Storage and Cost Efficient Mining on an OLTP System using Schema Enhancement Method

Developed By

Syed Mubashir Hasan

Co Supervised By

Muhammad Imran Saeed

Supervised By

Dr. Malik Sikandar Hayat Khiyal

Department of Computer Science
International Islamic University,
Islamabad
(2010)

1. Computer storage devices

2. Associative storage

**WITH THE NAME OF
ALMIGHTY ALLAH,
THE MOST BENEFICIENT,
THE MOST MERCIFUL**

A dissertation Submitted To

Department of Computer Science,

International Islamic University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the

Degree of MS in Computer Science

Department of Computer Science

International Islamic University Islamabad

Date: 12/08/2011

Final Approval

This is to certify that we have read the thesis submitted by **Syed Mubashir Hasan** Registration No 118-CS/MS/03. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of MS in Computer Science.

Committee:

External Examiner
Dr. Maqbool Uddin Shaikh
Professor
COMSATS: Department of Computer Science
Islamabad.

Internal Examiner
Asim Munir
Assistant Professor
Department of Computer Science,
International Islamic University,
Islamabad.

Supervisor
Prof. Dr. M. Sikandar Hayat Khiyal
Chairman, Department of Computer Science,
Fatima Jinnah Women University,
Rawalpindi.

SHayat

Co-Supervisor
Imran Saeed
Assistant Professor
Department of Computer Science,
International Islamic University,
Islamabad.

Dedicated To

The Most Beloved Hazrat Muhammad (SAW),

My Beloved Late Father

My Motherland

And My Family

Syed Mubashir Hasan

Declaration

I hereby declare that this Research ***"Storage and Cost Efficient Mining on an OLTP System using Schema Enhancement Method"*** neither as a whole nor as a part has been copied out from any source. It is further declared that I have done this research with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of my teachers especially my supervisor **Dr. Malik Sikandar Hayat Khiyal**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from any of the training institute or educational institutions, I shall stand by the consequences.

Syed Mubashir Hasan

118-CS/MS/03

Acknowledgement

First of all I am thankful to Allah Almighty the Most Merciful, the Most Beneficent and the source of all Knowledge, for granting me courage and knowledge to complete this Project.

I am obliged to **Dr. Malik Sikandar Hayat Khiyal** for his kind behavior and guidance as supervisor. Beyond doubt, I would never have completed this task without his interest and strict standards of perfection.

I want to thank all the respected faculty members for teaching me in a very professional way.

I want to show my regards and greetings to all of my class fellows and friends with whom i spent a very nice and memorable time.

Above all, I owe every thing to my beloved parents and my loving families for their love, guidance and their moral and financial support.

Syed Mubashir Hasan

118-CS/MS/03

Project In Brief

Project Title:	Storage and Cost Efficient Mining on an OLTP System using Schema Enhancement Method
Undertaken By:	Syed Mubashir Hasan
Supervised By:	Dr. Malik Sikandar Hayat Khiyal
Start Date:	December 2006
Completion Date:	December, 2009
Tools & Technologies	Oracle Server Database Server. Visual C#.Net (To Develop Simulating Software)
Documentation Tools	Microsoft Word XP Microsoft Visio XP Microsoft Project 2000 Rational Rose 98
Operating System:	Windows 2000 Professional
System Used:	Intel Pentium 4 1.6 GHz

Abstract

Data Mining has been a topic of wide interest now-a-days. Researcher have been focusing to improve the DSS solutions and to make them more reliable and practical for OLAP (DWH) systems. However to facilitate the small and medium size businesses, there is a need for efforts to implement the same mining techniques in OLTP systems. Schema Enhancement method has not only established itself to be very practical but also to improve the performance of the system [1]. Applying this method has also proven to be storage and cost effective for OLTPs.

Schema Enhancement Method [1] will be applied to the normal OLTP system and the Mining module developed by this method will be a part of actual schema of OLTP.

TABLE OF CONTENTS

Chapter No **Contents** **Page No**

Chapter No **Contents** **Page No**

CHAPTER 1

INTRODUCTION

1. Introduction

Healthy business competition has always been a mean of energy in the market that makes market runners to strive for the best; leading towards the success. The business competition is not a new phenomenon of modern era, but in fact it has got its roots deep in the history. Businessman of every time, no matter what the size of business is, tries to be at his best, as compared to others in the market. Well, "being the best" is not achieved simply; it rather requires extraordinary talent, deepest sincerity towards work, maddening passion, the thirst to prove ones' self, and above all, the ability to take the best decision at the right time.

In past, decision making was supported mainly by experience and luck. People with vast experience in market were considered to have monopoly/command in taking good timely decisions for their business. Sometimes, decisions taken by new comers also proved successful, but it was probably only an outcome of their sheer luck. Decision making was a difficult task because it was really hard to keep an eye over the entire market, analyze packages provided by competitors and to evaluate customer's response on those packages. In those days, decision making was based merely on human observation. A shopkeeper for example was supposed to decide for ordering new stock for his shop only by daily checking the shelves of his shop. If the stock was present in sufficient amount for a few days, it was ok; otherwise, it was time for the news stock to be ordered.

The problems with observation based decision making were that, firstly we were left with no summary of our business transactions. Secondly, it rendered us with a very limited past record of our business dealings. Thirdly, all decisions were wholly dependent on a single or a small group of experienced people and no one else was able to take decisions on his/their behalf, as he/they were the only knowledge bearers. Forth and most importantly, the observation method worked only for the decision making of a small shop or business that had few daily transactions. For widely spread businesses, having thousands and millions of simultaneous transactions per day, that too at multiple branches, this method failed to support meaningful decision making.

With the emergence of foreign products in the local markets, the situation became more complex, because now the customers had more choices available to pick from. The need for proper, accurate, and timely decisions became more prominent. There was a desperate need for a system where the executives and top management was provided with not only the sales performance, products and employees data of a single branch, but also the summery of the said matrices in all branches. A view of the overall market situation and some knowledge about what the competitors were doing would also be very fruitful.

Decision making is a process of choosing among alternative courses of actions for the purpose of attaining a goal or goals. According to Herbert A. Simon [6], managerial decision making is synonymous with the whole process of management. To illustrate the idea, consider the important managerial function of planning. Planning involves a series of decisions: what should be done? When? How? Where? By whom? Etc. Hence, planning implies decision making [1].

The emergence of information technology has changed the structure of whole system of decision making. Computer systems can provide the management with all the necessary data that enables them to make a smart decision, with just a few clicks. With information technology now systems are available those actually do the task of decision making for its users. Such systems are called Decision Support Systems.

A DSS is an interactive, flexible, and adaptable Computer Based Information System (CBIS) specially developed for supporting a nonstructural management problem for improved decision making. It uses data collected over a period of time, provides easy user interface, and can incorporate the decision maker's own insights. In addition, a DSS may use models, is built by an interactive process (often by end-users), supports all phases of decision making, and may include a knowledge component [1].

1.1 Need of Data for Decision Making

Involvement of computer systems in the business decision making has eased the job of managers to a greater extends. All they have to do is to gather more and more data, as according to a recent concept Information Resource Management (IRM) data is a major corporate resource. So efficient management of data will result in efficient retrieval of information and that consequently will result in good business decisions. It's not only necessary to gather data of one's own organization to facilitate the decisions, but we should also keep data of our competitors to analyze their strategies and to improve our products. Most importantly an organization must keep information about its customers. Customer's demographics, their buying patterns and interests will help an organization to forecast the future trends regarding market and their products.

Decisions nowadays are wholly dependent on data. It has been observed that there are two types of decisions:

1. Cyclic Decisions
2. Liner Decisions

1.1.1 Cyclic Decisions:

The sales of many products vary with the changing climatic/seasonal conditions. For example, in summers, the sales of ice-cream, air conditioner, water coolers, juices, cold drinks, shorts and light dresses increases. Whereas in winters, the sale of these products decreases drastically.

On the other hand, the sales of woolies, like sweaters, shawls, scarves, caps, mufflers, socks, gloves etc. increases in winters, and decreases in summers. Decisions dependent on this type of data are cyclic in nature, i.e., they keep on changing with seasons but also keep occurring again. Such decisions are called cyclic decisions.

1.1.2 Linear Decisions:

These are decisions that could be taken only by analyzing the previous sales records. It means that the past values of data will shape the future decisions. For example, data regarding the eating habits of people will depict that the business of fast food will keep on a rise, irrespective of the climatic conditions.

To facilitate both types of decisions, there is a definite need of data. Therefore accuracy of the decision is dependent on the amount, correctness and relevance of the collected data.

1.2. Available Tools

We have put great emphasis on the need of data for decision making, but "data" alone cannot help decision making. A number of analyses are to be performed to bring meaning and sense to this data. Only then a correct and in time decision can be taken. Mostly, analysis for decision making is performed on huge amounts of data. Applying certain statistical and mathematical analysis on this data converts it into meaningful information about what have been the past trends and what changes have occurred during a span of time. This study will help in getting future forecast of trends and patterns of emerging market. To facilitate the analysis on huge data, a number of tools are now available in the market. These tools have matured over a period of time. Some of them are as follows:

1.2.1 Data Marts:

A data Mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subject [2].

The example of Data Mart could easily be taken from a University System, where we have different departments, i.e., Accounts Department, Examination Department, and Students Affairs Department. If we collect historical data of an Accounts department from all available sources, clean it, transform it into a uniform format and then load it in our computer so that our analytical queries be answered, it means that we have built a separate Data Marts for Accounts department of a University. All Data Marts of an organization along with other data structures collectively form a Data Warehouse. The data loaded into this system would preferably be summarized data, like the two attributes of STUDENT entity namely "current_date" and "date_of_birth", after summarization, it would become a single attribute "student_age".

Depending upon the source of data, Data Marts can be categorized as Independent or Dependent. Independent Data Marts are sourced form data captured from one or more Operational Systems, or external information providers, or from data generated locally within a particular department or geographic area. Dependent Data Marts are sourced directly from enterprise Data Warehouses [2]. The Independent Data Marts are created using the Bottom-Up approach of creating the Data Marts, in which developers start implementing the Data Marts first, which eventually form a corporate-wide Data Warehouse. Whereas the Dependent Data Marts are created by dividing the corporate-wide Data Warehouse into subsets, according to the needs of the different departments. The approach followed for creating the Dependent Data Marts is the Top-Down approach.

Data Marts may be stored and accessed separately. The level is at a departmental, regional or functional level. These separate Data Marts are much smaller, and they more efficiently support analytical types of applications [3].

1.2.2 Data Warehouse:

The day-to-day data of an organization is kept in its *operational Systems* also known as Online Transaction Processing System (OLTP), where the data remains fresh for some specific time, and then dumped into some files for references. However, for doing analysis, we need to see the values of different attributes, not just for its current values, but also for its previous values, to measure the degree of changes occur, the variation in trends over some time series, and the predict about its future values, in order to grab the market in our hands. The biggest question mark is "how to retrieve that data?" To satisfy this demand, we have Data Warehouses.

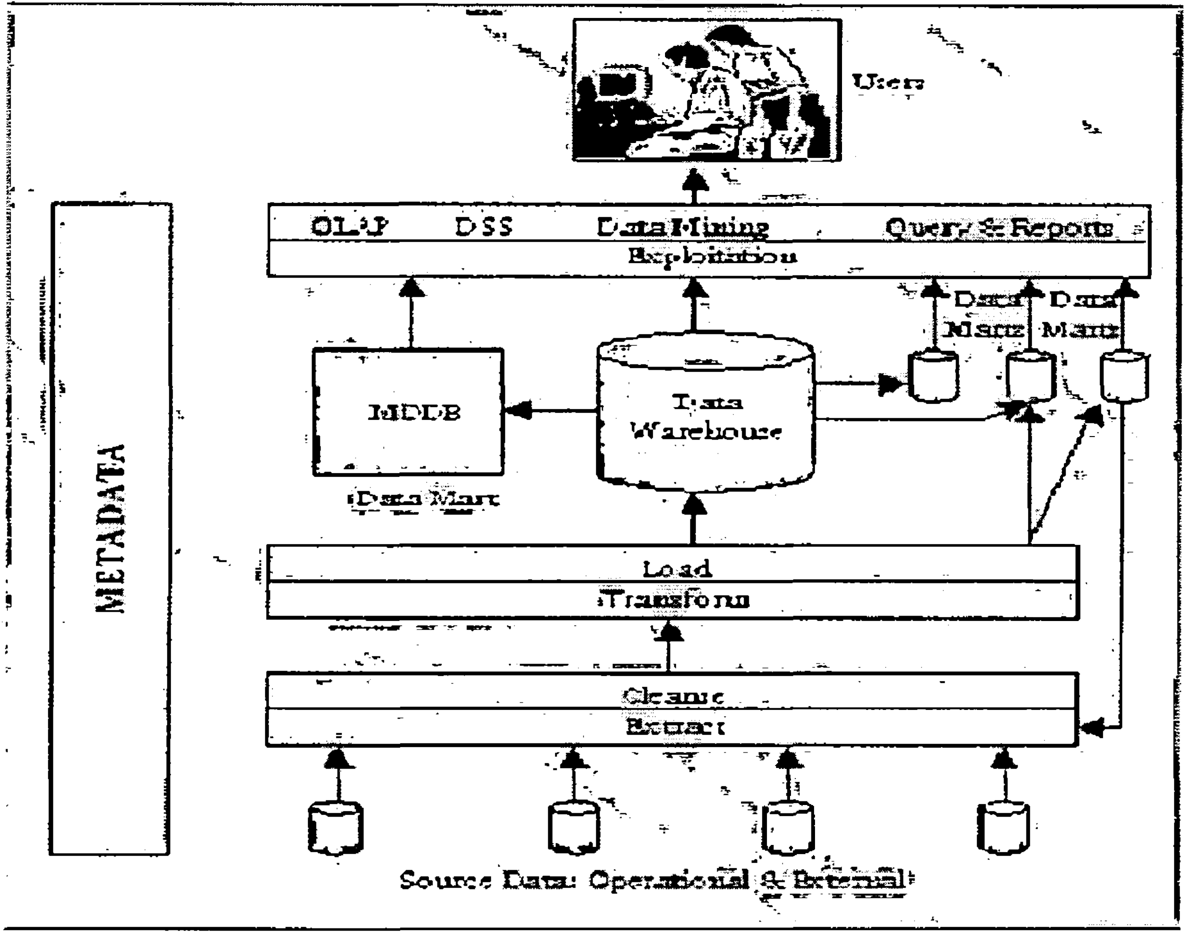


Fig 1-1 A Data warehouse Architecture [2]

A data Warehouse shown in fig 1-1 is a repository of information collected from multiple sources, stored under a unified schema, and which usually resides at a single site. Data Warehouses are constructed via a process of data cleaning, data transformation, data integration, data loading, and periodic data refreshing [2].

According to Inmon [7], Data Warehouse is "A subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making process" [4]. With Data Warehousing, corporate-wide data (current and historical) are merged into a single repository. It contains *informational data*, which are used to support other functions such as planning and forecasting [3]. The data has been gone through a number of phases to become informational data from operational data. There are many routines applied on it, commonly known as Extract-Transform-Load (ETL) routines, through which the data from different systems, files, mediums, data models and paradigms, is Extracted, and then Transformed into some unified and summarized format after cleansing, and then Loaded onto the Data Warehouse.

The basic motivation for this shift to the strategic use of data is to increase business profitability.

Traditional data processing supports the day-to-day clerical and administrative decisions, while Data Warehousing supports long-term strategic decisions. A 1996 report by International Data Corporation (IDC) stated that an average *return on investment* (ROI) in Data Warehousing reached 401% [3].

Data Warehouse is used for Ad hoc queries. The data remains static in the Data Warehouse as no changes or modifications could be done on it after it's once been loaded into Data Warehouse. The data schema used for it could either be the Star Schema (in which the tables or dimensions are kept demoralized), or the Snow-Flake schema (which normalizes the tables or the dimensions). Fig 1-2 shows a warehouse of the company.

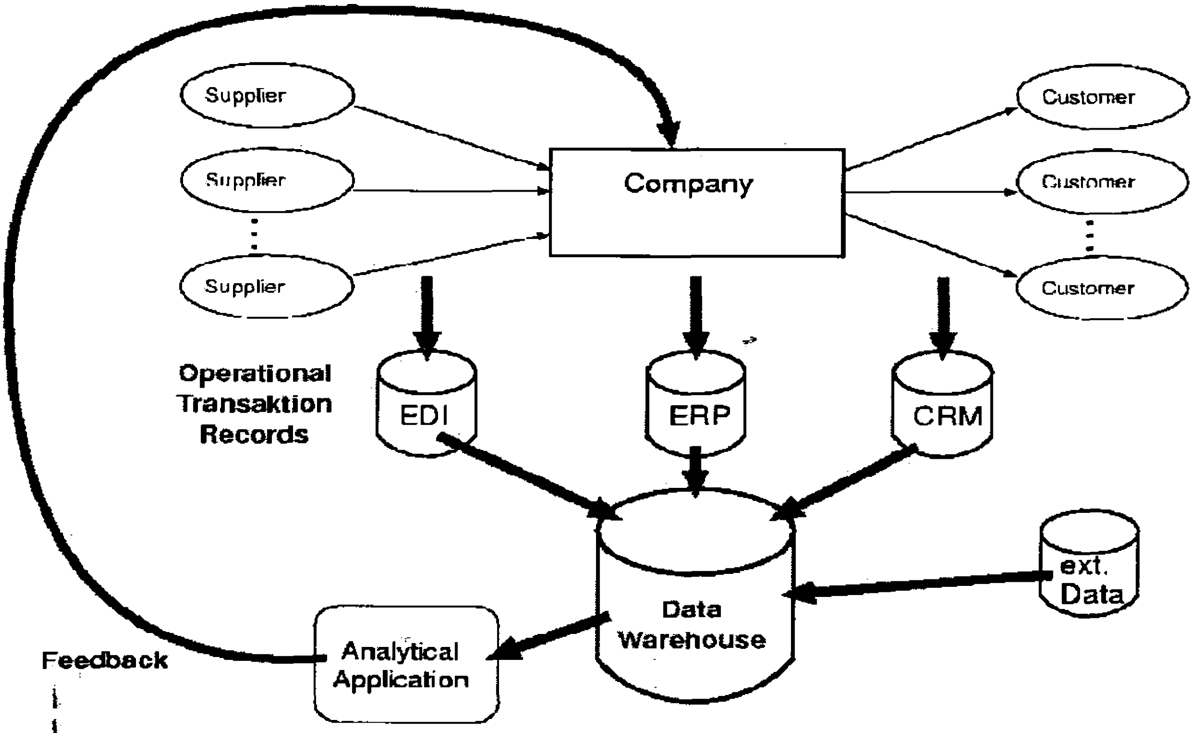


Fig 1-2 A Warehouse of a Company [3]

1.2.3 OLAP :

OLAP is a technology that uses multidimensional data representations called cubes for providing access to Data Warehouse data [5].

With operational data, we can have our simple queries answered. In order to entertain the complex queries (in which it might run Mining Algorithms to comprise a query), Online Analytical Processing (OLAP) systems are created. These applications analyze the data. So the OLAP applications are not only targeted to provide complex queries, but also to analyze the data, to answer the queries, which would eventually help to ease the decision making process.

OLAP operations make use of background knowledge regarding the domain of the data being studied in order to allow the presentation of data at different level of abstraction, to accommodate different user viewpoints [2]. By the level of abstraction, it means do we need to go down into more sophisticated search on the data, more deepest information about each product, employee etc., or we are confined with the average early sales of the north region? These OLAP operations are called Drill-Down and Roll-Up, respectively. Roll-Up allows the user to ask questions that move up in aggregation hierarchy and Drill-Down to get more detailed fact information by navigating lower in the aggregation hierarchy. To assist with Roll-Up and Drill-Down operations, frequently used aggregations can be precomputed and stored in the Warehouse [3].

OLAP can be a valuable and rewarding business tool. Aside from producing reports, OLAP analysis can aid an organization evaluate balanced scorecard targets. Fig 1-3 shows the Sequence data formation for Decision Making.

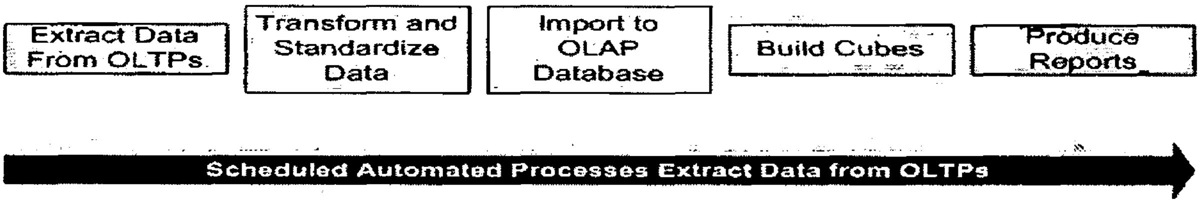

Extract Data From OLTPs
Transform and Standardize Data
Import to OLAP Database
Build Cubes
Produce Reports
Scheduled Automated Processes Extract Data from OLTPs

Fig 1-3 Sequence of Data formation for Decision Making [3]

1.2.4 Data Mining :

As we have seen that now a days every organization keeps Giga Bytes and Terra Bytes of data for decision making. DBMS is a technology to use or process that data with the help of SQL, Structured Query Language. But SQL has a limitation that it is structured language in which well known schemas, joins etc can be processed or accesses easily but in the case of warehouse we have terra bytes of data that is consolidated, aggregated, summarized and highly summarized depending upon the analysis requirement. So in order to explore that data we use another technique known as Data Mining. The process of extracting valid, previously unknown, comprehensive, and actionable information from large databases and using it to make crucial business decisions. [4]

Data Mining is the concerned with the analysis of data and use of software techniques for finding hidden and unexpected patterns and relationships in sets of data.

1.3 Overheads for a small or medium scale organization:

The tools mentioned in previous section are helpful in making a good, healthy and a timely decision, but there are some issues associated with them:

a) Volume of Data:

In the above mentioned tools, the huge amount of data is taken from different sources and platforms, and then is used for the analysis. In this case, it is intractable to conform the entire data of all sources, platforms and models, to a uniform standard. An endless number of synonyms, homonyms problems might arise. On the other hand, it is possible that we are entering the same data twice in the Data Warehouse, as different systems, from where we are taking the data and feeding in our Data Warehouse, will definitely be keeping the data about even the same entity in different ways. This problem degrades the performance of a Data Warehouse, because a user might be thinking that he has got the sufficient data, which in fact he does not.

b) Time Required:

Building a system like Data Warehouse is going to take a lot of time in development. Even its subset, the Data Mart, is taking no less time than 8-10 months (minimum) in creation. So an organization is going to wait for so long for the creation of its Data Warehouse, and then will be able to have full fruit out of it, which is not very likely condition for the growth of the business.

c) Hardware Requirement:

The Data Warehouses are comprised of the data spanning over the gigabytes or more, so it means a lot of hard disk is required to accommodate it. Also to execute the data mining queries on these Data Warehouse, an organization is supposed to provide the Efficient machines with high speed processors, more RAM and motherboards, otherwise, the query may crash or hang the entire system.

d) Software Cost:

Software costs of the Data Warehouses are also the same high as the hardware costs. The ETL routines available in the market cost a lot, which a businessman with small business can not afford. Similarly the data mining engines in the market are also costly for the small to medium business.

e) Specialized Staff:

The need for the qualified staff to take care of the Data Warehouse is also inevitable one, which is an extra cost for the small business.

1.4 Conclusion

Due to the importance of data and its importance in decision making it is unavoidable for any business, no matter what its size and nature is, to refrain from computerizing its system so almost every business do have its own Operational System due to its efficiency. And the growing awareness about its creation in the market also tempted the user to adopt it. But due to the overheads discussed above small and medium scale businesses can not afford to have a data warehouse. So if we make these Operational Systems to do analysis for us, by some schema enhancement, it will help us in many ways. For instance, no need to do the schema conformation, as the data is residing on the same disk, and is created by a single team, so it is already well structured, and free of anomalies. Secondly, no extra hardware, software, or administrative staff is required for it. The team creating the Operational System is going to do this schema enhancement on-the-fly. Which means, we can have a separate Entity, Table in our operational systems, in which to store the analytical data, so that it might provide us the decision making support. Last but not the least as the Decision Making related data will be kept separate from operation data so efficiency of the operational system will not be affected. And in addition to this small and medium scale organizations will be able to get the advantages of better decision making with the help of this enhanced schema.

CHAPTER 2

LITERATURE SURVEY

2. Literature Survey

Literature Survey is an important and unavoidable part of research. Without literature survey we cannot understand the basics of a subject, how far the researchers have taken the subject, the existing loopholes in the topic and what can enhancements can be brought in topic. For the very purpose we have gone through multiple research papers, articles and books to find out the current scenario of data mining in an OLTP and to discover new horizons waiting to be explored. Following is an article that discusses the concept of Analysis on an OLTP system on the basis of literature survey.

2.1 One Database Model for OLTP and OLAP:

Rehm et all [3] raise a question, "Can we use one database model and/or one database for both OLTP and OLAP worlds? Could we do justice to both worlds with a single model and/or database?" Sid Adelman, Les Barbusinki, Scott Howard, Mike Jenning, Chuck Kelley, David Marco, Joe Oates, Clay rehm reply the question which are stated below as it is.

Sid Adelman's Answer: [3]

"You are right! You cannot use the same database or even physical database model for both OLTP and any data warehouse (including OLAP) for the following reasons:

1. The designs are different. Trying to develop a design to satisfy both will be a compromise neither will like and the performance will be bad for at least one of them.
2. OLTP and data warehouses have different timeliness requirements. You do not need real time data for a data warehouse that you do for OLTP. In fact, analysts do not like a changing data warehouse.
3. A data warehouse query can sometimes suck resources to such an extent that you may severely hurt OLTP response time. Once that happens, the OLTP folks will kick you off their database.

The data warehouse has more stringent data quality requirements than are required in the OLTP system."

Les Barbusinski's Answer: [3]

"A "one size fits all" approach to database design *never* works. The database structures for OLTP and OLAP are totally at odds with each other because of the nature of the systems they serve. Bill Inmon [7] covers this very basic dichotomy in his landmark book, *Building the Data Warehouse*.

Whereas OLTP database structures:

- Are "atomic" (i.e., detailed)
- Are transaction-oriented
- Represent the current state of an entity
- Serve the clerical community
- Serve well-defined processes

OLAP database structures:

- Are aggregated and/or summarized
- Are analysis- oriented
- Represent a historical view of an entity
- Serve the management community
- Serve undefined *ad hoc* processes

As the saying goes: "form follows function." A database structure must reflect the function it is intended to perform, or it will not work."

Scott Howard's Answer: [3]

"Single Database? Possible. Single Model? You are living in the past.

Let's start with the easy part, single database. It is possible to have a single database engine, especially a parallel RDBMS handle both OLTP and OLAP needs. However, this is seldom recommended because the two workloads are very different. OLTP systems usually have a high constant transaction rate usually consisting of very simple read/write transactions. Systems administrators tune these systems to take most advantage of the resources available at constant rates, thus drive CPU and I/O usage as close to 100 percent as possible. This is in contrast to OLAP systems use which is inconsistent, primarily long- running and complex read-only transactions. This pattern leads to peaks and valleys in resource usage that when combined with OLTP usage can cause usage spikes well over resource capacity. These spikes can result in service-level violations for your OLTP system. Now this is a general scenario that may not apply to your specific implementation, so that's why we reserved judgment and claimed it's still possible to combine systems.

You can't combine models. The OLTP model is one that is intended to capture and efficiently manage the current state of your business. Short-term transactions, current inventory, monitoring current manufacturing processes and the like are the focus of most OLTP applications and systems. OLAP systems represent history and need to function in a way contrary to OLTP systems. That is they need to capture everything that goes on within our business including the net business result of an OLTP transactional update or delete, and represent and preserve that net meaning in a historical model. They also need to combine that with external events (promotions) and special external events (holidays, weather, manufacturing floor conditions, economic conditions etc.) so business analysts can make sense of the changes in our business captured from the OLTP models. These external events are also not generally represented in the OLTP models. Now we don't have room or time today to expand on how to do just that, but that's what OLAP or data warehouse modeling is all about and why if differs so from OLTP modeling."

Mike Jennings' Answer: [3]

"In most cases combining OLTP and OLAP traffic to a single database structure would be a mistake. Assuming that your OLTP application is running some segment of your company's business, you risking impacting its performance and ability to quickly process transactions by combining OLTP transaction processing and reporting with OLAP. In order for transactions to be processed efficiently, the data store would have to be in third normal form. This construct works fine for transaction processing but is inefficient for OLAP queries due to the excessive amount of joins that will be required to answer business questions. Both OLTP transactions and OLAP reporting will be competing for disk I/O which will degrade performance. End users running OLAP queries will experience ever- changing result sets of information in their queries as OLTP transactions are processed throughout the day. Aggregation, calculations, derived data and multipass processing will have to be performed during an OLAP query further degrading performance. In many cases, such as ERP systems, operational reporting against an OLTP system is performed in a secondary data store separate from where transaction processing occurs just to avoid the performance impact of operational reporting. Your company may decide to go down this path initially to save money but will quickly see the need to create a secondary data store for OLAP in order to be able to analysis strategic information in a efficient manner."

Chuck Kelley's Answer: [3]

"You are living in the now. You should not do both in a single database. May be you can use the same data model, as long as your model deals with historic views of data as it changes over time (which it probably doesn't – very few do). Vendors of middleware talk about this all the time, but there are some problems as i see it. Here is an excerpt I wrote as a Letter to the Editor of Computerworld published November 20, 2000.

"1. Do you really want hundreds of end users doing analysis of millions of rows asking queries into your transaction system, which is probably already undersized? I think not.

2. Do you have multiple definitions of the same object (Gender = M/F; 0/1; 1/2)? If so, do you really want users to be interpreting these in the product each time a user runs a query? I think not.

3. Do you have multiple applications that have different definitions for customers using different data types? If so, how is that handled within the product you are using? Do you really want that product to interpret "12345" to be different things to different systems each time a user runs a query? I think not.

4. Do you really want to keep 10-plus years of history in a transaction system, slowing it down? I think not.

5. Are your measurements in different metrics (currencies, metric vs. U.S. measurements)? Do you really want conversions on the fly? I think not.

6. Do you really want to process the same set of requests every time a user issues a query? I think not.

Granted, if you have a small single application that has an integrated environment (as very few do), then these products may work (though I would still be leery because of number I above)."

Of course, then my last statement has to discuss tuning. How do you tune an operating system with applications that does five reads, seven writes (typical transaction) and reads 100,00 rows and aggregate (typical data warehouse) at the same time? OK, if you could get past the operating system (Yes, I know all about MVS and the ability to run multiple versions of the OS, but there are some major limitations!), how do you do that for the database?

Well, I guess you can tell I have strong opinions on this topic."

David Marco's Answer: [3]

"You can do justice with a single logical model; however, different physical models will definitely be needed. The key to managing all of this data is a meta data repository. It is the system that manages your systems."

Joe Oates' Answer: [3]

"The simple answer is that you should have a separate database and machine for transaction systems and analytical systems. Data warehouse analytical reporting can often saturate I/O channels. This would certainly have a severe impact on transaction processing, especially OLTP. The same can be said of running a lot of reports while the OLTP system is up and running. I have seen many cases where the volume of operational reports made it necessary to duplicate the database on another computer and run reports only from that computer.

There are a couple of less desirable alternatives to the two separate machines. First, some of the larger hardware platforms can be partitioned so that a certain group of processors can be dedicated to OLTP and another group of processors can be dedicated to the data warehouse or other reporting functions. However, depending on the architecture, there still might be adverse impact on the OLTP systems because of the high I/O requirements of the data warehouse or other reporting requirements.

Second, analytic reports could be run at night when the OLTP systems are not running. However, this would probably interfere with nightly batch processing. Also, most employees would not be willing to come in at night to run ad hoc queries."

Clay Rehm's Answer: [3]

"In a perfect world, there would be one data model and one database. However in the operational world, companies are bought and merged, operational systems are retired, enhanced or newly built and it would be impossible to have a single database. The beauty of a data warehouse is that it is a separate database that integrates all of the operational databases into one, and it is designed for ad hoc query performance, not OLTP transaction update performance. I know there are RDBMS vendors who are working on improving their database system to handle both; however I am not sure we are there quite yet. And even so, for the reasons stated above, it just does not make political or financial sense."

2.2 Previous work

In order to get the clear picture of the previously done work we studied different papers few of which are discussed below:

2.2.1 Data Mining on an OLTP system (nearly) for free:

Erik Riedel, Christos Faloutsos, Gregory R. Ganger and David F. Nagle [1] present the idea of using Data Mining on an OLTP System by introducing a concept of scheduling disk requests that takes advantage of the ability of high-level functions to operate directly at individual disk drives.

According to the author this concept will not be resource hungry and time consuming and the load on an OLTP System will be approx zero when we will perform Mining on it. This means that a production OLTP system can be used for Data Mining tasks without the expense of a second dedicated system.

2.2.2 DBLearn: A System Prototype for Knowledge Discovery in Relational Databases:

Jiawei Han, Yongjian Fu, Yue Huang, Yandong Cai and Nick Cercone [2] describe a System DBLearn a Prototype system which was developed for knowledge Discovery in the large databases. This system adopts an attribute oriented induction approach which integrates a machine learning paradigm "learning from examples" with set oriented database operations and substantially reduces the computational complexity of database learning processes.

2.2.3 DBMiner: A System for Data Mining in Relational Databases and Data Warehouses:

Jiawei Han, Jenny Y. Chiang, Sonny Chee, Jianping Chen and Qing Chen. [3] describe DBMiner which is a system for Data Mining in Relational Databases and Data Warehouses. A system for Data Mining is developed by incorporating Data Mining Function including Characterization, Comparison, Association, Classification, Prediction and Clustering. and some Data Mining Techniques including OLAP and Attribute oriented induction, Statistical Analysis, Progressive deepening for Mining multiple level knowledge and meta rule guided mining.

As we know Mining on OLTP is not possible as the DBMS does not support this facility however by making some changes in the OLTP Models as proposed in the Paper by Erik Reidel, Christos Faloutsos, Gregory R. Ganger and David F. Nagle, Disk Scheduling Method, is an enhancement given to the DBMS to perform extra tasks without consuming more resources.

Similarly the Survey by Clay Rehm, Jeo Oates and David Marco also supports the idea of Mining on OLTP System by modifying the conventional OLTP Model but the Method of OLTP enhancement is not given.

Keeping in view above limitations and issues, OLTP Model [1] was proposed that will have some changes in the Schema. These changes will enhance the Scope of OLTP

Model and the Mining Process can be performed on OLTP Systems without consuming more resources.

2.2.4 Performance Efficient mining on an OLTP System using Schema Enhancement Method

A new approach has been proposed in this paper. The basic idea behind the research is to develop a new Processing Architecture that is more close to OLTP but it also support the OLAP features, the system takes the input from the OLTP system based on defined rules and map the requirement on OLAP for Mining Purpose. After implementation, the efficiency of system has increased and performance output is very positive. Performance aspect of the new approach has been discussed in details with identification of how the system will migrate to new approach.

Keeping in view we are going to work on the proposed OLTP Model that will have some changes in the Schema and it has already been tested. These changes will enhance the Scope of OLTP Model and the Mining Process can be performed on OLTP Systems without consuming more resources. After the implementation of this Schema, we will analyze the Storage and Cost Efficiency of the System.

The Performance aspects of the Schema Enhanced Method has already been discussed [3], however Cost and Storage aspects has not been included in that research. Cost and Storage has also been an issue that is faced by the OLTP. Daily increase in the data and problem of storage has forced the OLTP system to increase drastically.

OLTP system works on daily data inputs and thus running a 24 X 7 business is very critical if you also run the Reports during Business hours. One of the main problems that are faced by the OLTP system Architect is the Storage of Historical data. The data is very important to see the Past Trends, History and Revenue. Data Deletion and Archiving is not the solution because taking the Database Backups can also consume a lot of Resources and Cost.

CHAPTER 3

REQUIREMENTS ANALYSIS

3. Requirements Analysis

The requirement analysis is the first step towards software development. To minimize errors and to avoid deviations from the clients' expectations, analysis must be performed in a systematic and correct manner. Reliability and robustness of a software is also highly dependent on way the analysis is carried out. In the Requirement Analysis phase all possible requirements and expectations from the software are identified.

3.1 Problem Analysis

The analysis reveals the functional requirements of the system as under:

- OLTP Conventional Architecture will remain the same. All the changes required for Mining Module will be done independently in a different schema but on same machine.
- Depending on the requirement analysis of a business, Mining Module requirement gathering can be done at any time during the life of a system.
- It is not necessary to develop the Schema Enhancement Modules during the initial development.
- Once mining requirements are well understood, its schema is designed. The database is developed keeping in mind the existing system's data interface.
- Like most of the data warehouses, mining module is always dependent on existing system for its data requirements. The Mining module is fed by the existing system.
- Daily, monthly and yearly data and its summaries are stored in different tables for fast retrieval and to make the system easy to understand.
- Tables of enhanced schema are fed by Operational System (OLTP) for data. In case where the OLTP is already developed and functional and the mining system is deployed later, there is a need to transfer the data from OLTP to mining module, in order to mine the Data in Enhanced Schema.

- The enhanced schema can be populated through triggering processes or export routines. These data loading processes should be run in such a way that it does not have adverse affect on OLTPs performance.

3.2 Use Case Analysis

Analysis of a project can be represented in terms of use case diagrams indicating the actors and use case in expanded format. This helps in indicating system boundaries and in visualization of functionalities more clearly. The Use Case Model describes the behavior of the system when any of the actors send some stimuli.

Actors represent the role that can be played by the users of the system. These users can be humans, other computer, pieces of hardware and other software. Use case depicts the behavior in which the input by the actor is converted to and output by the use case. Alternate course of action is also depicted in extended use cases.

Use case Diagram of Query Processing is shown in Fig 3.1.

Use case Diagram of Time Calculation Process is shown in Fig 3.2.

Use case Diagram of Trigger or Loading Procedure is shown in Fig 3.3

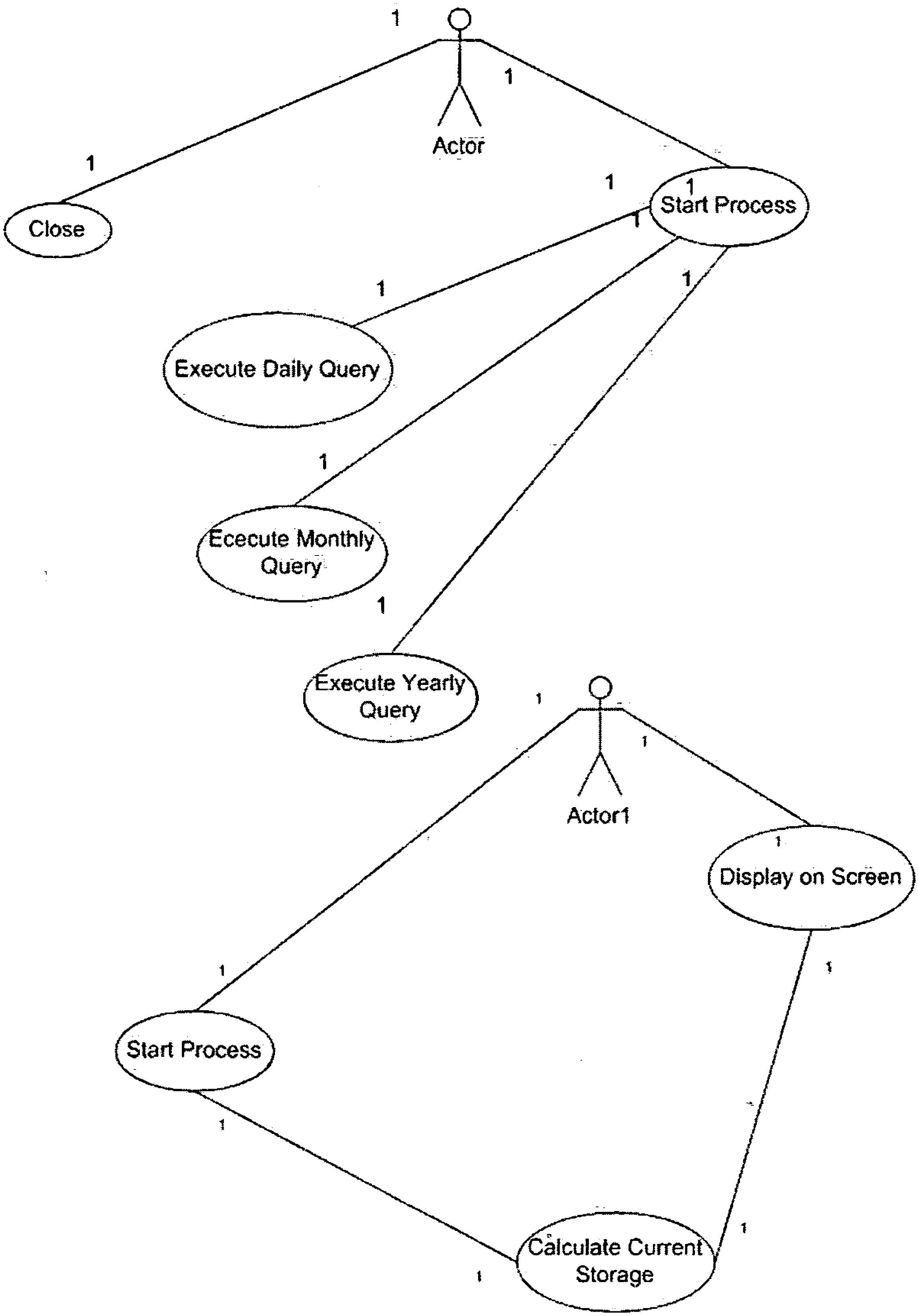


Fig 3-1 Use Case Diagram of Query Processing

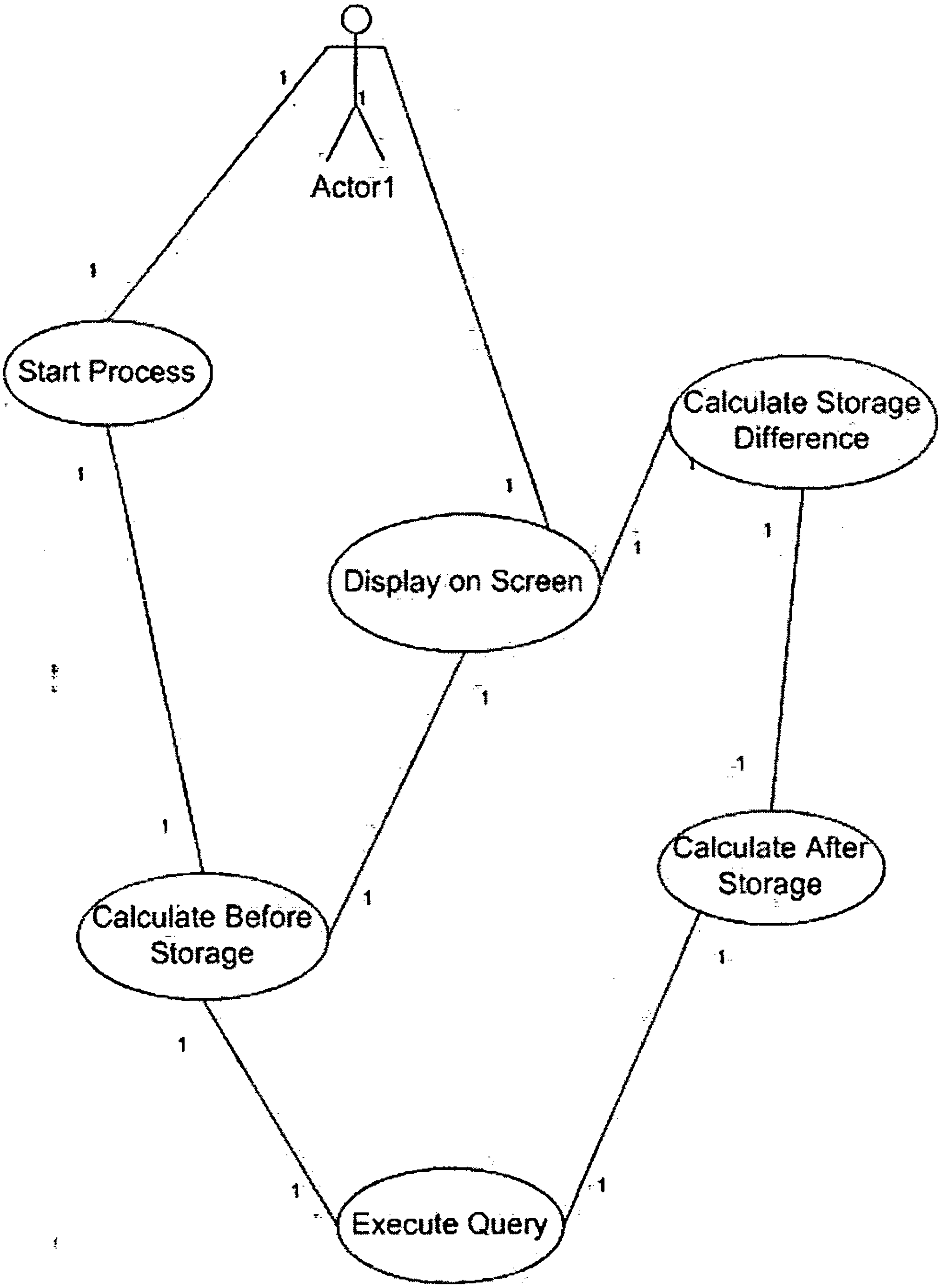


Fig 3-2 Use Case diagram of Time Calculation Process

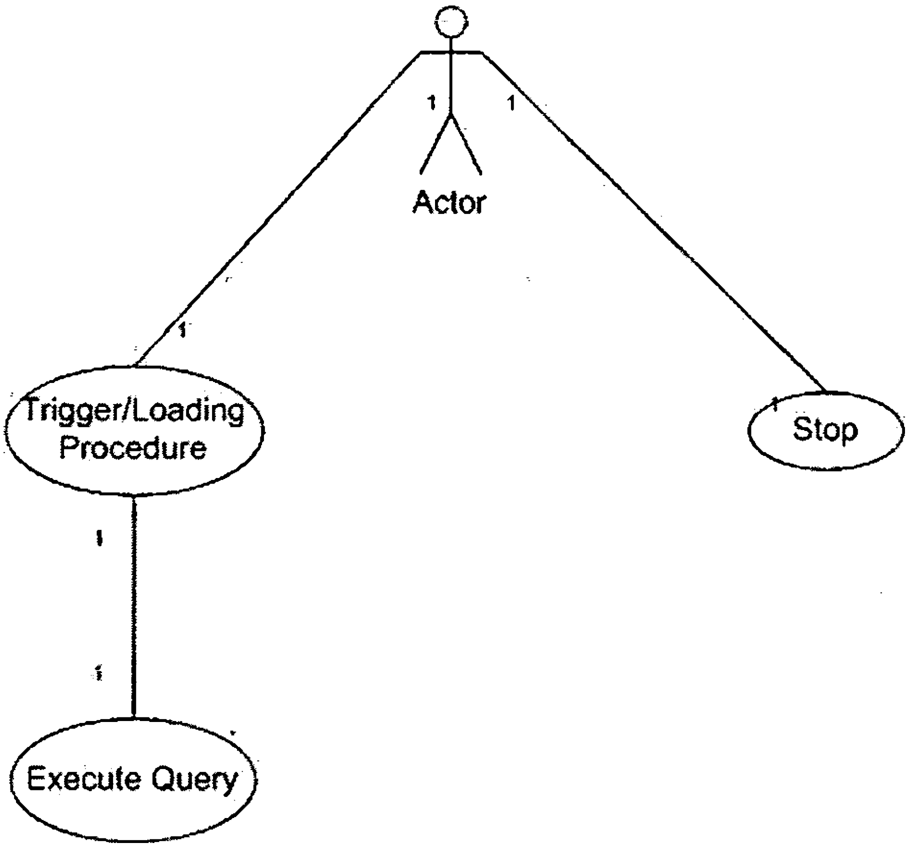


Figure 3.3 Use Case Diagram of Trigger or Loading Procedure

3.2.1 Use Case in Expanded Format

For each module of the project several use cases are identified and the description of each use case is as follows:

3.2.1.1 Start Application

a) Name: Start Application

b) Actor: User

c) Pre-Condition: None

d) Post Condition: Main Form Display on Screen.

e) Typical Course of Action:

Actor Action	System Response
1. User double clicks the application Icon.	2. OS Allocates memory and processor time to load and execute application. 3. System displays form on screen.

f) Alternate Course of Action:

Actor Action	System Response
1a. application is not executed. 3a. Repeat step 1 to 3	2a. Display OS error message.

3.2.1.2 Exit Application

a) Name: Exit Application

b) Actor: User

c) Pre-Condition: Application is in running state.

d) Post Condition: Application closes.

e) Typical Course of Action:

Actor Action	System Response
1. User presses close button.	2. All application variable and connection to SQL server are closed. 3. OS deallocates memory and removes it from process list. 4. Application closes.

f) Alternate Course of Action:

Actor Action	System Response
None	

3.2.1.3 SQL Server Connectivity

a) Name: SQL Server Connectivity

b) Actor: User

c) Pre-Condition: Start Process button clicked.

d) Post Condition: Connectivity Established.

e) Typical Course of Action:

Actor Action	System Response
1. Press Start Process Button	2. Connection String initializes. 3. Request sent to SQL Server for connectivity. 4. User name and password authenticated. 5. Establish Connectivity.

f) Alternate Course of Action:

Actor Action	System Response
No action	1a. SQL server error displayed on screen. 2b. No connectivity established.

3.2.1.4 Execute Query

a) Name: Execute Query

b) Actor: User

c) Pre-Condition: SQL server connectivity established.

d) Post Condition: Query result displayed

e) Typical Course of Action:

Actor Action	System Response
1. Press Start Process Button	1. Initialize SQL Query 2. Execute SQL Query 3. Fetch query results

f) Alternate Course of Action:

Actor Action	System Response
	2a. Error message displayed on screen.

3.2.1.5 Calculate Storage Difference

a) Name: Calculate Storage Difference

b) Actor: User

c) Pre-Condition: SQL server connectivity established.

d) Post Condition Results displayed to the user.

e) Typical Course of Action:

Actor Action	System Response
1. Press Start Process Button.	1. Calculate before storage 2. Display before storage on screen. 3. Execute query. 4. Calculate after storage. 5. Display after storage on screen 6. Calculate storage difference. 7. Display difference on screen.

f) Alternate Course of Action:

Actor Action	System Response
	4b. Error message is displayed.

CHAPTER 4

DESIGN

4. Design

In this chapter we will discuss the System and Database Design.

4.1 System Design (Object-Oriented Design Method)

System design is the specification or construction of a technical, computer-based solution for the business requirements identified in the system analysis. It is the evaluation of alternative solutions and the specification of a detailed computer-based solution. The design phase is the first step towards moving from problem domain to the solution domain. System design develops the architectural detail required to build a system or product. In this phase we have designed software that will be used to verify the proposed theory.

Object-Oriented design translates the Object Oriented Analysis (OOA) model of the real world into an implementation-specific model that can be realized in software. Object-oriented design transforms the analysis model, created using object-oriented analysis method, into a design model that serves as a blueprint for software construction. For the development of the system under consideration the same technique is used.

Object-oriented design (OOD) is concerned with developing an object-oriented model of a software system to implement the identified requirements.

Object Oriented Design builds on the products developed during Object-Oriented Analysis (OOA) by refining candidate objects into classes, defining message protocols for all objects, defining data structures and procedures, and mapping these into an object-oriented programming language (OOPL).

4.1.1 Class Diagrams

Class diagrams are the backbone of almost every object-oriented method including UML. They describe the static structure of a system. It can also be said that class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. The Class diagram is one of the most widely used diagrams from the UML specification.

Another purpose of class diagrams is to specify the class relationships and the attributes and behaviors associated with each class. Class diagrams are remarkable at illustrating inheritance and composite relationships. A class diagram consists of one major component and that is the various classes, along with these are the various relationships shown between the classes such as aggregation, association, composition, dependency, and generalization. Refer to figure 4.1 which represents the class diagram of the software that will show the processing of the queries and their time differences. This software module will help us to defend our concept of Storage and Cost efficiency in enhanced schema of OLTP.

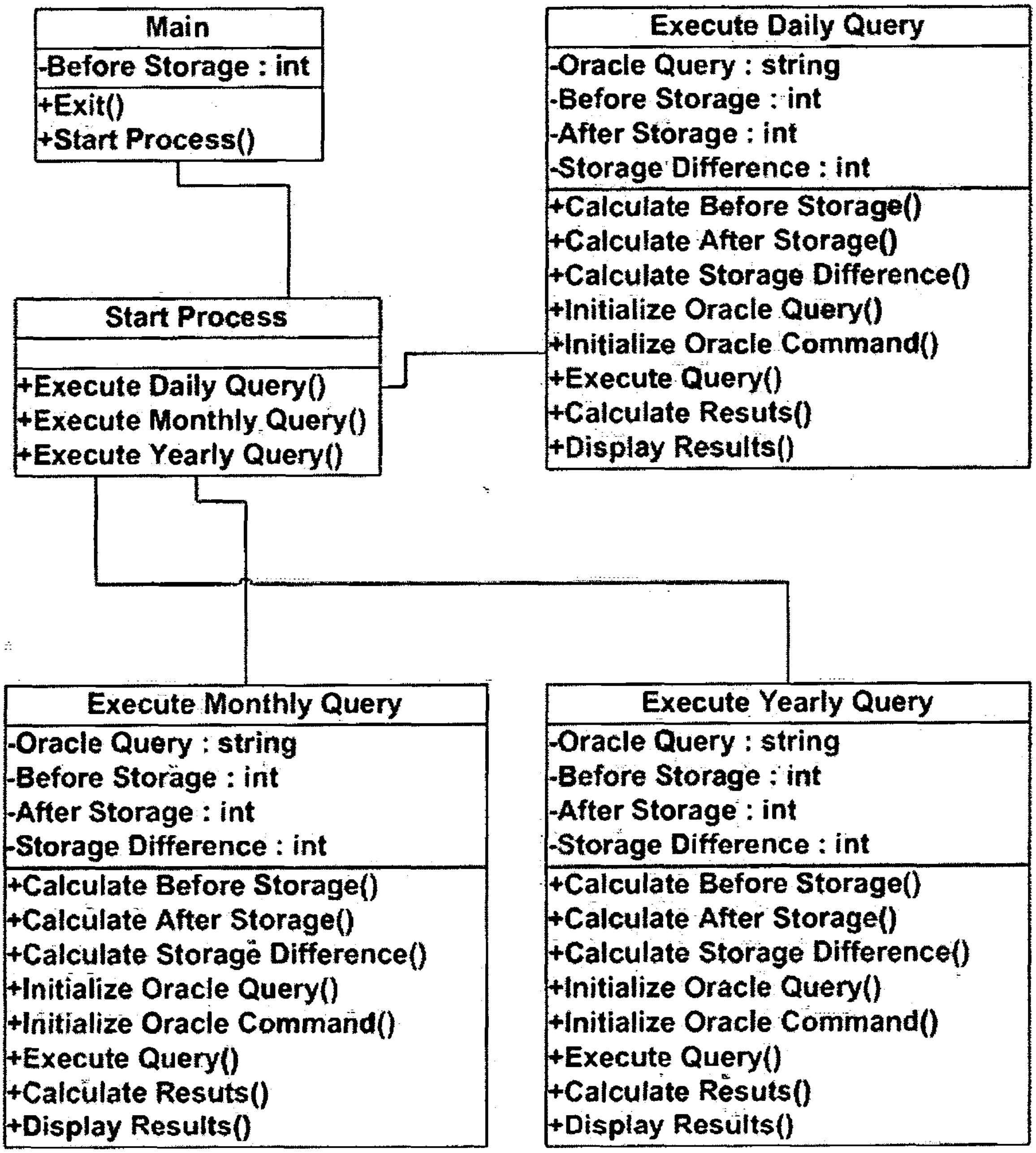


Fig 4-1 Class Diagram of Software Module

4.1.2 State Transition Diagram

In Fig 4.2 diagram shows the state transition of the software module that will represent the time for queries and also will calculate the time differences.

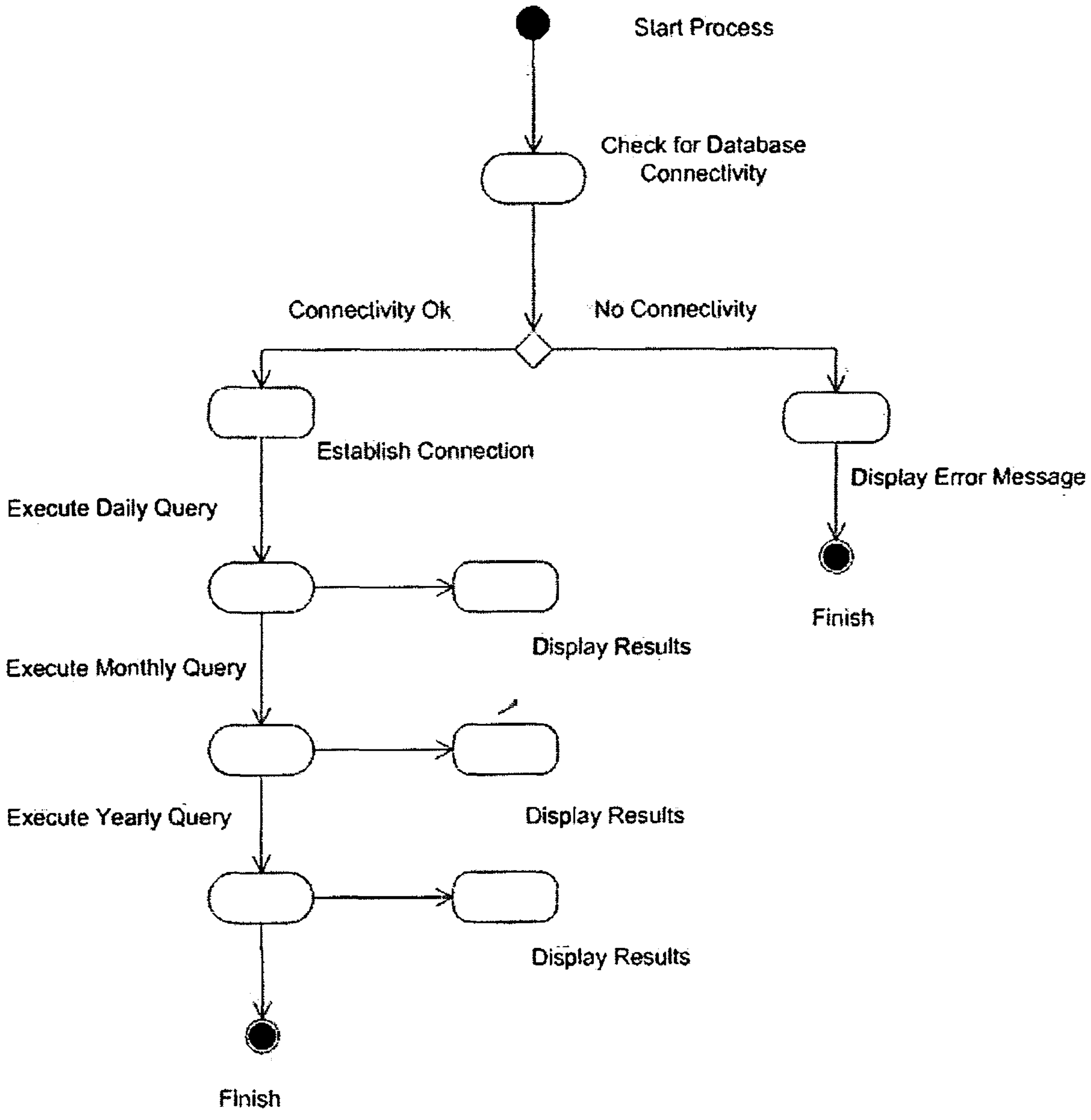


Fig 4-2 State Transition Diagram of Software Module

4.1.2.1 Detailed State Transition Diagram:

In Fig 4.3 diagram describes in detail that how a query works for a daily, monthly or yearly calculation.

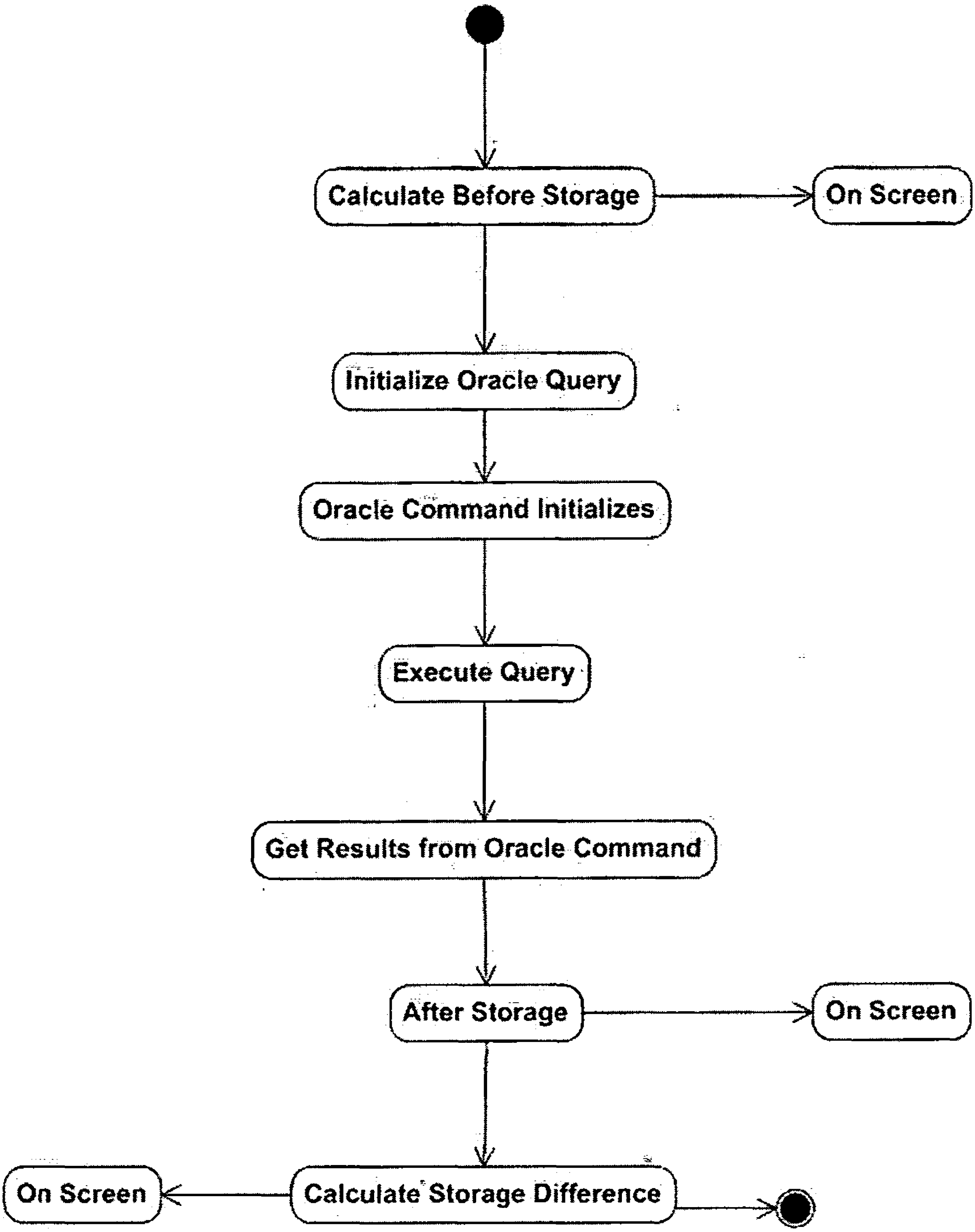


Fig 4-3 Detailed State Transition Diagram of Query Processing

4.1.3 Sequence Diagram

Once the use cases are specified, and some of the core objects in the system are prototyped, we can start designing the dynamic behavior of the system. Sequence diagrams demonstrate the behavior of objects in a use case by describing the objects and the messages they pass. Sequence diagrams emphasize the order in which things happen.

The Sequence Diagram with two Major events is shown below:

In Fig 4-4-a, the start application sequence is shown.

The user starts the application by clicking the Application Icon, the request is send to the Application Controller that will send execution request to OS. OS will accept the request and allocate the memory area and assign the process ID to this application and place it in the process table. After that the screen will be displayed on client area.

In Fig 4-4-b, the exit phase is shown.

User clicks the close button to allow the application to stop function. The request is send to OS that will de allocate the memory and close the process from the execution phase.

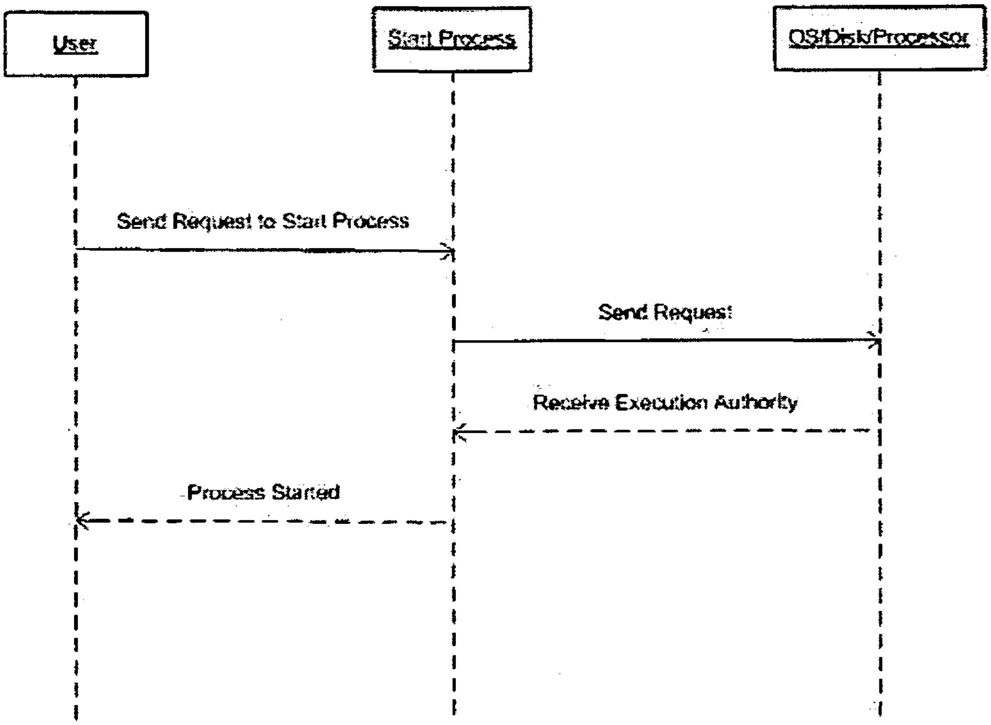


Figure 4.4-a: Start Process Sequence Diagram

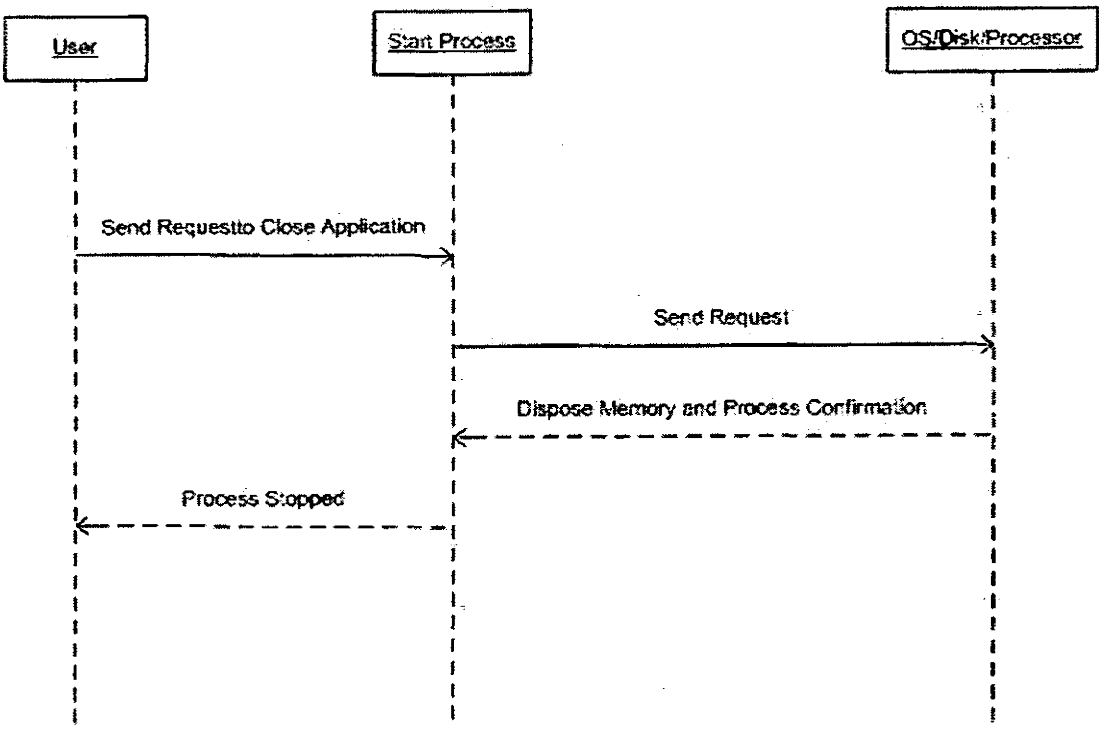

User
Start Process
OS/Disk/Processor
Send Requestto Close Application
Send Request
Dispose Memory and Process Confirmation
Process Stopped

Figure 4.4-b: Stop Process Sequence Diagram

4.2 Online Transaction Processing Architecture

Enterprise Architecture tells the physical structure of the System. It defines how the system hardware and software will work. User and client interaction with the system is also become clear from the diagram shown in Fig 4.5.

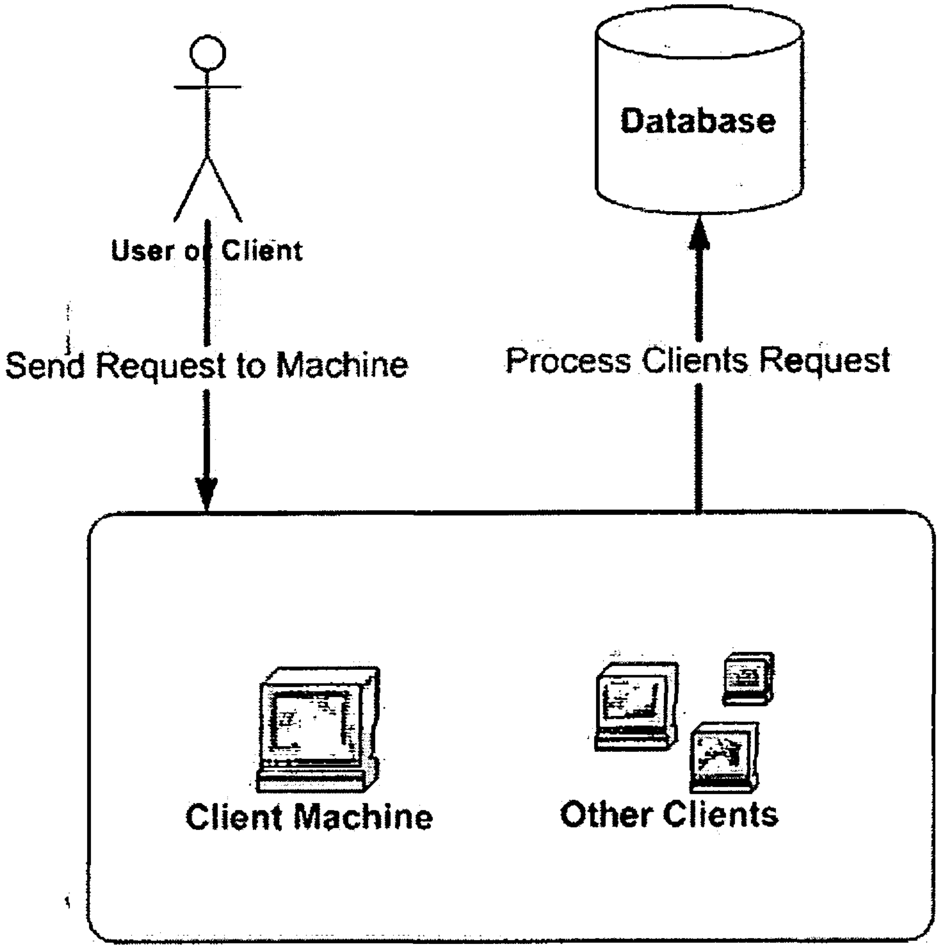


Fig 4.5: Architecture Design of OLTP

4.3 Database Design

The basic idea behind schema enhancement method is to design and create a database in the same operational system without affecting the performance of the operational system. In this method we take those attribute of the operational database that are required for the analysis and are used in data mining. We design another relational schema known as enhanced schema that is used for data mining.

4.4 Schema Enhancement Model:

Figure 4.6 represents the proposed schema enhancement model in which the database having the operational data is enhanced by designing another schema in which data required for data mining will be stored. While the other segment of the database i.e., operational database will keep on working as it is without any interruption or interference. A trigger or a developed procedure will be used to get the data from the operational database and it will be triggered or run at the off time when normal transactional load is low. So the efficiency of the system does not affect. [1]

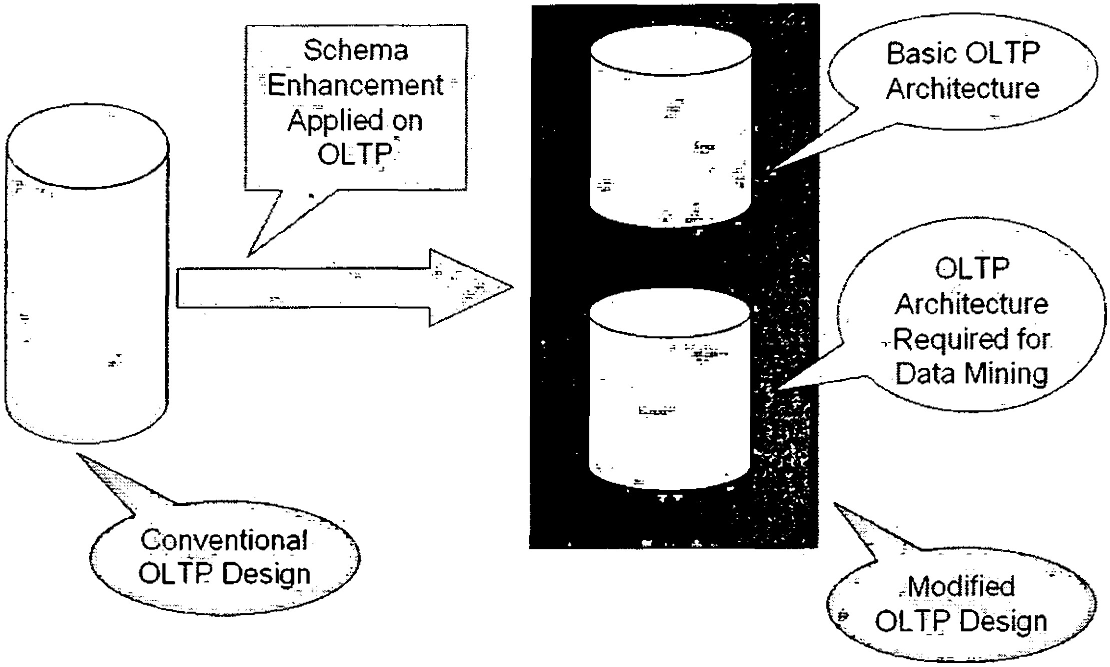

Schema Enhancement Applied on OLTP
Basic OLTP Architecture
OLTP Architecture Required for Data Mining
Conventional OLTP Design
Modified OLTP Design

Figure 4.6 Proposed Schema Enhancement Model [1]

4.5 Case Study:

In order to support this idea we have taken the case study of one module of PTML, UFONE (Pakistan).

4.5.1 Image Processing System:

The Schema (ERD) of that module is given in Fig 4.7. This is the ERD of Image Processing System. Now it is required by the authorities that how much data or how many records were gathers in a day or in a month or in a year or in years to analyze the trend. But the problem is if this query is executed in Operational System then it will affect the efficiency of the operational system especially when analysis is required on data of many years.

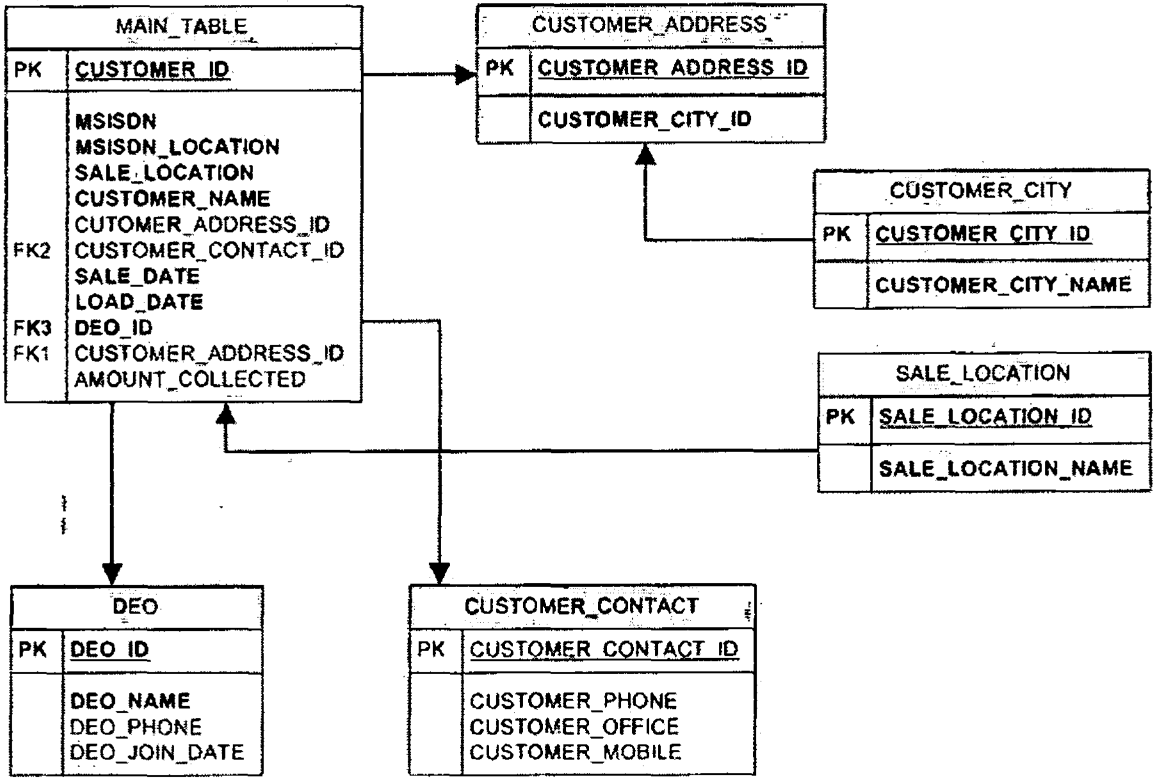


Figure 4.7: ERD of Image Processing System, UFONE.

4.5.2 Enhance Schema Model for Image Processing System:

In Figure 4.8 we have shown the enhanced schema that will be used for data mining without interrupting the operational system. It can be seen that many attributes have been removed which were not required for the mining and analysis. One thing must be clear that the proposed schema is still relational and fulfills the requirements to be a relational schema. One thing is very important that the summarized data is stored in this enhanced schema that reduces the requirements of bigger storage space.

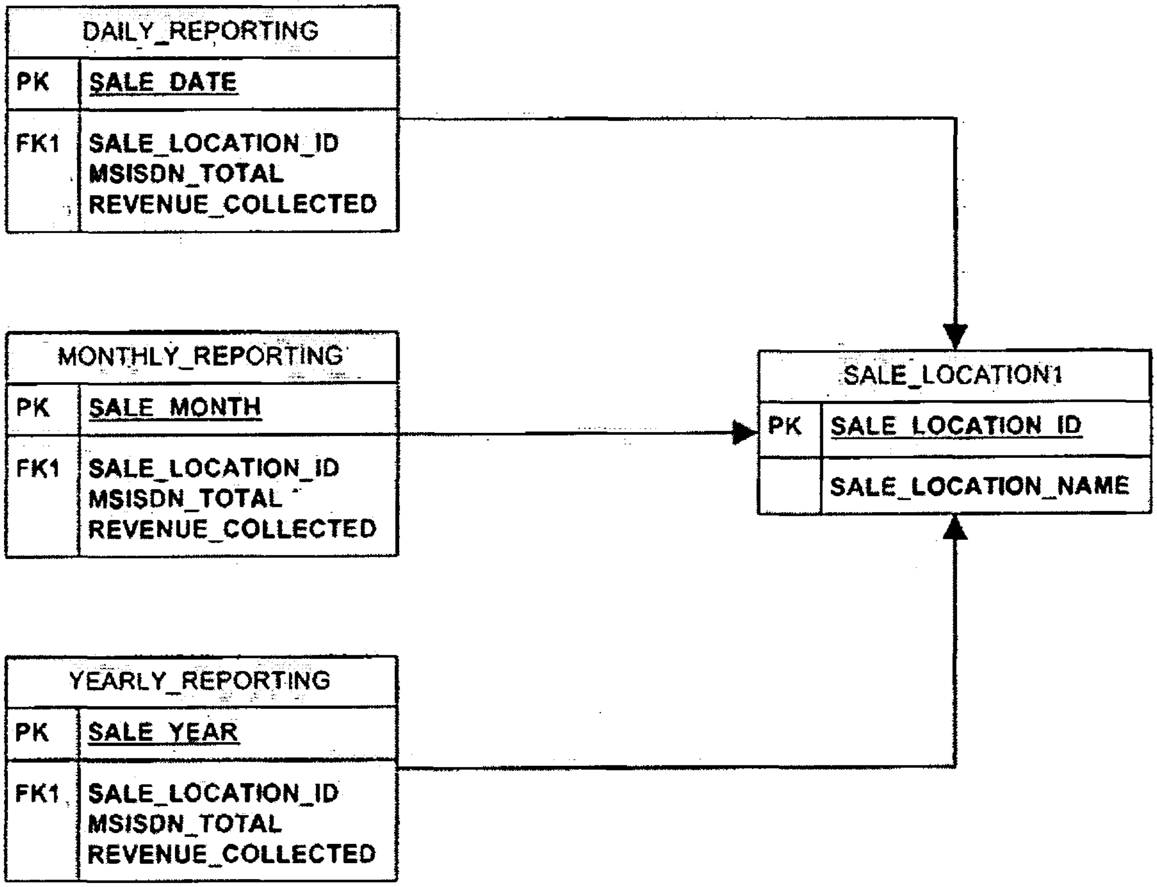


Figure 4.8 Proposed Enhanced Schema

CHAPTER 5

IMPLEMENTATION

5. Implementation

This project has been implemented in C#.Net. C# as described by wikipedia is a simple, modern, general-purpose, object-oriented programming language. As discussed earlier we are supposed to populate the enhanced schema from the existing OLTP system. For this purpose we use both triggers and procedures to populate the new schema designed for data mining.

5.1 Procedures:

Different Procedures have been developed to perform different tasks. The description of those procedures and their code is given below.

5.1.1 Procedure for Loading Data for the first time in enhanced schema:

This procedure has been written to load the available (previous) data for the first time into the enhanced schema relations when enhanced schema relations are created and are empty. Following code (query) describes the query that loads data into the relation:

```
---------------------------COMPLETE DATA LOADING

Insert into ips_daily_reporting
select to_char(sale_date,'DD-MON-
YYYY'),SALE_LOCATION,COUNT(*),SUM(AMOUNT_COLLECTED)
FROM IPS_MAIN
GROUP BY to_char(sale_date,'DD-MON-YYYY'),SALE_LOCATION
```

5.1.2 Procedure for Loading Data in enhanced schema on routine basis:

Following code contains queries that populate the enhanced schema on routine basis. This is done by automatic triggers in which we define the time to execute that code:

```
Insert into ips_daily_reporting
select to_char(sale_date,'DD-MON-
YYYY'),SALE_LOCATION,COUNT(*),SUM(AMOUNT_COLLECTED)
FROM IPS_MAIN where to_char(sale_date,'DD-MON-YYYY') = '12-AUG-2006';
GROUP BY to_char(sale_date,'DD-MON-YYYY'),SALE_LOCATION
```

5.1.3 Procedure for Loading Monthly and Yearly Data in enhanced schema:

the following code snippet are queries which calculate the monthly and yearly data from the data stored in the enhanced schema:

```
Insert into ips_monthly_reporting
select to_char(sale_date,'MON-
YYYY'),SALE_LOCATION,COUNT(*),SUM(AMOUNT_COLLECTED)
FROM IPS_MAIN
GROUP BY to_char(sale_date,'MON-YYYY'),SALE_LOCATION
```

```
Insert into ips_yearly_reporting
select
to_char(sale_date,'YYYY'),SALE_LOCATION,COUNT(*),SUM(AMOUNT_COLLEC
TED)
FROM IPS_MAIN
GROUP BY to_char(sale_date,'YYYY'),SALE_LOCATION
```

5.2 Simulation Software

In order to show results of this project by comparing its efficiency and performance a simulation software for this project is developed in Microsoft Visual C#.Net. Oracle 9i is used to store Database of the OLTP and enhanced schema. For connectivity we use the OracleCLient.dll library of the .Net frame work. The library provides instant and reliable connectivity with the Oracle Server.

5.2.1 Declaration of Library

Using C# the following method is used for library declaration.

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Data.OracleClient;
```

5.2.2 Timer Component to Display the Current Date and Time

Timer is component developed to display the current date and time value for reference and to compare the query time of the simulator.

```
label1.Text  =DateTime.Now.ToLongDateString() + " " +
              DateTime.Now.ToLongTimeString ();
label1.Refresh();
label1.Update();
```

5.2.3 Oracle Server Connectivity

Oracle Server Connectivity is achieved by using the build-in connectivity procedures of the C#.Net

```
System.Data.OracleClient.OracleConnection con = new
System.Data.OracleClient.OracleConnection();
            OracleCommand cmd;
            String mysql;
            OracleDataReader dr;
            con = new OracleConnection("Data Source=isb-msc-l-mubha;Persist Security
Info=True;User ID=ips;Password=ips;Unicode=True");
            con.Open();
```

5.2.4 Calculation of Daily Data Storage

This procedure is used to calculate storage of data used by IPS_MAIN table from Oracle Data Dictionary.

```
mysql = "select bytes/(1024*1024) size_mb from user_segments where
segment_NAME = 'IPS_MAIN'";
                cmd = new OracleCommand(mysql, con);
                cmd.CommandTimeout = 0;
                dr = cmd.ExecuteReader();
                while (dr.Read())
                {
                    label2.Text = dr["size_mb"].ToString();
                    label2.Refresh();
                    label8.Text = dr["size_mb"].ToString();
                    label8.Refresh();
                    label14.Text = dr["size_mb"].ToString();
                    label14.Refresh();
                }
```

This procedure is used to calculate storage of data from IPS_DAILY_REPORTING table used as Enhanced Schema table.

```
mysql = "select bytes/(1024*1024) size_mb from user_segments where
segment_NAME = 'IPS_DAILY_REPORTING'";
                cmd = new OracleCommand(mysql, con);
                cmd.CommandTimeout = 0;
                dr = cmd.ExecuteReader();
                while (dr.Read())
                {
                    label4.Text = dr["size_mb"].ToString();
                    label4.Refresh();

                }
```

5.2.5 Calculation of Monthly Data Storage

This procedure is used to calculate storage of data from IPS_MONTHLY_REPORTING table used as Enhanced Schema table.

```
                mysql = "select bytes/(1024*1024) size_mb from
user_segments where segment_NAME = 'IPS_MONTHLY_REPORTING'";
                cmd = new OracleCommand(mysql, con);
                cmd.CommandTimeout = 0;
                dr = cmd.ExecuteReader();
                while (dr.Read())
                {
                    label10.Text = dr["size_mb"].ToString();
                    label10.Refresh();
                }
```

5.2.6 Calculation of Yearly Data Storage

This procedure is used to calculate storage of data from IPS_YEARLY_REPORTING table used as Enhanced Schema table.

```
                mysql = "select bytes/(1024*1024) size_mb from
user_segments where segment_NAME = 'IPS_YEARLY_REPORTING'";
                cmd = new OracleCommand(mysql, con);
                cmd.CommandTimeout = 0;
                dr = cmd.ExecuteReader();
                while (dr.Read())
                {
                    label16.Text = dr["size_mb"].ToString();
                    label16.Refresh();
                }
```

5.2.7 Exception Handling

The code has been handled for any abnormal termination. Exception handling provides the actual error with detail description whenever the code returns an error during execution.

```
try
      {
      ---------Main Code---------
      }


      catch (Exception ee)

      {
            MessageBox.Show (ee.Message );
      }
```

5.2.8 Exiting from Application

```
this.Close() ;
```

CHAPTER 6

RESULTS

6. Results

After populating the data in the enhanced schema we executed the queries on the normal operational system as well as on the Enhanced Schema Model. The simulation software was then used to calculate the storage for each query and also its comparison.

The results are displayed on the screen along with the calculated comparison.

6.1 Main Screen

Test Run.exe Application is executed from the Application folder and following screen Fig 6.1 is displayed.

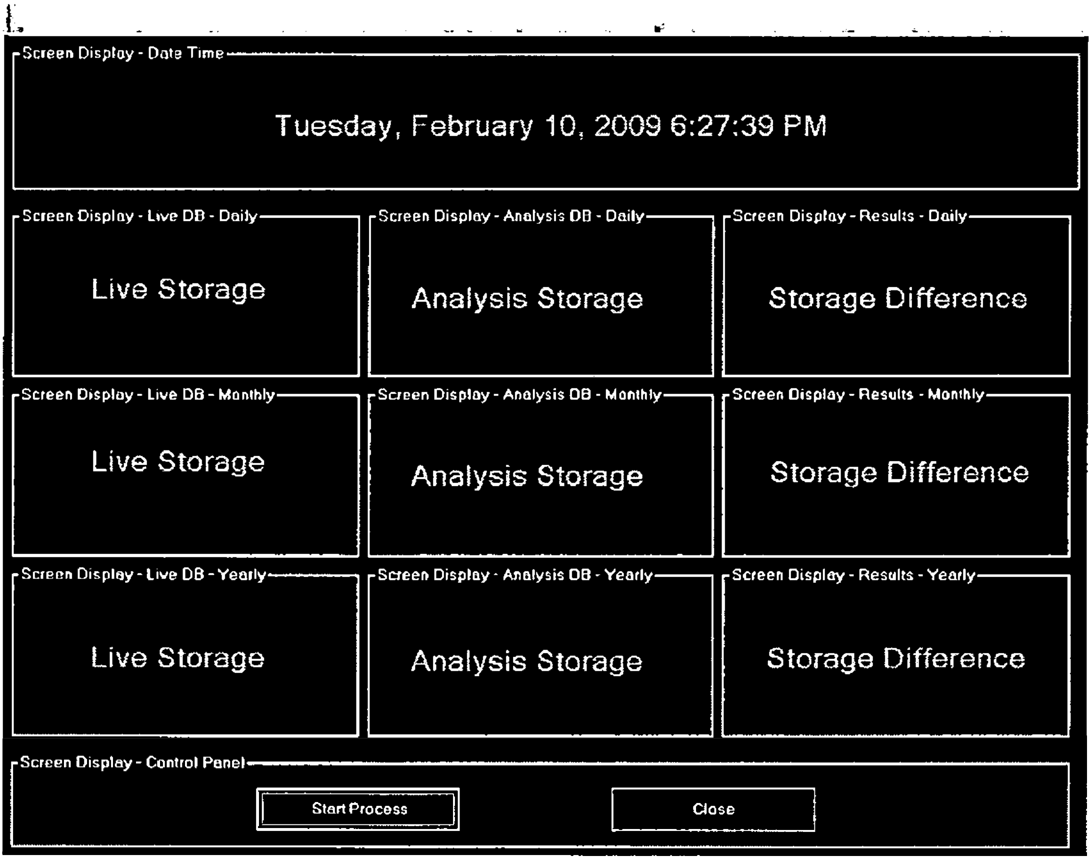
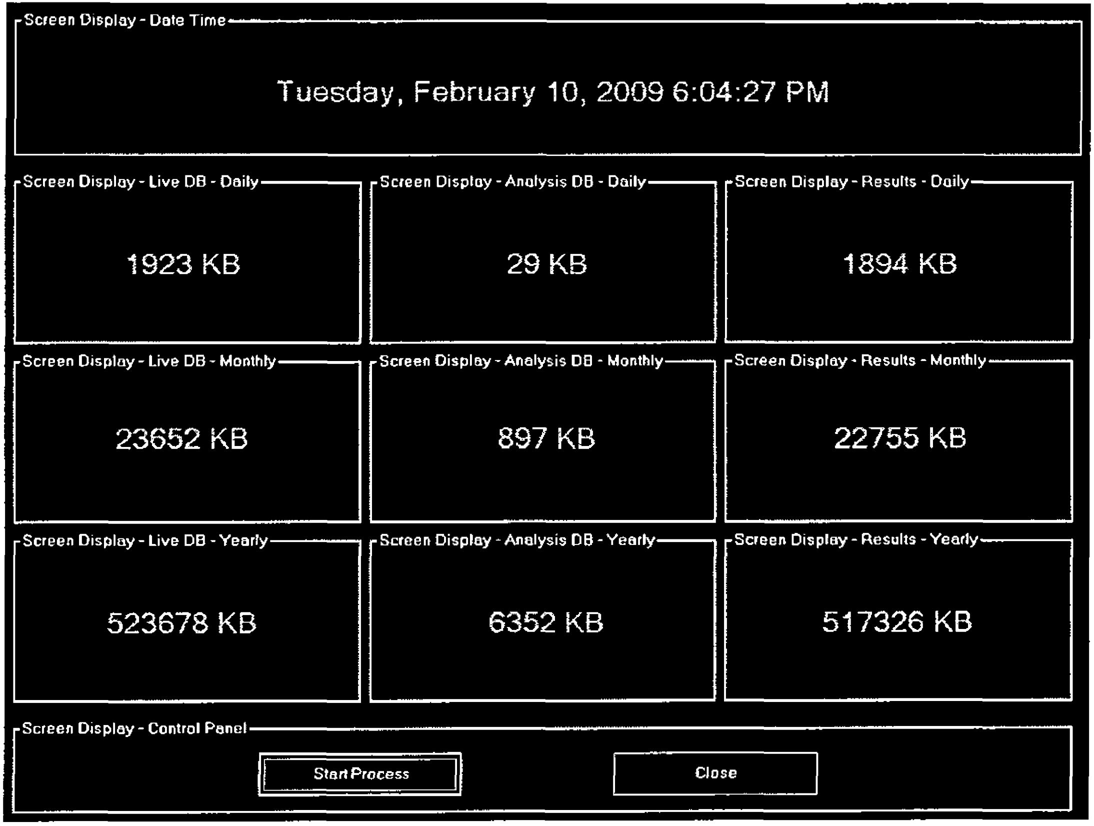

Screen Display - Date Time
Tuesday, February 10, 2009 6:27:39 PM
Screen Display - Live DB - Daily
Live Storage
Screen Display - Analysis DB - Daily
Analysis Storage
Screen Display - Results - Daily
Storage Difference
Screen Display - Live DB - Monthly
Live Storage
Screen Display - Analysis DB - Monthly
Analysis Storage
Screen Display - Results - Monthly
Storage Difference
Screen Display - Live DB - Yearly
Live Storage
Screen Display - Analysis DB - Yearly
Analysis Storage
Screen Display - Results - Yearly
Storage Difference
Screen Display - Control Panel
Start Process
Close

Fig 6.1 Main Screen of Simulation Software

6.2 Result Screen

After Start Process is clicked the process will start running and after completion of the process the following Fig 6.2 screen will be displayed.



Screen Display - Date Time

Tuesday, February 10, 2009 6:04:27 PM

Screen Display - Live DB - Daily: 1923 KB

Screen Display - Analysis DB - Daily: 29 KB

Screen Display - Results - Daily: 1894 KB

Screen Display - Live DB - Monthly: 23652 KB

Screen Display - Analysis DB - Monthly: 897 KB

Screen Display - Results - Monthly: 22755 KB

Screen Display - Live DB - Yearly: 523678 KB

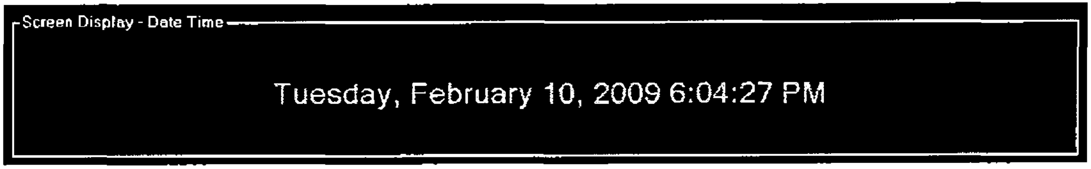
Screen Display - Analysis DB - Yearly: 6352 KB

Screen Display - Results - Yearly: 517326 KB

Screen Display - Control Panel

Fig 6-2 Result Screen

Fig 6-3 shows the current date time for reference purpose.

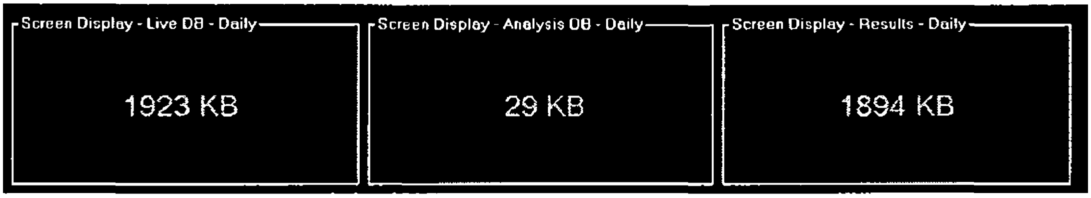


Screen Display - Date Time

Tuesday, February 10, 2009 6:04:27 PM

Fig 6-3 Date and Time Panel

Fig 6-4 shows times for Daily data query.



Screen Display - Live DB - Daily

1923 KB

Screen Display - Analysis DB - Daily

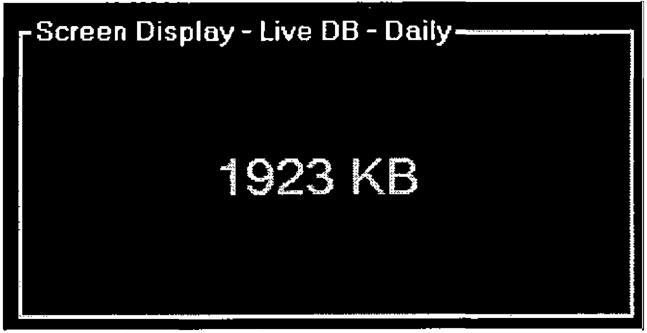
29 KB

Screen Display - Results - Daily

1894 KB

Fig 6-4 Daily Analysis

Fig 6-4-a shows the initial storage results for daily data

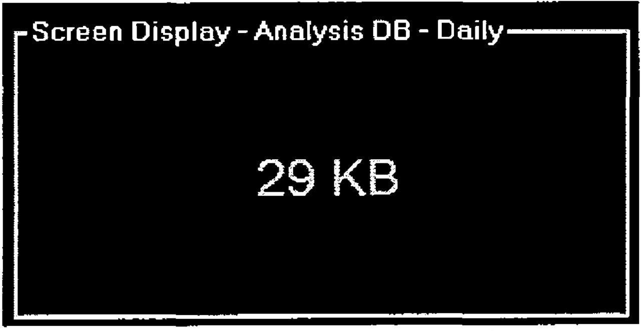


Screen Display - Live DB - Daily

1923 KB

Fig 6-4-a Initial Storage Panel

Fig 6-4-b shows the final storage results.

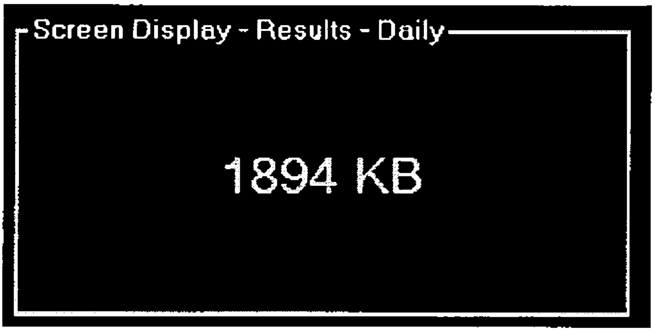


Screen Display - Analysis DB - Daily

29 KB

Fig 6-4-b Final Storage Panel

Fig 6-4-c shows the storage difference results in OLTP and enhance schema

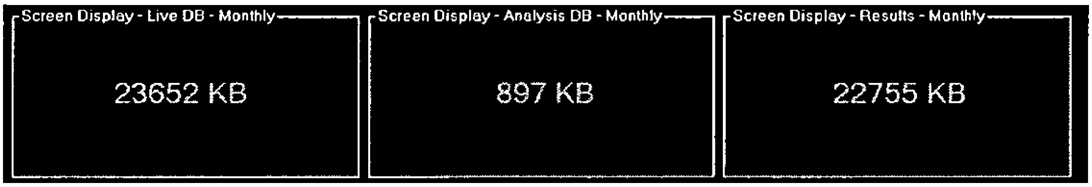


Screen Display - Results - Daily

1894 KB

Fig 6-4-c Storage Difference Panel

Fig 6-5 shows the analysis for monthly data query



Screen Display - Live DB - Monthly

23652 KB

Screen Display - Analysis DB - Monthly

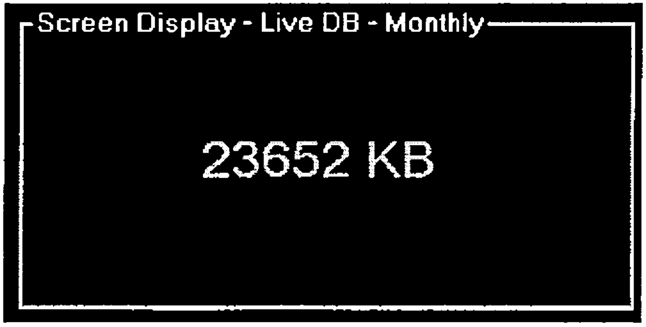
897 KB

Screen Display - Results - Monthly

22755 KB

Fig 6-5 Monthly Analysis

Fig 6-5-a shows the initial storage for monthly data.

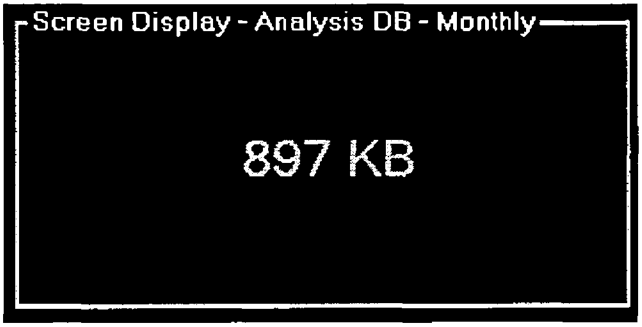


Screen Display - Live DB - Monthly

23652 KB

Fig 6-5-a Initial Storage Monthly Data Query Panel

Fig 6-5-b shows the final Storage for monthly data.

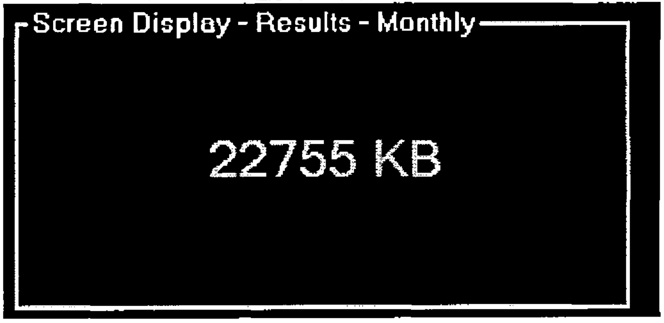


Screen Display - Analysis DB - Monthly

897 KB

Fig 6-5-b Final Storage for Monthly Data Query Panel

Fig 6-5-c shows initial storage for monthly data

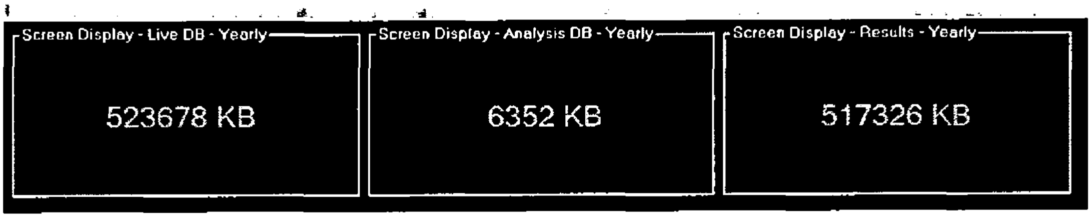


Screen Display - Results - Monthly

22755 KB

Fig 6-5-c Result Panel

Fig 6-6 shows the Yearly Data storage.



Screen Display - Live DB - Yearly

523678 KB

Screen Display - Analysis DB - Yearly

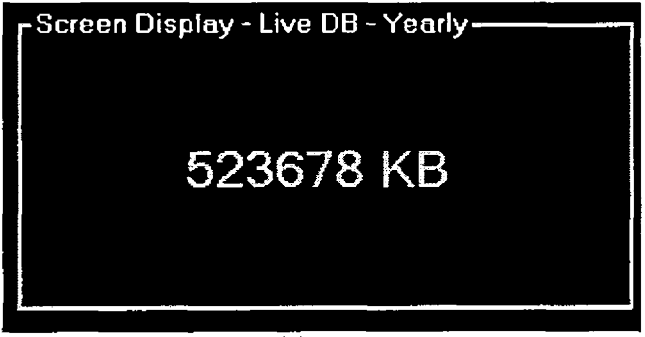
6352 KB

Screen Display - Results - Yearly

517326 KB

Fig 6-6 Storage Panel for Yearly Data

Fig 6-6-a shows the yearly initial storage .

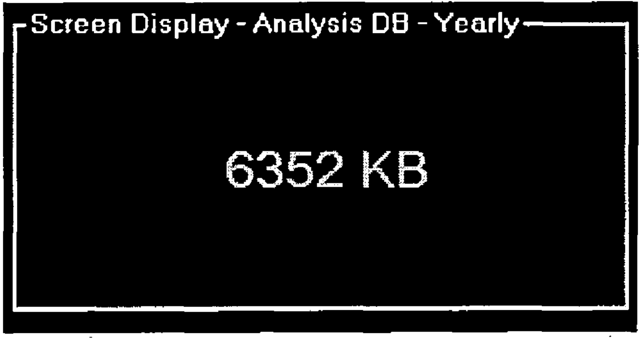


Screen Display - Live DB - Yearly

523678 KB

Fig 6-6-a Yearly initial storage

Fig 6-6-b shows the yearly final storage.

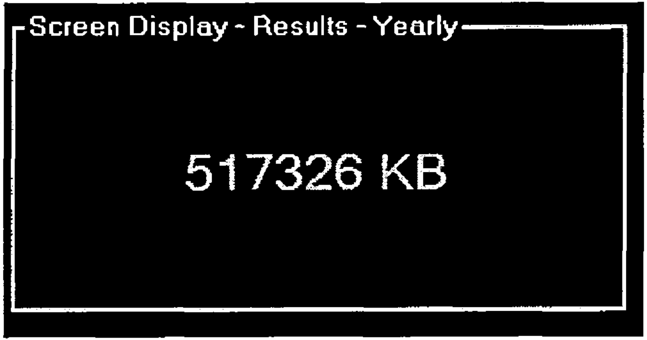


Screen Display - Analysis DB - Yearly

6352 KB

Fig 6-6-b Yearly final storage data

Fig 6-6-c shows the difference of storage for yearly data.



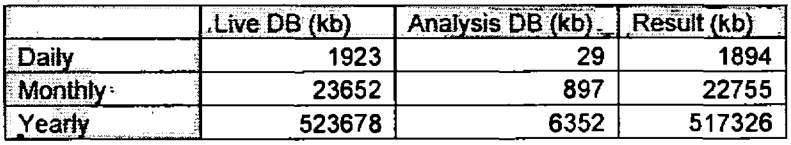
Screen Display - Results - Yearly

517326 KB

Fig 6-6-e Storage Comparison Panel for Yearly data

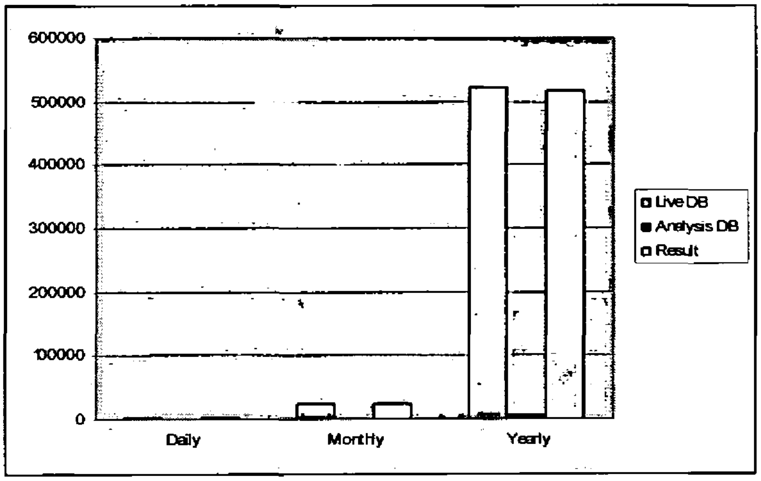
6.3 Comparison Table and Graph

Table 6.1 and Graph 6.1 shows the comparison of Data in Live and Analysis DB.



	Live DB (kb)	Analysis DB (kb)	Result (kb)
Daily	1923	29	1894
Monthly	23652	897	22755
Yearly	523678	6352	517326

Table 6.1 Storage Comparison Table



Graph 6-1 Storage Comparison Graph

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7. Conclusion and Future Enhancements

The simulation software has been tested on small scale and with limited volume of data. However as compared to the OLTP system the Schema Enhancement Method shows a very good result for Storage and Cost usage.

7.1 Conclusion

By applying the Schema Enhancement Method we have achieved our target i.e., OLTP system that supports Mining features and uses very less Storage and Cost as compared to the Conventional system. The Simulation Process also shows the same results and proves our research.

After that we have produced the research paper titled "Storage and Cost efficient mining on an OLTP System using Schema Enhancement Method" refer to Appendix B-1 which has been published in Euro Journal European Journal of Scientific Research
ISSN 1450-216XVol.19 No.3 (2008), pp.435-437
© EuroJournals Publishing, Inc. 2008
http://www.eurojournals.com/ejsr.htm
Research paper has been attached in Appendix C

7.2 Future Enhancements

The research area is still open because we have only discussed the Storage and Cost Efficiency aspect of the Schema Method and Performance Efficiency has already been discussed earlier [1].

APPENDIX A

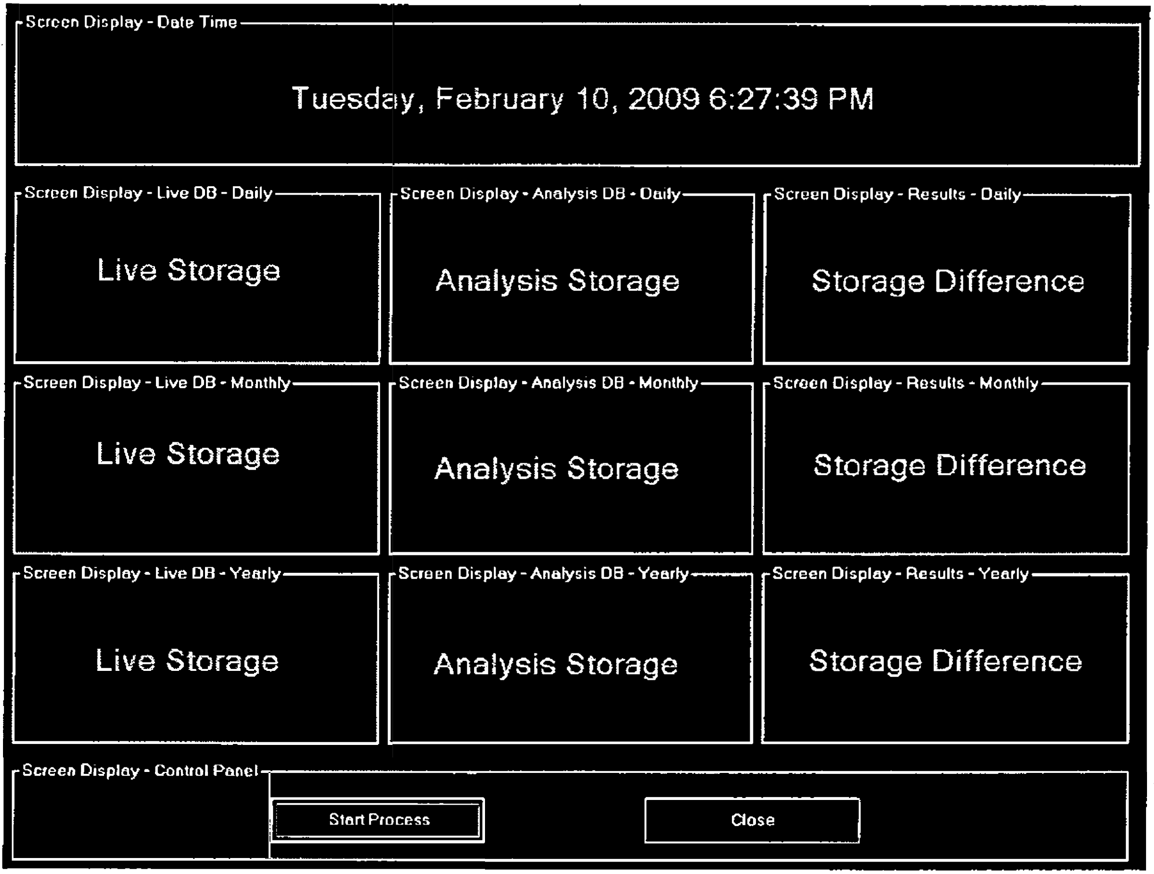
USER MANUAL

A. User Manual:

Following is the description of software that is being used to compare the results. The screenshots and their descriptions will be useful to understand the software and its working.

A-1 Main Screen

Test Run.exe Application is executed from the Application folder and following screen Fig A-1 is displayed.

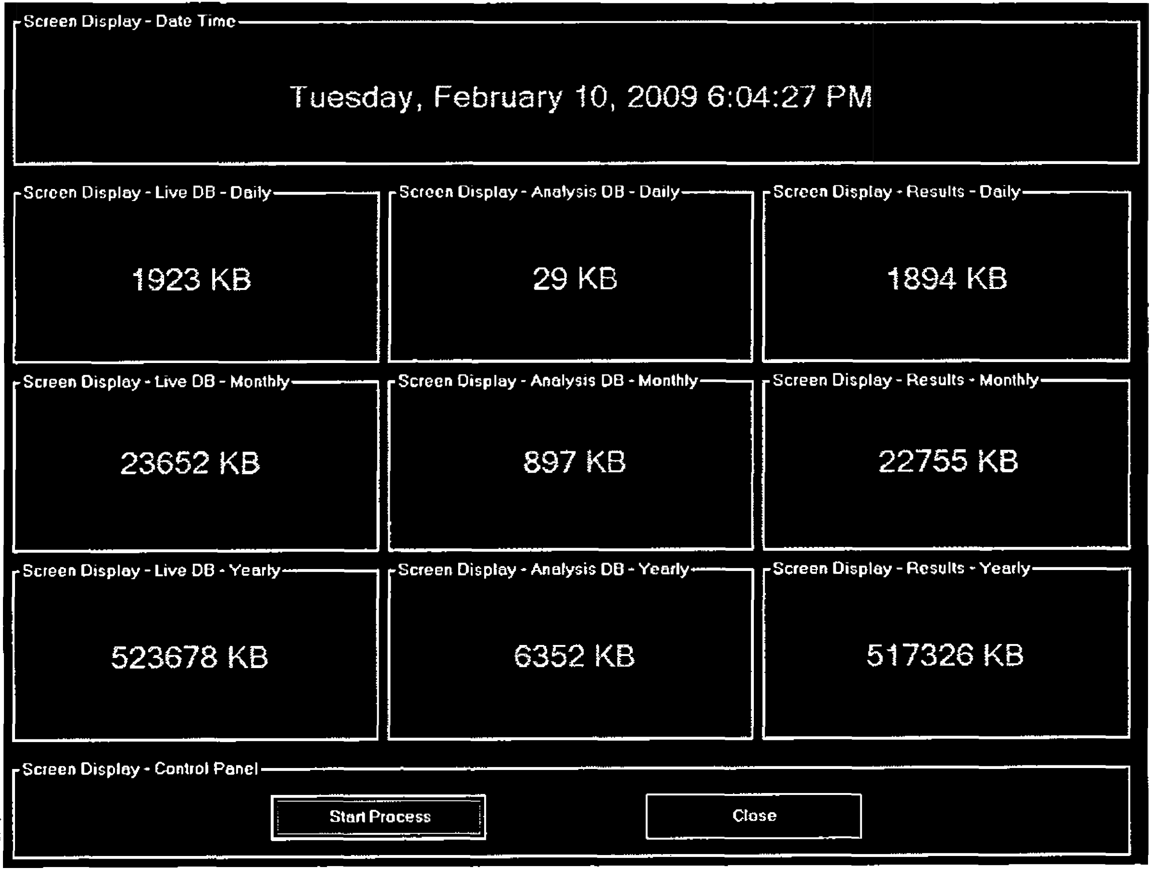


Tuesday, February 10, 2009 6:27:39 PM

Screen Display - Date Time
Screen Display - Live DB - Daily
Live Storage
Screen Display - Analysis DB - Daily
Analysis Storage
Screen Display - Results - Daily
Storage Difference
Screen Display - Live DB - Monthly
Live Storage
Screen Display - Analysis DB - Monthly
Analysis Storage
Screen Display - Results - Monthly
Storage Difference
Screen Display - Live DB - Yearly
Live Storage
Screen Display - Analysis DB - Yearly
Analysis Storage
Screen Display - Results - Yearly
Storage Difference
Screen Display - Control Panel
Start Process
Close

Fig A.1 Main Screen of Simulation Software

In Fig A.2 Result screen will appear after click Start Process Button.



Screen Display - Date Time

Tuesday, February 10, 2009 6:04:27 PM

Screen Display - Live DB - Daily	Screen Display - Analysis DB - Daily	Screen Display - Results - Daily
1923 KB	29 KB	1894 KB

Screen Display - Live DB - Monthly	Screen Display - Analysis DB - Monthly	Screen Display - Results - Monthly
23652 KB	897 KB	22755 KB

Screen Display - Live DB - Yearly	Screen Display - Analysis DB - Yearly	Screen Display - Results - Yearly
523678 KB	6352 KB	517326 KB

Screen Display - Control Panel

Fig A.2 Result Screen

APPENDIX B

REFERENCES AND BIBLIOGRAPHY

B. References and Bibliography

B.1 Books

[1]. **Decision Support Systems and intelligent Systems**. 5th edition. By Efraim Turban, Jaye. Aronson, Prentice Hall, New Jersey, 1998

[2]. **Data Mining: Concepts and Techniques**. By Jiawei Han, Micheline Kamber. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, August 2000, ISBN 1-55860-489-8

[3]. **Data Mining: Introductory and Advanced Topics**. By Margaret H. Dunham. Publisher: Prentice Hall; 1st edition (August 22, 2002), ISBN: 0130888923

[4]. **Database Systems: A Practical Approach to Design, Implementation, and Management**. By Thomas Connolly, Carolyn Begg. (3rd Edition), Publisher: Addison Wesley; 3 edition (August 1, 2001) , ISBN: 0201708574

[5]. **Modern Database Management** (7th Edition) by *Jeffrey A. Hoffer,* Mary Prescott, Fred McFadden, Prentice Hall; (April 6, 2004), ISBN: 0131453203

[6]. **Decision Making and Problem Solving**, Herbert A. Simon and Associates, 1986, National Academy of Sciences. Published by National Academy Press, Washington, DC.

[7]. **Building the Data Warehouse**, W.H. Inmon, Katherine Schowalter, 1996, 2nd Edition

B.2 References

[1] Muhammad Hasan Rasheed, Muhammad Imran Saeed and Dr. M Sikandar Hayat Khiyal
"Performance Efficient Mining on an OLTP System using Schema Enhancement Method"
Information Technology Journal 6 (4): 589-592, 2007

[2]. Erik Riedel, Christos Faloutsos, Gregory R. Ganger and David F. Nagle "**Data mining on an OLTP system (nearly) for free**
"ACM SIGMOD international conference on Management of data archive, Dallas, Texas, United States ISSN: 0163-5808 (2000)

[3]. Clay Rehm, Joe Oates and David Marco
"One database model for OLAP and OLTP"
Published in DM Review Online, DMReview.com (2002)

[4]. Jiawei Han, Yongjian Fu, Yue Huang, Yandong Cai and Nick Cercone.
"DBLearn: A System Prototype for Knowledge Discovery in Relational Databases."

[5]. Jiawei Han, Jenny Y. Chiang, Sonny Chee, Jianping Chen and Qing Chen.
"DBMiner: A System for Data Mining in Relational Databases and Data Warehouses".

[6]. Zhang, T., Ramakrishnan, R. and Livny, M.
"BIRCH: A New Data Clustering Algorithm and Its Applications"
Data Mining and Knowledge Discovery 1 (2), 1997.

[7]. Widom, J. **"Research Problems in Data Warehousing"** CIKM, November 1995.

[8]. Riedel, E., Gibson, G. and Faloutsos, C.
"Active Storage For Large-Scale Data Mining and Multimedia"
VLDB, August 1998.

[9]. Paulin, J. **"Performance Evaluation of Concurrent OLTP and DSS Workloads in a Single Database System"**
Master's Thesis, Carleton University, November 1997.

[10]. Guha, S., Rastogi, R. and Shim, K.
"CURE: An Efficient Clustering Algorithm for Large Databases"
SIGMOD, 1998.

[11]. Chaudhuri, S. and Dayal, U.
"An Overview of Data Warehousing and OLAP Technology"
ACM SIGMOD Record, March 1997.

[12]. Fayyad, U. **"Taming the Giants and the Monsters: Mining Large Databases of Knowledge"**
Database Programming and Design, March 1998.

APPENDIX C

PUBLICATION

European Journal of Scientific Research
ISSN 1450-216XVol.19 No.3 (2008), pp.435-437
© EuroJournals Publishing, Inc. 2008
http://www.eurojournals.com/ejsr.htm

Storage and Cost Efficient Mining on an OLTP System Using Schema Enhancement Method

Syed Mubashir Hasan
Department of Computer Science, International Islamic University
H-10, Islamabad, Pakistan

M. Sikander Hayat Khiyal
Chairperson Department of Computer Science/Software Engineering
Fatima Jinnah Women University, The Mall, Rawalpindi, Pakistan

Abstract

Mining process has been in research now-a-days. There have been many efforts done to improve the DSS solution and make it reliable and practical for OLAP (DWH) systems however there need a lot of efforts to implement the same for OLTP systems. Schema Enhancement method to improve the performance of the system has been very practical [1] however by applying the same method, Storage and Cots of OLTP can also be decrease. The method will be applied to the normal OLTP system and the Mining module developed by this method will be the part of actual schema of OLTP.

Keywords: Online Transaction Processing, Online Analytical Processing, Decision Support System, Performance Efficient Mining on an OLTP System using Schema Enhancement Method [1].

1. Introduction

OLTP systems are in high demand in developing countries like Pakistan. These systems are implemented in shops and small factories. These systems are very helpful to run the daily business and help improve the efficiency of the business and by applying the DSS [1], efficient business forecasting is done without implementing high cost DSS software.

In the research paper "performance enhancement mining on an OLTP system using Schema Enhancement Method", the performance effective aspect of the Schema Enhancement Method is discussed. The system become very efficient and query time decreases thus resulting in the effective solution for an OLTP environment.

Storage and cost is also one of the issues for OLTP systems, as the volume of daily business is very huge, it become very difficult to manage the data after 3-4 years and it is also not practical to save all the data in the system instead only some data is required for reporting and business foresting purposes. Schema Enhancement method helps to achieve these targets by modifying the schema to include the DSS module in the OLTP system normal function of the system is not affected by this enhancement.

2. Case Study (UFone PTML Pakistan, Image Processing System)

UFone, Pakistan Telecommunication Mobile Limited, provides Mobile services in the country. Company has its Head Office in Islamabad. Company has developed an Image Processing System for the Data entry of CSAF. These forms are filled and submitted by the customers at the time of new SIM purchase. It also benefits two folds i.e. data availability for the PTA and relevant government departments and the elimination of the manual data entry of the subscriber antecedents (through the ICR technique).

Schema Enhancement was applied on the existing system without making any changes in the software. In the scanning system, complete data is loaded including the Image of the form and identity card copies (front and rear side) of the customer. Image is stored for permanent basis after QC is done from data. After the process text data is not needed and can safely be deleted from the system however for reporting purpose it is not deleted. After the implementation of Enhanced Schema, separate space is allocated and data for reporting module (DSS) is loaded in that schema. This schema is a part of actual system however data is loaded on regular interval time. This save a lot of storage as text data is deleted and only reporting data is saved. Cost of hardware is also saved as there need less storage and no high fi system is required to run reporting.

System efficiency is increased and less storage is required to save the reporting data.

Table 1.1: Comparison Report

f	No of Records	Bytes	Kilo Bytes	Mega Bytes
Conventional OLTP	5680435	5091885056	4972544	4856
Enhanced Schema	5608	20971520	20480	20

Table 1.2: Yearly Analysis

Year of Sale	No. of Records
2004	20806
2005	80146
2006	1001345
2007	1680435

3. Storage and Cost Efficient Schema Enhancement Method

Storage and cost efficient system is the demand of every business of the 21th century. Storage efficient method as compared to the conventional OLTP is very practical approach for some organizations. As from the comparison it is clear that the storage required to store the summary data that can be used for future reporting and analysis purpose occupy very less hard drive space as compared to the actual data and thus reducing the cost of the overall system.

4. Conclusion

Schema Enhancement Method has shown very positive results on this implementation. We are hopeful that the results on other systems will also be very practical. This process is very easy to implement and the normal structure of the system is not affected by the change.

5. Acknoledgement

We are thankful to Ufone, Pakistan for providing us the information we have mentioned in our paper. We are also thankful to Dean and Head of Computer Science Department, International Islamic University Islamabad, Pakistan for providing us the working environment and resources for completing our research.

References

[1] Muhammad Hasan Rasheed, Muhammad Imran Saeed and Malik Sikander Hayat Khiyal. **"Performance Efficient Mining on an OLTP System Using Schema Enhancement Method"**. Infomation Technology Journal 6(4): 589-592, 2007.

[2] Erik Riedel, Christos Faloutsos, Gregory R. Ganger and David F. Nagle. **"Data Mining on an OLTP system (nearly) for free"**. ACM SIGMOD International Conference on Management of Data Archive, Dallas, Texas, United States, ISSN: 0163-5808 (2000).

[3] Clay Rehm, Joe Oates and David Marco. **"One Database Model for OLAP and OLTP"**. Published in DM Review Online, DMReview.com (2002).

[4] Jiawei Han, Yongjian Fu, Yue Huang, Yandong Cai and Nick Cercone. **"DBLearn: "A System Prototype for Knowledge Discovery in Relational Databases"**. In Proc. ACM SIGMOD 1994 Int. Conf. on management of data. Page 516. Minneapolis, Minnesota, United States, May 24-27, 1994.

[5] J. Han, J.Y. Chiang, S. Chee, J Chen, Q. Chen, S Cheng, W. Wang, M. Kamber, K. Koperski, G. Liu, Y. Lu, N. Stefanovic, L. Winstone, B.B. Xia, O.R. Zaiane, S. Zhang and H. Zhu. **"DBMiner: A System for data mining in Relational Databases and Data Warehouses"**. Data mining research group, intelligent database system research laboratory, school of computing science, Simon Fraser University, British Columbia, Canada, V5A 1S6.