityType	NoOfRelations	DateAtt	Maxlevel
1	CATEGORIES	S	0	0	0
2	CUSTOMERCUSTOMERDEMO		0	0	0
3	CUSTOMERDEMOGRAPHICS	S	0	0	0
4	CUSTOMERS	S	0	0	0
5	EMPLOYEES	S	0	0	0
6	EMPLOYEETERRITORIES		0	0	0
7	ORDER DETAILS		0	0	0
8	ORDERS		0	0	0
9	PRODUCTS		0	0	0
10	REGION	S	0	0	0
11	SHIPPERS	S	0	0	0
12	SUPPLIERS	S	0	0	0
13	TERRITORIES		0	0	0

Fig 5.7 showing marked starting points

These starting points are also entered into the table name "Res_Hierarchy" as shown in the figure below. As all these tables are master tables and will be referred by some other entities present in the source input schema.

	EntityName	ref_entityname	HierarchyLevel
1	CATEGORIES		0
2	CUSTOMERDEMOGRAPHICS		0
3	CUSTOMERS		0
4	EMPLOYEES		0
5	REGION		0
6	SHIPPERS		0
7	SUPPLIERS		0

Fig 5.8 showing Starting points entered in entity Res_Hierarchy

5.3.3 Third Step (Finding out Hierarchies)

In this step a recursive function is designed to find the hierarchy level and reference entity at next level of each entity. All this information is kept in the table

"Res_Hierarchy" as shown below. Information captured in this table is important to find out fact table dimensions and dimension hierarchy.

	EntityName	ref_entityname	HierarchyLevel
1	CATEGORIES		0
2	CUSTOMERDEMOGRAPHICS		0
3	EMPLOYEES		0
4	CUSTOMERS		0
5	REGION		0
6	SHIPPERS		0
7	SUPPLIERS		0
8	SUPPLIERS	PRODUCTS	1
9	SHIPPERS	ORDERS	1
10	REGION	TERRITORIES	1
11	CUSTOMERS	CUSTOMERCUSTOMERDEMO	1
12	CUSTOMERS	ORDERS	1
13	EMPLOYEES	EMPLOYEETERRITORIES	1
14	EMPLOYEES	ORDERS	1
15	CUSTOMERDEMOGRAPHICS	CUSTOMERCUSTOMERDEMO	1
16	CATEGORIES	PRODUCTS	1
17	ORDERS	ORDER DETAILS	2
18	PRODUCTS	ORDER DETAILS	2
19	TERRITORIES	EMPLOYEETERRITORIES	2

Fig 5.9 showing referenced hierarchies and levels

5.3.4 Fourth Step (Entities Classification)

This step classifies the OLTP source schema entities and updates the information in file "Res_Entities". F denotes the fact table, D for dimension and M for many to many relations and H for hierarchy. A query is designed to find the candidate fact tables are show in the figure below.

```
SELECT REF_ENTITYNAME,HIERARCHYLEVEL,COUNT(REF_ENTITYNAME) NOOFRELATION
FROM RES_HIERARCHY
WHERE HIERARCHYLEVEL <> 0 AND REF_ENTITYNAME NOT IN
(SELECT * FROM RES_MANYTOMANY)
GROUP BY REF_ENTITYNAME,HIERARCHYLEVEL ORDER BY HIERARCHYLEVEL
```

Fig 5.10 Query to find relations

	ref_entityname	HierarchyLevel	noofRelation
1	CUSTOMERCUSTOMERDEMO	1	2
2	ORDERS	1	3
3	PRODUCTS	1	2
4	TERRITORIES	1	1
5	ORDER DETAILS	2	2

Fig 5.11 showing candidate fact entities

In this thesis we also give the idea of finding hundred percent fact tables. The entities having maximum relations treat as fact table and are called hundred percent fact tables. Table which are directly attached to the hundred percent fact tables and having date attribute is also qualify for fact table. In the Northwind database ORDERS table is the hundred percent fact tables, and it is having maximum relations as shown in the figure above. Entities which are directly attached to the fact table are called dimensions. Our algorithm tagged each entity. The remaining entities are called hierarchy entities. At the end of this step all the entities in the source OLTP schema will be tagged with the procedures defined in our algorithm. The tagged entities are show in the figure below.

	entityname	entitytype
1	CATEGORIES	S
2	CUSTOMERCUSTOMERDEMO	M
3	CUSTOMERDEMOGRAPHICS	S
4	CUSTOMERS	D
5	EMPLOYEES	D
6	EMPLOYEETERRITORIES	M
7	ORDER DETAILS	F
8	ORDERS	F
9	PRODUCTS	D
10	REGION	S
11	SHIPPERS	D
12	SUPPLIERS	S
13	TERRITORIES	H

Fig 5.12 Tagged entities of Northwind schema

5.3.5 Fifth Step (Schema Generation)

This step uses information gathered in previous steps to generated OLAP schema of the source OLTP schema. In this step algorithm reserve table "res_factdimensions" is populated with fact tables and their corresponding dimensions. These dimensions and fact tables are already classified in the fourth step of algorithm. The content of reserve table res_factdimensions is shown in the figure below.

	FactEntity	DimensionEntity
1	ORDER DETAILS	CUSTOMERS
2	ORDER DETAILS	EMPLOYEES
3	ORDER DETAILS	PRODUCTS
4	ORDER DETAILS	SHIPPERS
5	ORDERS	CUSTOMERS
6	ORDERS	EMPLOYEES
7	ORDERS	SHIPPERS

Fig 5.13 content of res_factdimensions

After the population of res_factdimensions table algorithm uses reserve tables "res_columnnames" and res_factdimensions to generate all fact tables and dimensions. All the field names and corresponding data types of the fields are managed efficiently by our algorithm. In this step of algorithm all the relationship between dimensions and fact tables are automatically made.

5.4 Algorithm Pseudo codes

Pseudo code of the algorithm is give below,

```
AutoSchemaCreation (OLTP logical schema)
{
CaptureMetadataInfo ()
{
    TableInitialization()
        {
        //All the reserve tables are created in the source schema which are used to
        //kept Meta data information.
```

```
        If reserve tables (table names) exists
          Drop tables
        End if
        Create Reserve tables in source schema

        If reserve Views (view names) exists
          Drop Views
        End if
        Create Reserve Views in source schema
        }
TablePopulation()
        {
        //All the tables of source OLTP schema are placed on one of the reserve
        //table, created in above step.
        Dim AllTables as table
        // AllTables is variable having all user created tables excluding reserve
        //tables.
        While AllTables traversed
                Insert into RES_ENTITIES (EntityName) values(tablename)
        Next table
        Wend
        }
ColumnPopulation()
        {
        //All entities of source OLTP along with their attributes and data types are
        //also stored in one of the reserve entities.
        Dim Alltables as table
        Dim Attributenames as array
        While AllTables traversed
                {
                        For all attributes in the table
```

```
                                Datatype=finddatatype(attribute)
                                if isprimarykey(attribute)
                                  Primarykey=true
                                 Else
                                   Primarykey=False
                                End if

                                if isForeignkey(attribute)
                                  Foreignkey=true
                                 Else
                                   Foreignkey=False
                                End if
                                Insert into Res_ColumnNames
                               (EntityName,columnName
                                datatype,isprimaryKey,isforeignKey) values
                                (table,attribute,Datatype,PrimaryKey,Foreignkey)
                              Next attribute
                        }
                Next table
                Wend
                }
}
FindStartingPoints ()
{
        //All the entities which are not referred by any other entities or the entities didn't
        //have any foreign key are called starting point entities.
        Dim IsStart as boolean
        For all entities in the OLTP schema
                IsStart=true
                For all entities used a foreign key
                        If foreingkeyentity=oltpentity then
```

```
                    IsStart=false
                    Exit for
                    End if
                Next entity

        If  IsStart=True then
        // Here S denotes the starting point
        UPDATE RES_ENTITIES set EntityType = 'S' where entityname=OLTPentity
        End if
        Next entity

}
FindingHierarchies (H_level)
{
        //Initially hierarchy level is set to 0
        //This recursive function find hierarchy at each level and put them in the reserved
        //table.
        // initial value of H_level is 0
        If  H_level=0 then
                SQL=SELECT DISTINCT EntityName FROM RES_HIERARCHY
                WHERE HIERARCHYLEVEL= H_level
        Else
                SQL=SELECT DISTINCT ref_entityname FROM RES_HIERARCHY
                WHERE HIERARCHYLEVEL= H_level
        End if
        Recordset=Getrecorset(SQL)
        If no record in Recorset then
                End function
        End if
        For All records in the Recordset
                For All Foreign Keys in OLTP Schema
```

```
                    If  Recordset entity refered the foreign key entity
                    INSERT INTO RES_HIERARCHY
                    (ENTITYNAME,REF_ENTITYNAME,HIERARCHYLEVEL)
                    VALUES(entityname,refrenceeity, H_level)
                    End if
            Next Foreign Key entity
        Next Record
        FindingHierarchies (H_level+1)
}
EntityClassification ()
{
        // Entities are classified as fact entities, dimension entities, entities forming many
        // to many relationship and the entities forming hierarchical dimensions.
        For all entities in view res_manytomany
                Update Res_Entities set entitytype='M' for defined entity
        Next entity
        // 100 % fact table
        For all entities having maximum hierarchy level
                UPDATE Res_Entities SET entitytype='F' for defined entity
        Next entity
        // Another fact entity which is attached directly with 100% fact and having
        //attribute with Date data type will qualify for fact table

        For all entities having date attribute and directly attached with fact table
                UPDATE Res_Entities SET entitytype='F' for defined entity
        Next entity

        // Entities directly attached to fact table will be dimensions.
        For all entities directly attached with fact table
                UPDATE Res_Entities SET entitytype='D' for defined entity
        Next entity
```

```
        // All the remaining unmarked entities will be hierarchy
        For all unmarked entities
                UPDATE Res_Entities SET entitytype='H' for defined entity
        Next entity
}
SchemaGenration ()
{
        //This procedure generate star schema automatically from the information
        //available in system reserve tables. The relationships between OLAP logical
        //schema entities are automatically established.

        For all classified entities
                If classified entity is fact table
                        Create fact table
                Else if classified entity is dimension table
                        Create dimensional table
                End if
        Next entity
        Create relationship between all
        Fact tables and dimensions
}

}
```

5.5 Summary

This chapter focuses on the implementation phase of the research. It is lists the tools used to develop the proposed system. The system flow diagram shows the flow of data at each step and information contain in it. Then pseudo code and algorithm of the system is also defined in detail. Finally it contains the list of classes and method involved in the implementation. The next chapter discusses the testing and performance evaluation of the proposed system.

6. Testing and Performance Evaluation

Testing and Performance Evaluation

6.1 Introduction

To test and evaluate our robust algorithm we took different source OLTP schemas as an input. We will explain our test scenarios in the next section on four different source OLTP schemas. These sample schemas are NorthWind, Salesdb, purchasedb and Banking OLTP schemas. We didn't care about the user needs during data warehouse logical schema creation. Our algorithm generating automatically candidate star schema from input OLTP logical schema. Our algorithm only analyzes the information available in the source schema to generate candidate star schema. Slightly manual refinement is required to improve the star schema to full fill the business need of the user. Manual refinement of the automatically generated star schema is explained in section 6.3. Our proposed algorithm gives accurate results on three schemas but deviate in fourth one. This is limitation of this algorithm. This limitation is explained in section 6.2.4 and in next chapter also.

6.2 Test Scenario

The automatic OLAP schema generating algorithm is tested on four OLTP schemas. All of these schemas have different logical structures structures.

- Northwind
- Salesdb
- Purchasedb
- Banksmart

We will explain and show our results on these four OLTP schemas one by one.

6.2.1 Northwind schema and Resulting Star Schema

Northwind source OLTP schema is shown in the figure 4.1 in fourth chapter. After applying our proposed algorithm it generate candidate star schemas which are shown in the figure 6.1

We can see in the figure 6.1 there is no date dimension. User can add this dimension table in the schema manually. May be some information is missing in the generated start schema. This information could be easily added or modified as per requirement of user.

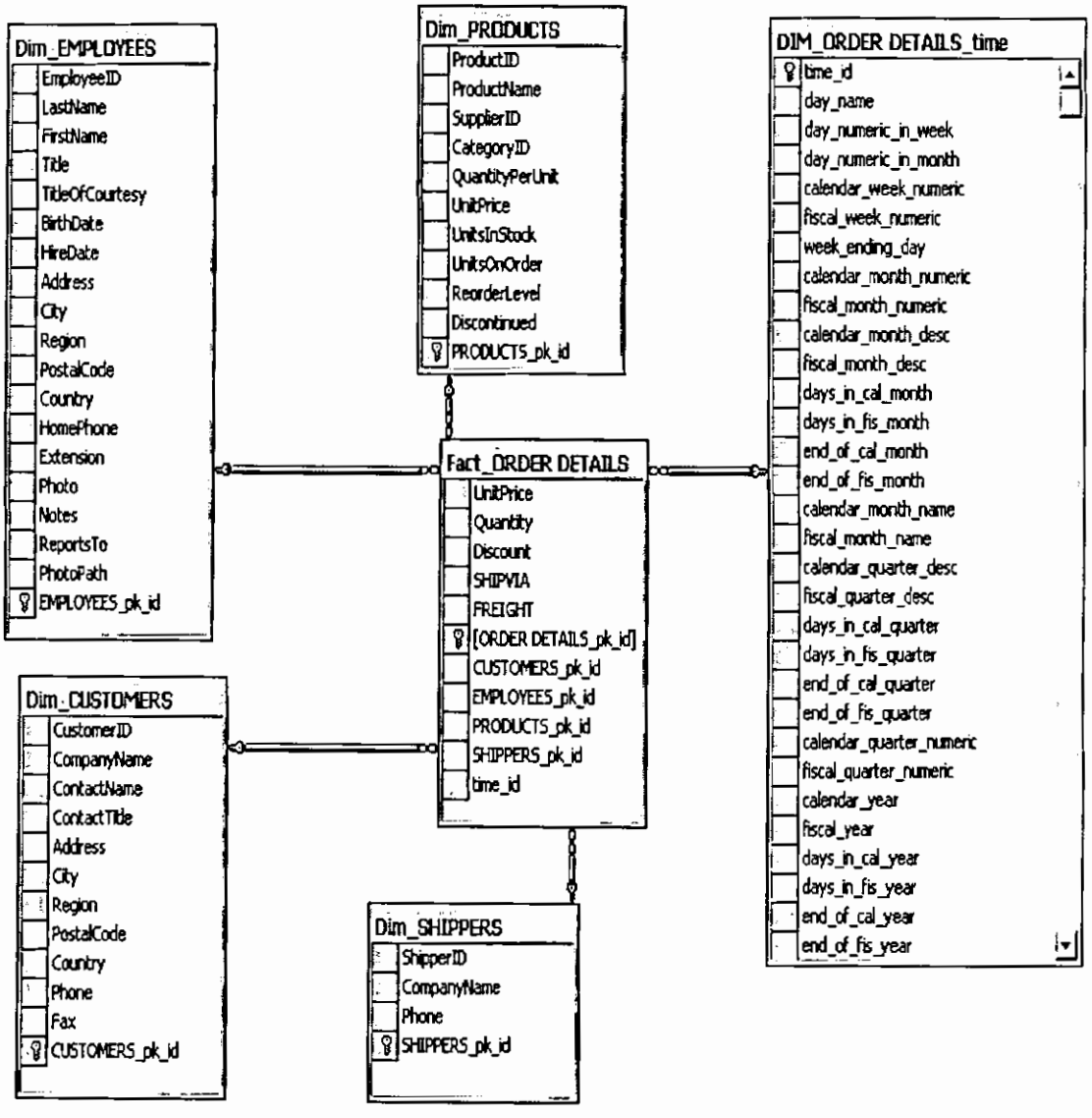


Fig 6.1: Star schema generating from Northwind OLTP Schema

6.2.2 Salesdb Schema and Resulting Star Schema

We also run our algorithm on Salesdb OLTP Schema to test the reliability, consistency and efficiency of the algorithm. Salesdb OLTP Schema is shown in the figure 6.2.Relational databases are the bread and butter for the design of data warehouses. In source driven approach lots of characteristic of the source system come into the resultant

data warehouse. We can say that OLTP systems are driving force to generate the concept of data warehouse. ER diagram of OLTP schema which one of the input to our algorithm for testing and evaluation purpose is shown in fig 6.2.

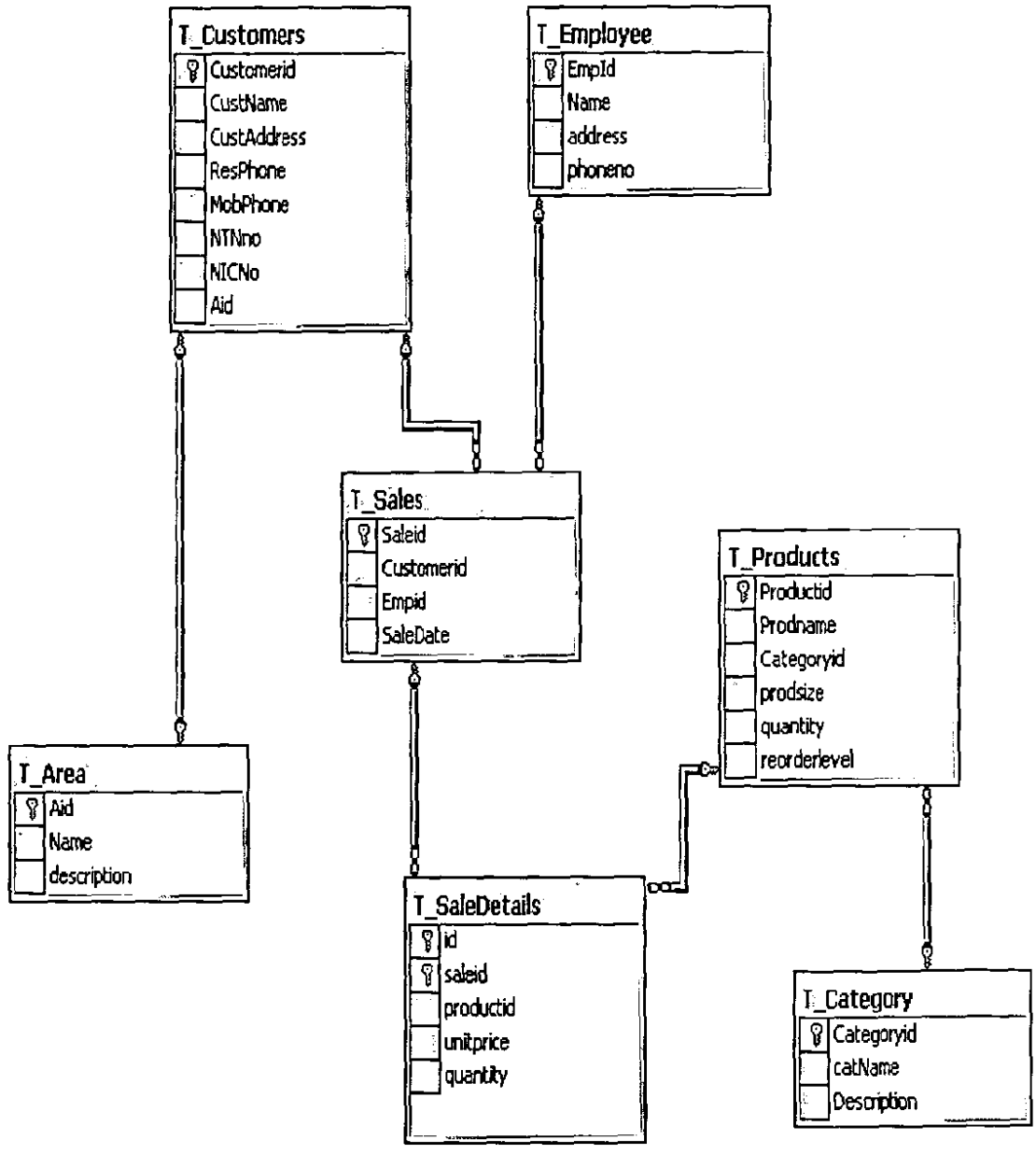


Fig 6.2: Salesdb OLTP Schema

Fig 6.3 is the result of OLTP schema mentioned in figure 6.3. The generated schema is the star schema. Star schema is very useful modeling technique for data warehouse. It is suitable where performance is the objective.

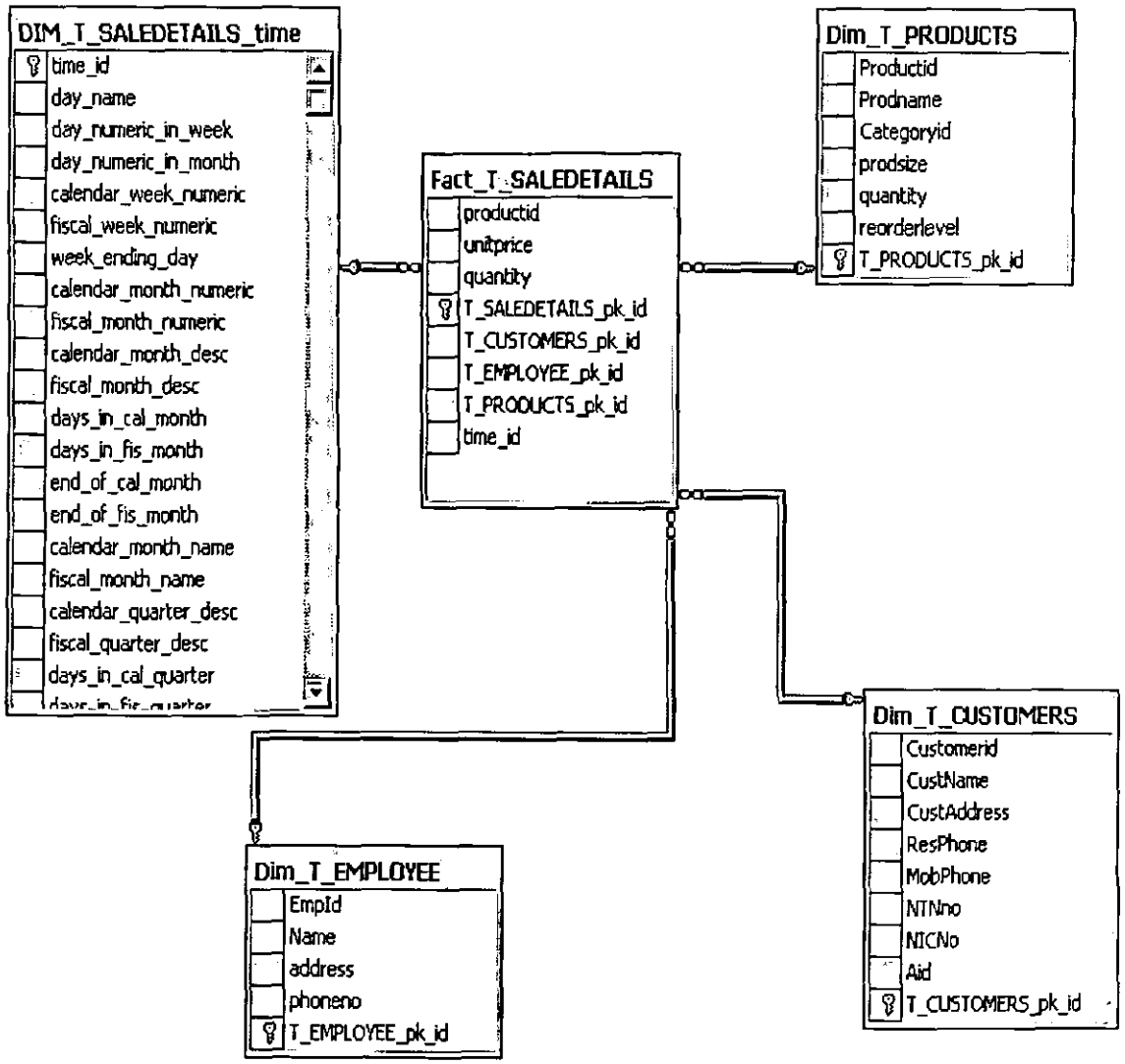


Fig 6.3: Star schema generating from Salesdb OLTP Schema

6.2.3 Purchasedb Schema and Resulting Star Schema

We also run our algorithm on Purchasedb OLTP Schema to test the reliability, consistency and efficiency of the algorithm. Purchasedb OLTP Schema is shown in the figure 6.4. Relational databases are the bread and butter for the design of data warehouses. In source driven approach lots of characteristic of the source system come

into the resultant data warehouse. We can say that OLTP systems are driving force to generate the concept of data warehouse. ER diagram of OLTP schema which one of the input to our algorithm for testing and evaluation purpose is shown in fig 6.4.

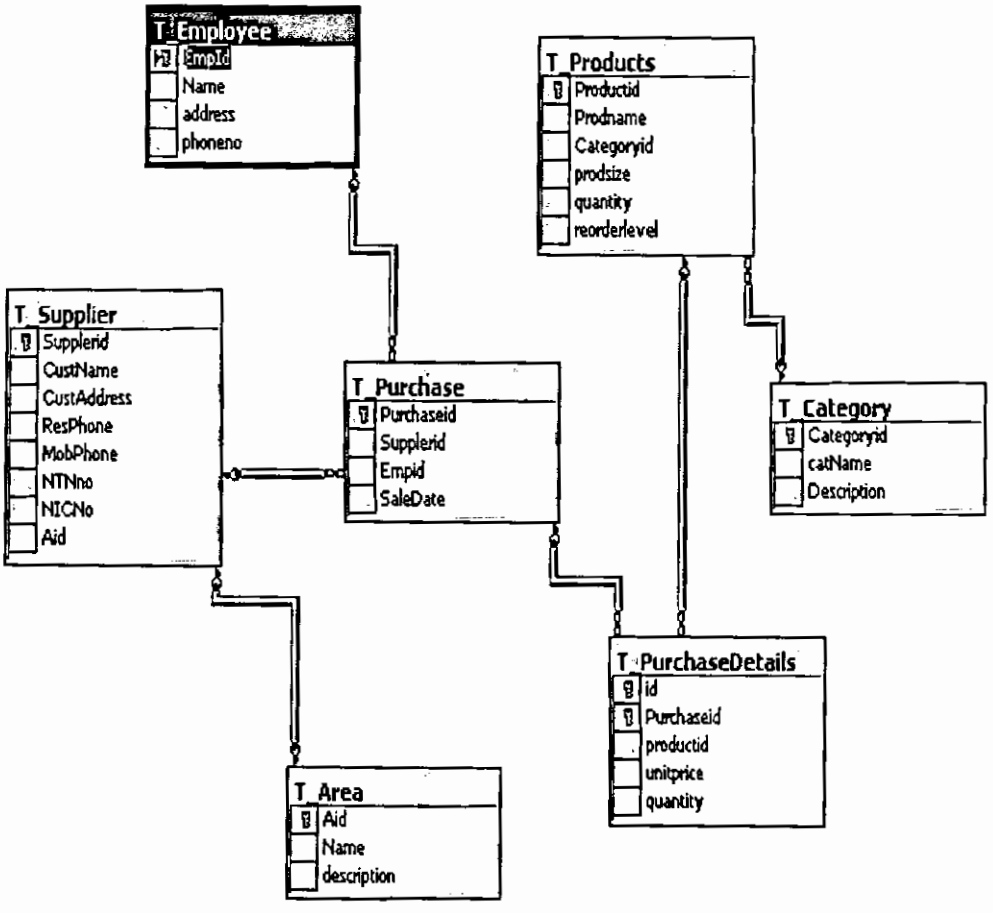


Fig 6.4: ER diagram for OLTP Schema

Star schema is considered as the main and basic step to enter into the world of dimensional modeling. In start schema the large table exists in the center having facts of the part or full business object called the fact table. This fact table is surrounded by some objects called the dimensions. In this model the query processing speed is fastest because there are less joins in star schema. As the number of dimension tables are reduced by collapsing the dimensions lying in the hierarchy as discussed in previous chapters. There

could be any number of dimensions in star model. The star model generated from the sample OLTP model is shown in the figure below.

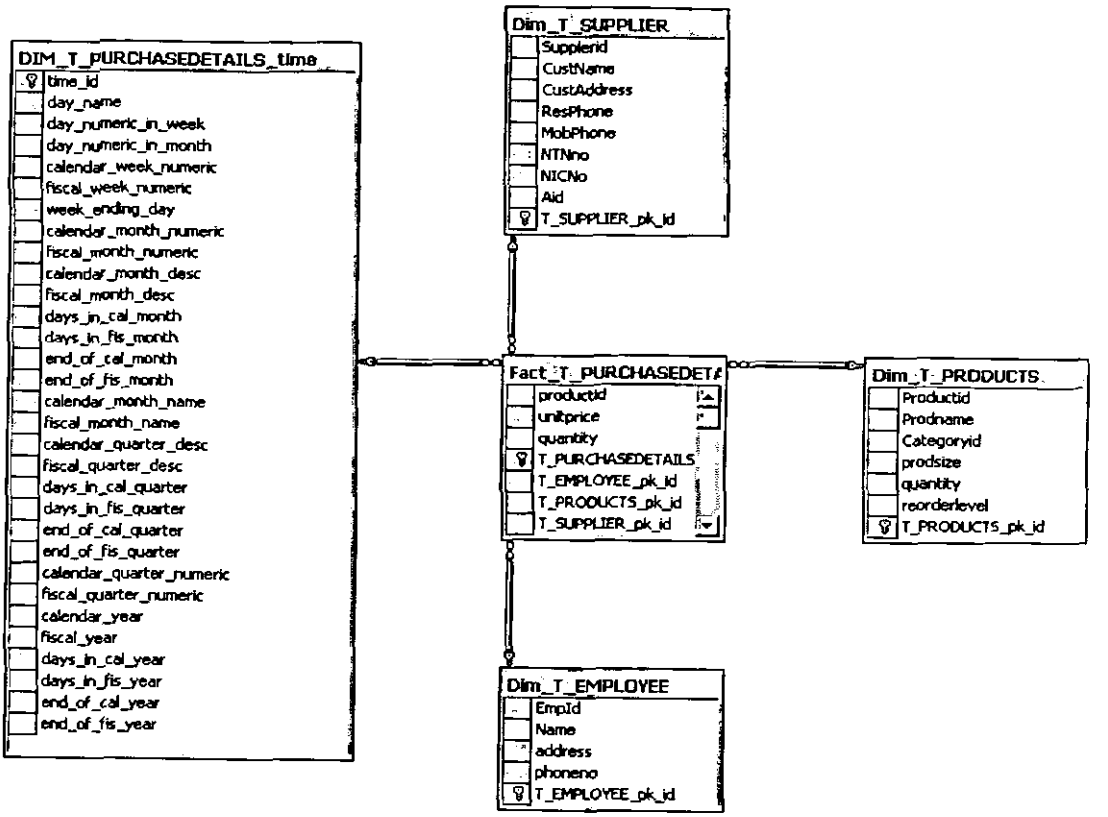


Fig 6.5: Star schema generating from Purchasedb OLTP Schema

6.2.3 BankSmart Schema and Resulting Output

We have discussed the limitation of our algorithm. Our algorithm did not produce accurate result when some measures and stored in non numeric data type. For example in this input schema as shown in fig 6.6 the file t_products have attributes minCreditinteres and minDebitinterest in character field. These fields are important measures and should be part of the fact table. But our algorithm ignores these attributed while constructing the target OLAP Schema.

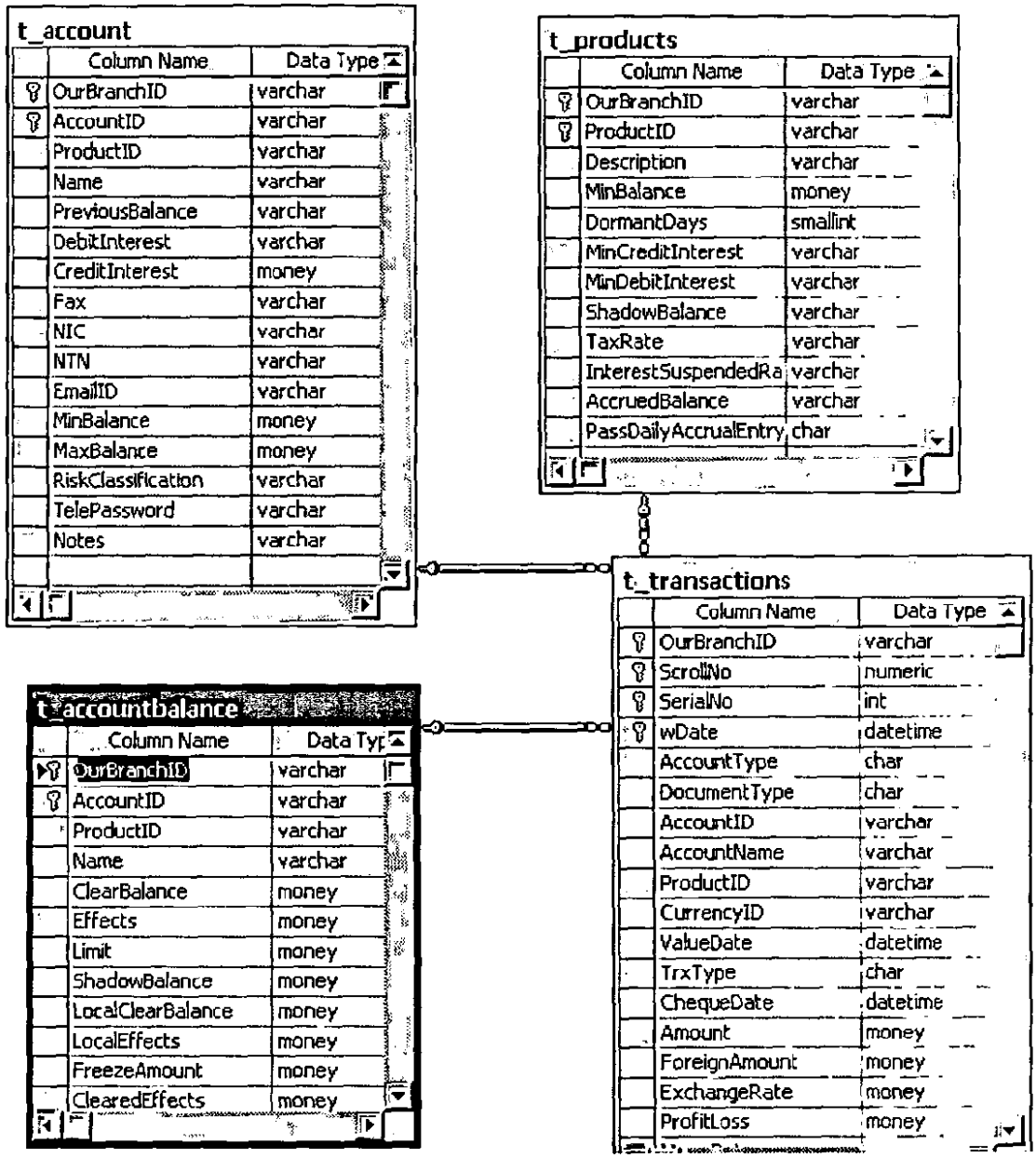


Fig 6.6: ER diagram for OLTP Schema banksmart

The limitation of our algorithm could be analyzed by keeping view the fig 6.6 and 6.7. All the numeric attributes are transferred to the fact table fact_t_transactions. While non numeric fields remaincd in the dimensions tables' dim_t_products, dim_t_accounts. It is our future work to handle this kind of limitation.

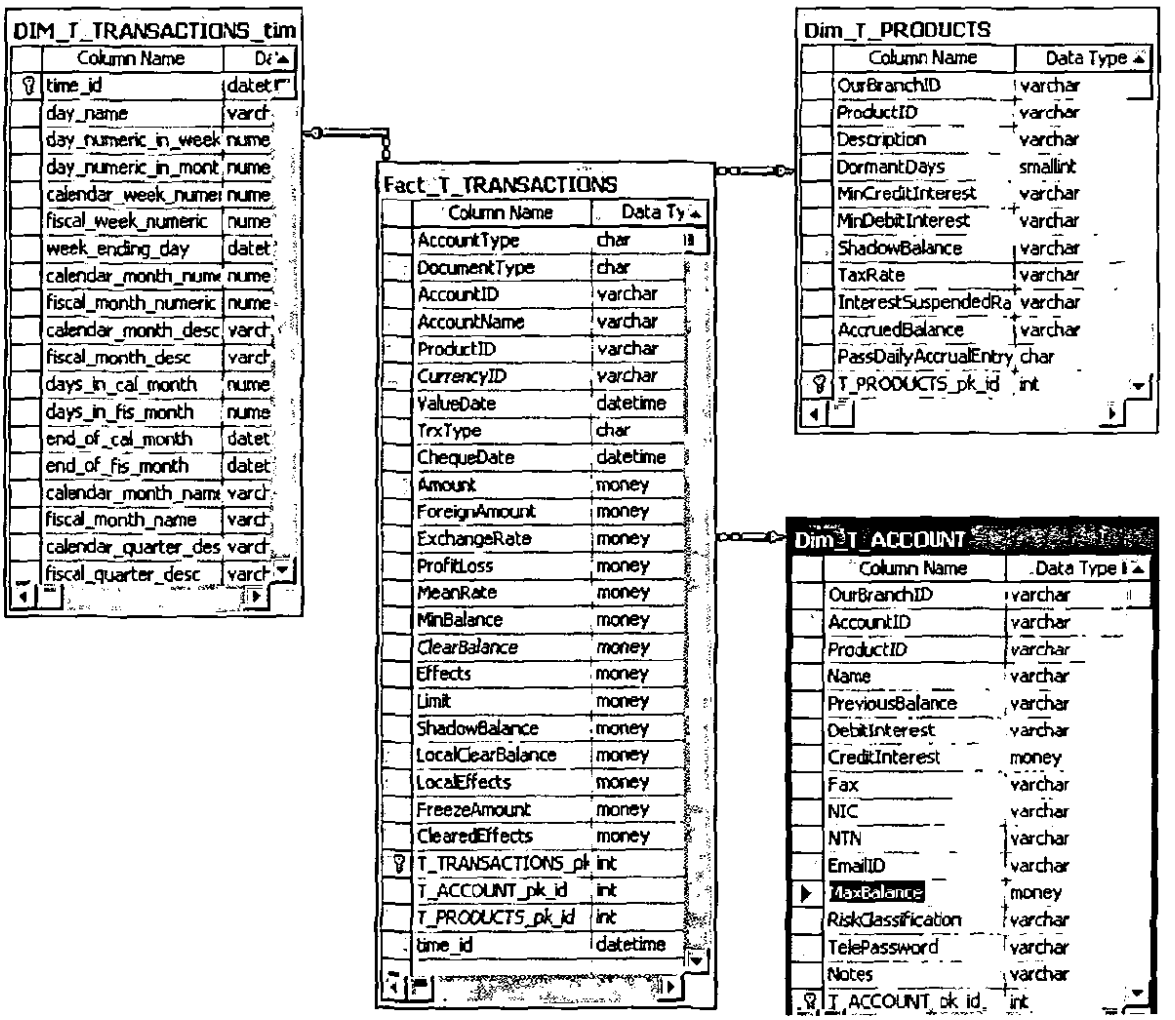


Fig 6.7: Star schema generating from Banksmart OLTP Schema

6.3 Performance and Evaluation

By implementing the proposed technique the system becomes

- Efficient
- Having Quick Response
- Flexible

6.3.1 Efficient

Our proposed algorithm requires less manual refinement as compared to other algorithms for the resulted OLAP schema. Moreover approach followed by phips & devis generated

semi automated approach for logical schema generation. In this approach first all ME/R model was created for conceptual design , then all these ME/R model were traversed to find the fact table. On the other hand our algorithm takes OLTP schema and finds directly fact tables. Our algorithm didn't require any conceptual model. Our algorithm decide fact table on the basis of hierarchy level of entities and thus reduce the recursive traversing. This algorithm out class the phips & devis algorithm by reducing conversion time.

6.3.2 Having Quick Response

Our five step algorithm is efficient as compare to other algorithm for automatic schema transformation from OLTP systems to OLAP. As discussed in previous section Phips and Devis algorithm check iteratively every candidate conceptual schema to find the fact table. On the other hand our algorithm finds the fact table on predefined hierarchies. So the response time to get the results increases.

6.3.3 Flexible

Our proposed algorithm is flexible enough to convert any kind of relational OLTP schema to OLAP. This algorithm is flexible enough to generate logical schema for date mart or for whole data warehouse.

6.3.4 Refinement of Logical Schema

Fully automation of data ware house design according to the requirement of user is very difficult. Our algorithm is fully automated as per information available in OLTP schema. But it is open to make refine the candidate schemas according to the user requirements. Any designer of data warehouse requires little bit know how of the existing operational system databases to make amendments according to the need of organization. Manual refinements are necessary in the generated schema to full fill all the user needs and requirements. In this thesis we introduce the manual task to full fill the user needs. Other wise the generated schema is complete. It is near to impossible to automate all kind of user requirement in source driven automated mechanism. How ever our algorithm is generating Star schema, user may have some different requirements. Which may be full

filled by the Snowflake schema? As our output schema is Logical schema which may be difficult for the users in perspective of understandability. It is possible to generated conceptual schema from logical schema by reverse engineering. The after taking input from the user convert it back to logical schema that could be used by any kind of data warehouse systems. It is recommended that the designer input is valuable for further refinement and enhancement of the system. After these refinements one could not blame the design in perspective of completeness. We have given six manual steps for logical schema refinement.

1. The in depth analysis of fact table to handle the missing value and additional attributes.
2. The granularity by keeping in view each dimension.
3. Maintenance of the desired aggregation level.
4. Is there any possibility to combine more than one fact tables into one.
5. Dimension table pruning.
6. Additional data requirement which ware not available in the source systems.

The above mentioned steps are quite comprehensive for the manual refinement of the generated candidate logical schema. How ever designer can make further changes based on his experience and requirement made by any organization.

6.4 Summary

This chapter focuses on the testing and performance evaluation of the proposed algorithm. First it briefly discussed the working environment of the proposed system. Then test scenario is mentioned and at the end results of previous system and proposed system is compared and theses. In the next chapter we give the concluding remarks and give some future recommendations.

7. Conclusion and Outlook

Conclusion and Outlook

7.1 Introduction

Our proposed algorithm is robust enough to handle all kind of relational data base. But we simulate it only on SQL Server 2000, the tool available in the market. However the design of algorithm is stable as we tested it on the schema having different levels of relationship between entities and different size of entities and schemas. There is no doubt in its durability and robustness. For the conversion of schemas is not dependent on conceptual models of exiting OLTP systems. But it is dependent on the logical structure of underlying source systems. More over it is not dependent on the physical structures. There is no special requirement of hardware to run our proposed algorithm. The hardware requirement suitable of OLTP systems desired to convert is also suitable for our algorithm. The major contribution of the approach is the autonomous. By this automation the designers' time is tremendously reduced required to generated data warehouse. It is our claim that our algorithm full fills most of the requirements of the user. How ever for hundred percent out put, designer's refinement are necessary However it designer did not do any manual refinement, the generated logical structure fulfill the basic requirement of any data warehouse. The target out put is the star schema that is efficient in query processing point of view.

7.2 Achievements

This research work is a direction towards algorithmic approach for automatic conversion of existing OLTP systems to OLAP. By this work it will be quite easy for novice data warehouse developer to understand and develop data warehouse logical schemas. It is our achievement that we will be able to give an approach to convert complex OLTP systems to OLAP. That is ultimately very helpful to the community to generated complex logical structure from the existing OLTP system by a single click.

We have contributed the data ware house community by evaluating the exiting architectures of data warehouse. And discuss the merit and demerits the architecture.

Moreover our algorithm is independent of the architecture. Both approaches could be followed.

Another contributing is the automated approach which converts exiting OLTP schemas to OLAP logical schema.

And the last contributing is that we give the steps for manual refinement.

7.3 Improvements

We can improve our algorithm by adding one more step prior to these five steps mentioned in our algorithm. This step is schema refinement step. We can refine the source OLTP schema in multiple way. For example our proposed algorithm uses numeric data type with some other measure to find hundred percent fact tables. In source OLTP system there may be a character data type having numeric data. We have to find these fields in source system and convert them in some numeric data type. So that our algorithm will able give accurate results.

7.4 Future Recommendations/Outlook

There is very little work done for the automation of the data warehouse. However it is very tedious job to fully automate the data warehouse according to business flow of an organization. How ever our work is good enough to automate the concept of data warehouse schema creation. How ever still few dimensions exist that require some attention. The problem area is the measures. We assume that all the measure will be available in numeric form. But some designers follow practice that used keep numeric data in textual field. Our automated algorithm will miss such kind of measures. That may lead to incorrect design of the out put logical schema. On the other hand some attributes are numeric in nature but these attributes are not measure of fact table. Our algorithm did not handle such kin of data. This also causes an inappropriate design of the target schema. That is why we introduce manual refinements in our work to remove such kind of bottlenecks. We are working these problems as our future work. We may introduce

some pre conversion steps in the underlining source systems as schema enhancement. Then this enhance schema should be the input of our algorithm.

Another limitation as discussed in above paragraph, some time it is desired that the OLTP developers have to keep numeric data in textual fields. These kinds of situations come across many times in the development environment. Our algorithm should be adoptable. We are planning to introduce adaptability in our algorithms as a future work. Then our algorithm will handle such kind of problems also.

In our algorithm hierarchy level and numeric field are basic measures for the selection candidate logical schema. Designer of the OLTP systems keep some information in the numeric fields that is actually not measure for the fact table. This is a disinformation to our algorithm. On the basic of this misinformation, algorithm may not generate desired results. We give manual refinement step to handler such kind of problems. This is also our future work, which may be handled by schema enhancement mechanism

One advantage of a data warehouse versus an OLTP system it that the data warehouse can integrate the data of multiple OLTP systems for user analysis. This can pose a problem to schema automation and evaluation. If the OLTP systems are completely separate then it is impossible for an automated algorithm to know how any two are related to represent in a single schema. User queries may be created along the OLTP divisions in which case the evaluation of a schema would only look at candidate schemas from a specific OLTP system. This is not a very integrated data warehouse. If the OLTP schema is across various business units and OLTP systems then the relationships are defined and the algorithm works well. But enterprise-wide schemas cannot be easily generated and may not exist. If they do not then the process of integrating business functions is a manual one. Only queries spanning business operations will show the schemas that are truly useful to an enterprise-wide data warehouse. On the other hand, if queries exist that span the business yet there is no schema that does, it will be impossible to fully answer the queries. Although a number of limitations have been identified most have viable (possibly manual) solutions. Most of the limitations are not common. In most cases one-to-one relationships in OLTP systems would already be merged into a single entity. It should be rare that numeric type fields are stored in character strings because of the lack of applicable constraints that could then

be applied. Even though these scenarios require manual work they should be less frequent, meaning the automated algorithm would pick up most of the work.

7. 5 Summary

This chapter has the concluding remarks of the research work. In first section we briefly describe the need of the new technology and improvements in Schema conversion algorithm. In the next section we mention the achievements which we got from the proposed . Then improvements are discussed which can be attained by implementing our technique. At the end we propose some recommendations and a direction where more attention is required.

References

References and Bibliography

[1] P. Ponniah "Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals." John Wiley & Sons Inc. 2001

[2] Kimball, Ralph; Ross, Margy. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition, John Wiley and Sons, Inc., Chichester, 2002.

[3] Moody, D.; Kortnik, M.: "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design". DMDW'00, Sweden, 2000.

[4] N. Tryfona, F. Busborg, and J. G. B. Christiansen,"StarER: A Conceptual Model for Data Warehouse Design," Proceedings of the ACM DOLAP99 Workshop,Missouri, November 2-6, 1999.

[5] Hüsemann, B., J. Lechtenbörger and G. Vossen, 2000. Conceptual data warehouse design. Proc.Intl. Workshop on Design and Management of Data Warehouse (DMDW '2000), pp: 6-1-6-11.

[6] Tryfona, N., F. Busborg and J.G.B. Christiansen,1999. starER: conceptual model for datawarehouse design. Proc. ACM 2nd Intl. Workshop on Data Warehousing and OLAP, pp: 3-8.

[7] J. Srivastava and P. Chen, "Warehouse Creation A Potential Roadblock to Data Warehousing," IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, pp.118–126, January/February 1999.

[8] C. Ballard , D. Herreman, D. Schau, R. Bell, E. Kim, and A. Valencic, Data Modeling Techniques for Data Warehousing, IBM Redbook, IBM International Technical Support Organization, Feb-26-1998, ISBN No. 0738402451. http://www.redbooks.ibm.com/abstracts/sg242238.html

[9] M. Golfarelli and S. Rizzi, "Designing the Data Warehouse: Key Steps and Crucial Issues," Journal of Computer Science and Information Management, Vol. 2, No. 1, pp. 1-14, 1999.
http://www-db.deis.unibo.it/~srizzi/Pubs.html

[10] R. Kimball, "Slowly Changing Dimensions," DBMS Magazine, April 1996.
http://www.dbmsmag.com/9604d05.html

[11] R. Kimball, "Factless Fact Tables," DBMS Magazine, Sept. 1996.
http://www.dbmsmag.com/9609d05.html

[12] R. Kimball, "There Are No Guarantees: Entity-relationship Modeling Is Far from Being a Universal Solution for Data Warehouse Business Rules," Intelligent Enterprise, Aug. 2000.
http://www.intelligententerprise.com/000801/webhouse.shtml

[13] S. Mahajan, "Building a Data Warehouse Using Oracle OLAP Tools," Oracle Technical Report, ACTA Journal, Sept. 1997.
http://www.oracle.com

[14] F. McGuff, "Designing the Perfect Data Warehouse," 1998.
http://members.aol.com/fmcguff/dwmodel/index.htm

[15] N. Raden, "Modelling the Data Warehouse," Archer Decision Sciences, Inc., 1996. http://netmar.com/~nraden/iw0196_1.htm

[16] N. Raden, "Technology Tutorial – Modeling A Data Warehouse – Value to an Organization Means Turning Data into Actionable Information," InformationWeek, Jan. 1996, Issue 564.
http://www.techweb.com

[17] M. Boehnlein and A. Uldrich-von Ende, "Deriving Initial Data Warehouse Structures from the Conceptual Data Model of the Underlying Operational Information Systems," Workshop Proceedings DOLAP99, Missouri, November 2-6, 1999.
http://www.cis.drexel.edu/faculty/song/DOLAP99/DOLAP99.htm

[18] M. Golfarelli, D. Maio, and S. Rizzi. "Conceptual Design of Data Warehouses from E/R Schemes," Proceedings of the 31st Hawaii

International Conference on System Sciences (HICSS-31), vol. VII, Kona, Hawaii, pp. 334-343, 1998. http://www-db.deis.unibo.it/~srizzi/Pubs.html

[19] J. M. Firestone, "Architectural Evolution in DataWarehousing and distributed Knowledge Management Architecture," White Paper No. Eleven, executive Systems, Inc., 1998. http://www.dkms.com/ARCHEV.html

[20] W. H. Inmon, Building the Data Warehouse, John Wiley and Sons, Inc., New York, 1992.

[21] K. Hahn, C. Sapia, and M. Blaschka, "Automatically Generating OLAP Schemata from Conceptual Graphical Models," Proceedings DOLAP 2000, pp 9-16, 2000.

[22] S. Sapia, M. Blaschka, G. Hofling, and B. Dinter, "Extending the E/R Model for the Multidimensional Paradigm," Proc. Intl. Workshop on Data Warehouse and Data Mining (DWDM `98), Singapore, November 1998, volume 1552 in LNCS, Springer, 1999. http://www.forwiss.tu-muenchen.de/~system42/publications/dwdm98.pdf

[23] N. Tryfona, F. Busborg, and J. G. B. Christiansen, "starER: A Conceptual Model for Data Warehouse Design," Workshop Proceedings DOLAP99, Missouri, November 2-6, 1999. http://www.cis.drexel.edu/faculty/song/DOLAP99/DOLAP99.htm

[24] M. Golfarelli and S. Rizzi, "A Methodological Framework for Data Warehouse Design," Proceedings ACM First International Workshop on Data Warehousing and OLAP (DOLAP), Washington, pp. 3-9, 1998. http://www-db.deis.unibo.it/~srizzi/Pubs.html

[25] R. Kimball, "A Dimensional Modelling Manifesto," DBMS Magazine, Aug. 1997. http://www.dbmsmag.com/9708d15.html

[26] M.C. Wu and A.P. Buchmann, "Research Issues in Data Warehousing," Intl. Conference on Databases in Office, Engineering and Science (BTW'97), Ulm, Germany, March, 1997.

[27] R. Kimball, The Data Warehouse Toolkit, John Wiley and Sons, Inc., New York, 1996.

[28] R. Kimball, "Data Warehousing Gets the Data Out," DBMS Magazine, Sept. 1995. http://www.dbmsmag.com/9509d05.html

[29] J. Trujillo and M. Palomar, "An Object-Oriented Approach to Multidimensional Database Conceptual Modeling (OOMD)," Proceedings of the DOLAP Workshop, Washington, D.C., November 7, 1998. http://www.cis.drexel.edu/faculty/song/dolap.html

[30] Phipps, C.; Davis, K.: "Automating data warehouse conceptual schema design and evaluation". DMDW'02, Canada, 2002.