Mining on an OLTP System using Schema Enhancement Method

Developed By

Muhammad Hasan Rasheed

Muhammad Imran Saeed

Supervised By

DR. Malik Sikandar Hayat Khiyal

Department of Computer Science
International Islamic University,
Islamabad
(2006)

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

Department of Computer Science

International Islamic University Islamabad

Date: 31-08-2006

Final Approval

This is to certify that we have read the thesis submitted by **Muhammad Hasan Rasheed** 112-CS/MS/03 and **Muhammad Imran Saeed** 55-CS/MS/01. 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. Qasim Rind
Professor, Department of Computer Science,
Preston University,
Islamabad.

Internal Examiner
Mr. Muhammad Amir Aman
Lecturer, Department of Computer Science,
Faculty of Applied Sciences
International Islamic University,
Islamabad.

Due to on S/Leave

Supervisor
Dr. M. Sikandar Hayat Khiyal
Head, Department of Computer Science,
Faculty of Applied Sciences,
International Islamic University,
Islamabad.

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.

Dedicated To

The Most Beloved Hazrat Muhammad (SAW)

And My Family

Hasan

Dedicated To

The Most Beloved Hazrat Muhammad (SAW),

My Beloved Late Mother

My Motherland

And My Family

Imran

Declaration

We hereby declare that this Research ***"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 we have done this research with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our 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, we shall stand by the consequences.

Muhammad Hasan Rasheed

112-CS/MS/03

Muhammad Imran Saeed

55-CS/MS-01

Acknowledgement

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We are bound to thank **DR. Malik Sikandar Hayat Khiyal** for his kind behavior and guidance as supervisor. Beyond doubt, we would never have completed this task without his interest and strict standards of perfection.

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

We want to show our regards and greetings to all of our class fellows and friends with whom we spent a very nice and memorable time.

We also thank **Miss Zakia Jalil** for her assistance in this project.

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

Muhammad Hasan Rasheed

112-CS/MS/03

Muhammad Imran Saeed

55-CS/MS/01

Project In Brief

Project Title:	Mining on an OLTP System using Schema Enhancement Method
Undertaken By:	Muhammad Hasan Rasheed 112-CS/MS/03 Muhammad Imran Saeed 55-CS/MS/01
Supervised By:	DR. Malik Sikandar Hayat Khiyal
Start Date:	14th December 2005
Completion Date:	May, 2006
Tools & Technologies	SQL 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:	Pentium III (Celeron) 700 MHz Pentium II (Celeron) 333 MHz. Pentium 4 (Centrino) 1.6 GHz Xeon Server 3.2 GHz

Abstract

The basic idea of the project is to develop and test the performance of Mining on an Online Transactional Processing System (OLTP). OLTP systems are transaction based systems used to perform daily tasks, execute simple but thousands of queries daily. Mining on an OLTP System is possible however the cost of running the queries is not feasible, the query time increases and the performance of the System decreases. It is simply impossible to run the mining module during the business hours. Running the mining module will definitely compromise the performance of the System and might slow down the business.

Online Analytical Processing Systems are very easy to implement for Mining because the structure of the Data ware house is build in such a way that we can easily run complex queries and satisfies the need of management. However implementing a Data ware house is not an easy task. It takes a lot of financial and Human resources to invest and take a lot of time to mature. Such high cost and time factors are not feasible for an organization running OLTP environment.

There is a need to develop or to modify the existing OLTP Architecture that works in a normal way and also support the Mining Process without mush cost. The system that will works as the existing OLTP however if some changes are done in the architecture to support the Mining, the system will be very beneficial for the small and medium level organizations and business demands.

Schema Enhancement Method is proposed, implemented and tested on the OLTP. The structure of Database is modified without disturbing the System, according to the needs of the organization and mining module is introduced. The mining module is dynamic and can be changed at any time depends on the new requirements. The enhancement made the system very efficient and mining module can be run at anytime without disturbing the normal work of the Software.

TABLE OF CONTENTS

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

LIST OF FIGURES

Contents **Page No**

Contents **Page No**

LIST OF TABLES

Contents **Page No**

LIST OF GRAPHS

Contents **Page No**

CHAPTER 1

INTRODUCTION

1. Introduction

Healthy business competition has always been a mean, leading towards the success. It provides energy in the market, so every one strives for the best. The business competition is not a new phenomenon of the modern era, but in fact it has got its roots deep in the history. Businessman of every time, no matter what the size of the business is, tries to be the best, as compared to the others, in the market. Well, "being the best" is not to be achieved simply, but it requires extraordinary talent, deepest sincerity towards the work, maddening passion, the thirst to prove ones self, and above all, the ability to take the best decision in time [1].

In past, the decision making was supported mainly by the experience and luck. The people with vast experience in the market were considered to have monopoly/command in taking good timely decisions for their business. Sometimes, the decisions taken by the new comers also proved to be as a success, but it was only the outcome of their sheer luck. Because it was really hard to keep an eye over the entire market, analyze the packages provided by the competitors, and the customer's response over those packages, and then take your decisions accordingly. Those days, decision making was based on the observation. The shopkeeper was supposed to decide for the new stock to his shop only by daily checking the shelves of his shop. If the stock was present in the sufficient amount for the day, it was ok; otherwise, it was time for the new stock to be ordered [1].

The problem with observation based decision making was, we were left with no summary of our business transactions. Secondly, it rendered us with a very limited past record of our business dealings. Also, in that technique, all decisions were wholly dependent over the single person and no one else was able to take decisions on his behalf, as all the records were only stored in the shop owner's brain only. And above all, the observation method only worked for the decision of a small shop, business, with only few daily transactions. For wide spanned /spread businesses, having millions/hundreds of transactions per second, simultaneously at multiple branches, this method failed to support the decision making [1].

With the emergence of foreign market/products in the local markets, the situation became more complex, because now the customers had more choices available for him. The need for proper, accurate, and timely decisions became more prominent. There was a desperate need for some system where the executives/top management was not just provided with the summary of all sales of all the branches, about the performance summary of a specific branch, product, employee, over the user's specified time, but also have a look over what the competitors are doing and what are they up to? What are the keys of their success and where do we lack behind?

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? Hence, planning implies decision making [1].

The emergence of information technology has changed the structure of whole system of decision making. The computer systems can provide the top management with all the necessary data, to enable them make a smart decision, with a single mouse button click. Also, there are systems available that 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 the solution of a nonstructural management problem for improved decision making. It uses data, 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 ease the job of managers to a greater extends. All what they are required 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 which consequently will result into good business decisions. It's not only necessary to gather the data of one's own organization to facilitate the decisions, but also we should keep the data of our competitors, to analyze their keys to success and removing flaws from our products. Another important area about which an organization must keep information is their customers. The customer's demographics and their buying patterns will help an organization to forecast about the future trends, regarding their products.

Decisions are nowadays wholly dependent on the 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 some /many products vary with the changing climatic/seasonal conditions. For example, in summers, the sales of ice cream increases. So there is the greatest demand for ice cream in summers by the customers. Similarly, the sales of refrigerators also increase during summers, whereas in winters, the sale of both the 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. So the decisions dependent of these types of data are cyclic in nature, i.e., they keep on changing with seasons. Such decisions are called cyclic decisions.

1.1.2 Linear Decisions:

These are the decisions that could be taken by simply looking at the previous sales records. It means that the past values of data are going to help about its future values. For example, the eating habits of people tell that the business of fast food will keep on rising high, irrespective of the climatic conditions.

To facilitate both types of decisions, there is a definite need of data. More the data, accurate will be the decisions.

1.2. Available Tools

It's clear now that for decision making, there is a definite need of data. On the basis of available data, different analyses are being performed. And as a result of these analyses, a correct and in time decision is taken. Mostly, for analysis, huge amount of data gluts are being used, which after performing certain operations on it, give the user some idea, trend, thus facilitating the user to reach certain decision. There are tools available in the market for doing analysis on the data, which 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 be easily taken from a University System, where we have different departments, i.e., Accounts Department, Examination Department, and Students Affairs Department. If we collect the historical data of a particular department from all sources, and clean them, transform into a uniform format, and then load it in our computer, in order to be able to get analytical queries answered, then it means we are building separate Data Marts for each department of the University System. All Data Marts of an organization collectively form a Data Warehouse. The data loaded on the computer system is supposed to be summarized data, like if two attributes of STUDENT Entity class are in the format "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, 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 is "How to retrieve that data?" To satisfy this demand, we have Data Warehouses.

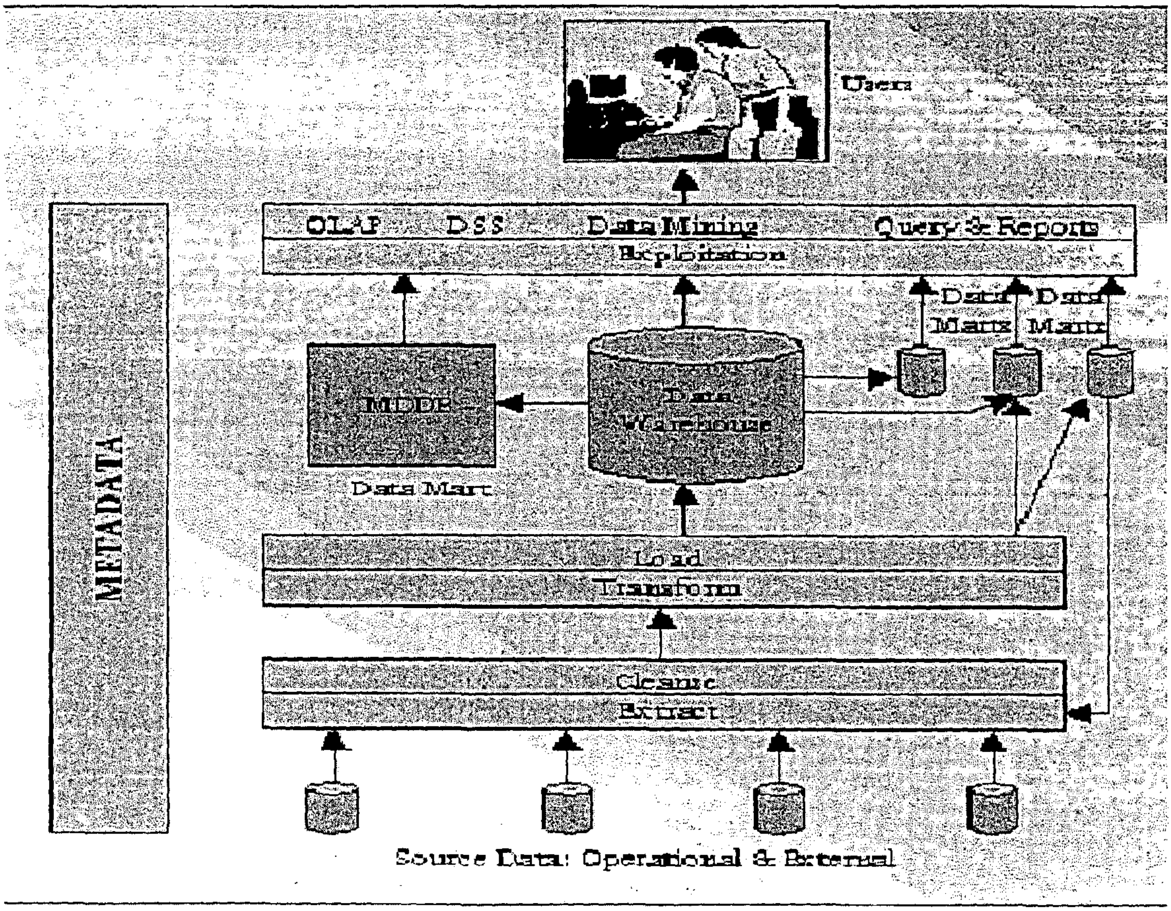


Fig 1-1 A Data warehouse Architecture

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.

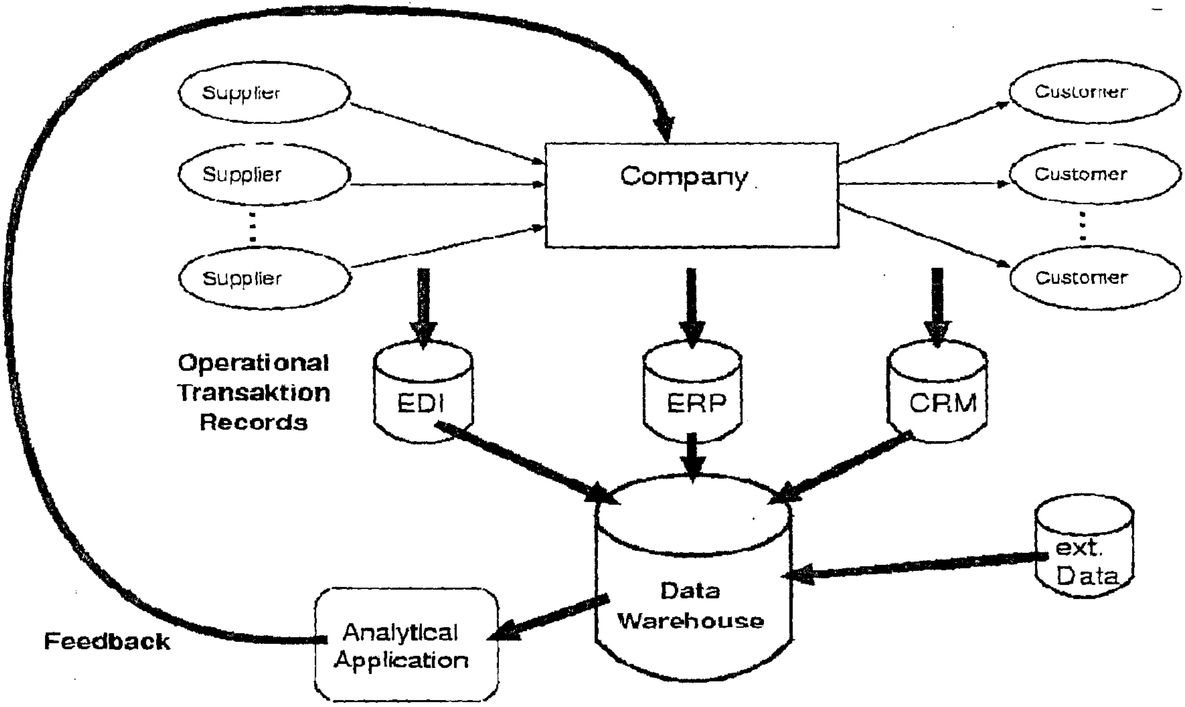


Fig 1-2 A Warehouse of a Company

1.2.3 OLAP :

Online Analytical Processing System (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.

Online Analytical Processing System (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.

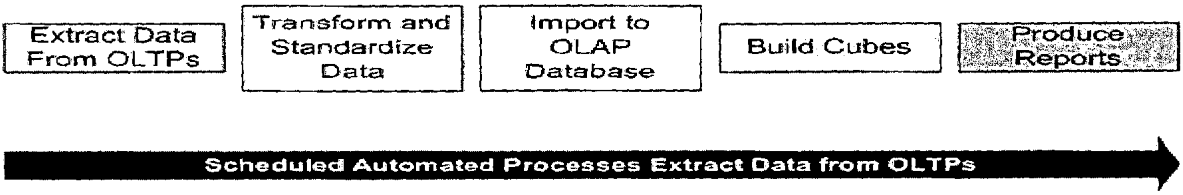

Extract Data From OLTPs
Transform and Standardize Data
Import to OLAP Database
Build Cubes
Produce Reports
Scheduled Automated Processes Extract Data from OLTPs

Fig 1-3 Sequence of Data formation for Decision Making

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. Database Management Systems (DBMS) is a technology to use or process that data with the help of Structured Query Language (SQL). 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 Extract, Transform and Load (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 Scope of the Project

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.

The overheads discussed above small and medium scale businesses can not afford to have a data warehouse and at the same time cannot stop the business demands. In order to run the online transactional processing system and at the same time required a mining module to run on the same system there is a need of some enhancements.

The Solution is to make 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

REVIEW OF LITERATURE

2. Review of Literature

Literature Survey is an important and unavoidable part of research, as without literature survey we cannot understand the scenario that till what point the researchers have reached and what are the loopholes in the topic and what can be enhanced in that area. We also have gone through multiple research papers, articles and books to find out the current scenario of this area and also to know the new aspects of data mining in an OLTP.. Following is an article that discusses the concept of Analysis on an OLTP system.

2.1One Database Model for OLTP and OLAP:

Rehm at el [2] 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 Kelly, David Marco, Joe Oates, Clay Rehm reply the question which are stated as below as it is.

Sid Adelman's Answer:

"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:

“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:

“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:

"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:

"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 1 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:

"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:

"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:

"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 work done 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 we are going to propose the OLTP Model 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.

CHAPTER 3

REQUIREMENTS ANALYSIS

3. Requirements Analysis

The requirement analysis is the first step towards developing software. Analysis must be performed in a systematic and correct manner so as to have as few mistakes as possible in the software and to have an end product completely fulfilling the expectations of the client. The reliability and the robustness of the software are highly dependent on the fact that the analysis is carried out properly. The main objective of this phase is to identify all possible requirements. Problems are identified and then a possible solution is proposed.

3.1 Problem Analysis

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

- OLTP Conventional Architecture remains the same. All the changes required for the Mining Module will be done independently on the same machine but in a different schema.
- Mining Module requirement gathering is done at any time during the life of the System depending upon the analysis requirement of the business.
- It is not necessary to develop the Schema Enhancement Modules at the time of initial Development.
- Once the mining requirements are known, the schema for the mining module designed and then the database is developed keeping in mind the data interface of the existing system.
- Mining module is always dependent on the data present in the existing system. And the data that is fed in the enhanced schema will come only from the existing system.
- The daily, Monthly and yearly data is stored on different tables to fast the retrieval process and make the system easy to understand.

- When we create the tables for the enhanced schema, we need to populate the new tables from the Operational System (OLTP). In case the system is developed and working earlier than the mining system, there is a need to transfer the data present in the OLTP, in order to Mine the Data in Enhanced Schema.
- The enhanced schema is populated through triggering processes or export modules that can be developed and run according to the ease in such a way that it does not affect the performance of Operational System.

3.2Use Case Analysis

Analysis of the project is presented in terms of use case diagrams indicating the actors and use cases in expanded format. This helps visualizing the work and indicating the system boundaries while presenting the functionalities. The Use Case Model describes the proposed functionality of the new system.

Use case depicts a set of scenarios that describing an interaction between a user and a system. Use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

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

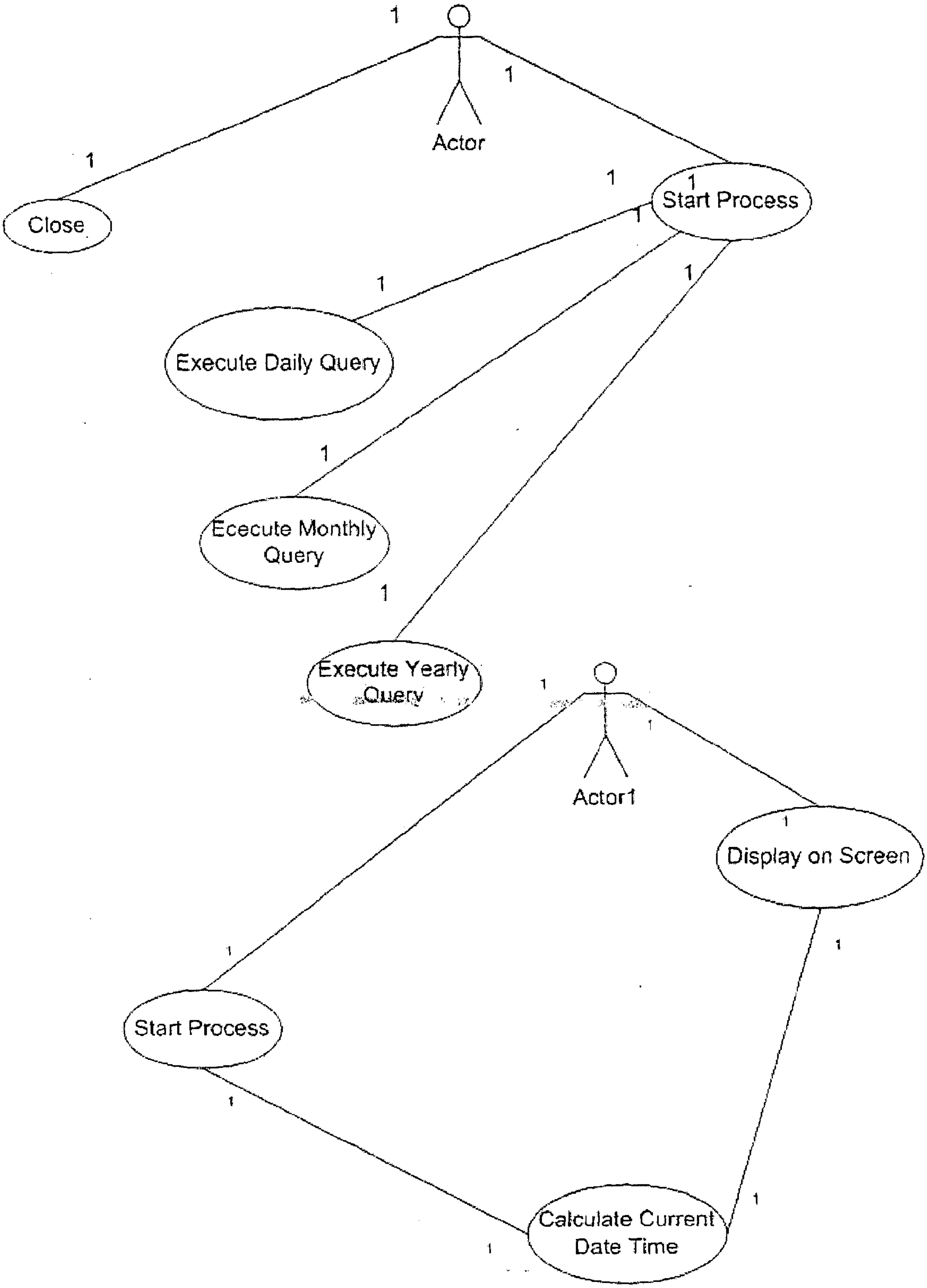


Fig 3-1 Use Case Diagram of Query Processing

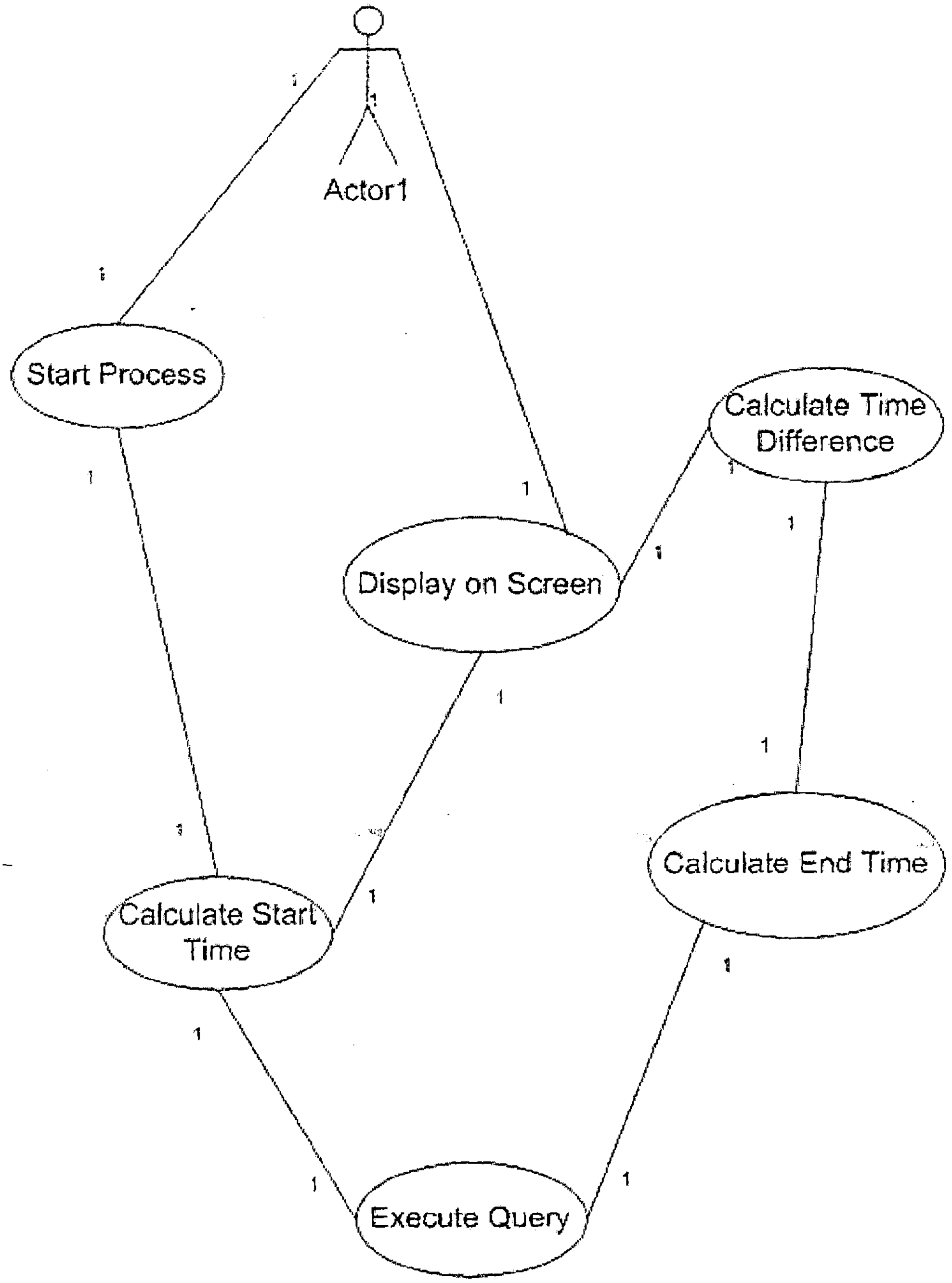


Fig 3-2 Use Case diagram of Time Calculation Process

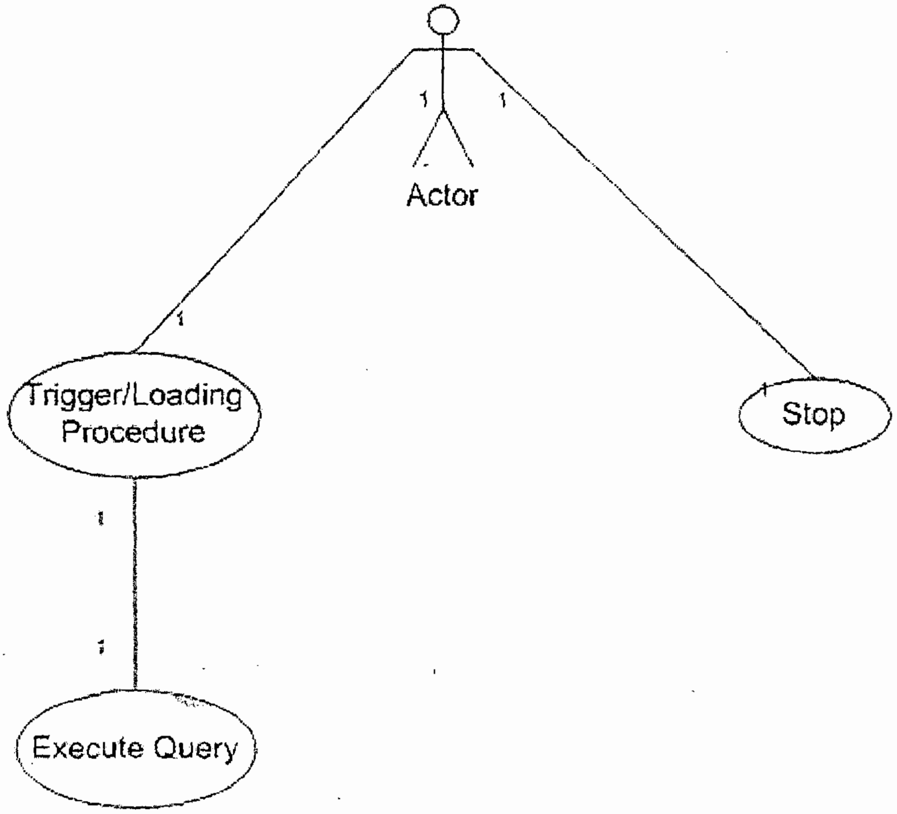


Fig 3.3 Use Case Diagram of Trigger or Loading Procedure

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 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 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 disconnects. 3. OS de allocates 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 click.

d) Post Condition: Connectivity Established.

e) Typical Course of Action:

Actor Action	System Response
1. Press Start Process Button	2. Connection String initializes. 3. Send request to SQL Server for connectivity. 4. Check user and password. 5. Establish Connectivity.

f) Alternate Course of Action:

Actor Action	System Response
No action	1a. SQL Server Error Displays 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.

d) Post Condition:

e) Typical Course of Action:

Actor Action	System Response
1. Press Start Process Buton	1. Initialize SQL Query 2. Initialize SQL Command 3. Execute SQL Query 4. Fetch Results using SQL Command.

f) Alternate Course of Action:

Actor Action	System Response
	3a. Error Message Displayed on screen.

3.2.1.5 Calculate Time Difference

a) Name: Calculate Time Difference

b) Actor: User

c) Pre-Condition: SQL Connectivity.

d) Post Condition:.

e) Typical Course of Action:

Actor Action	System Response
1. Press Start Process Button.	1. Calculate Star Time of Query 2. Display on screen. 3. Execute Query. 4. Calculate End Time. 5. Display on Screen 6. Calculate Time Difference. 7. Display 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 a software that will be used to verify the efficiency of proposed enhanced schema technique.

Object-Oriented Design (OOD) 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 (OOD) 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 efficiency in enhanced schema of OLTP.

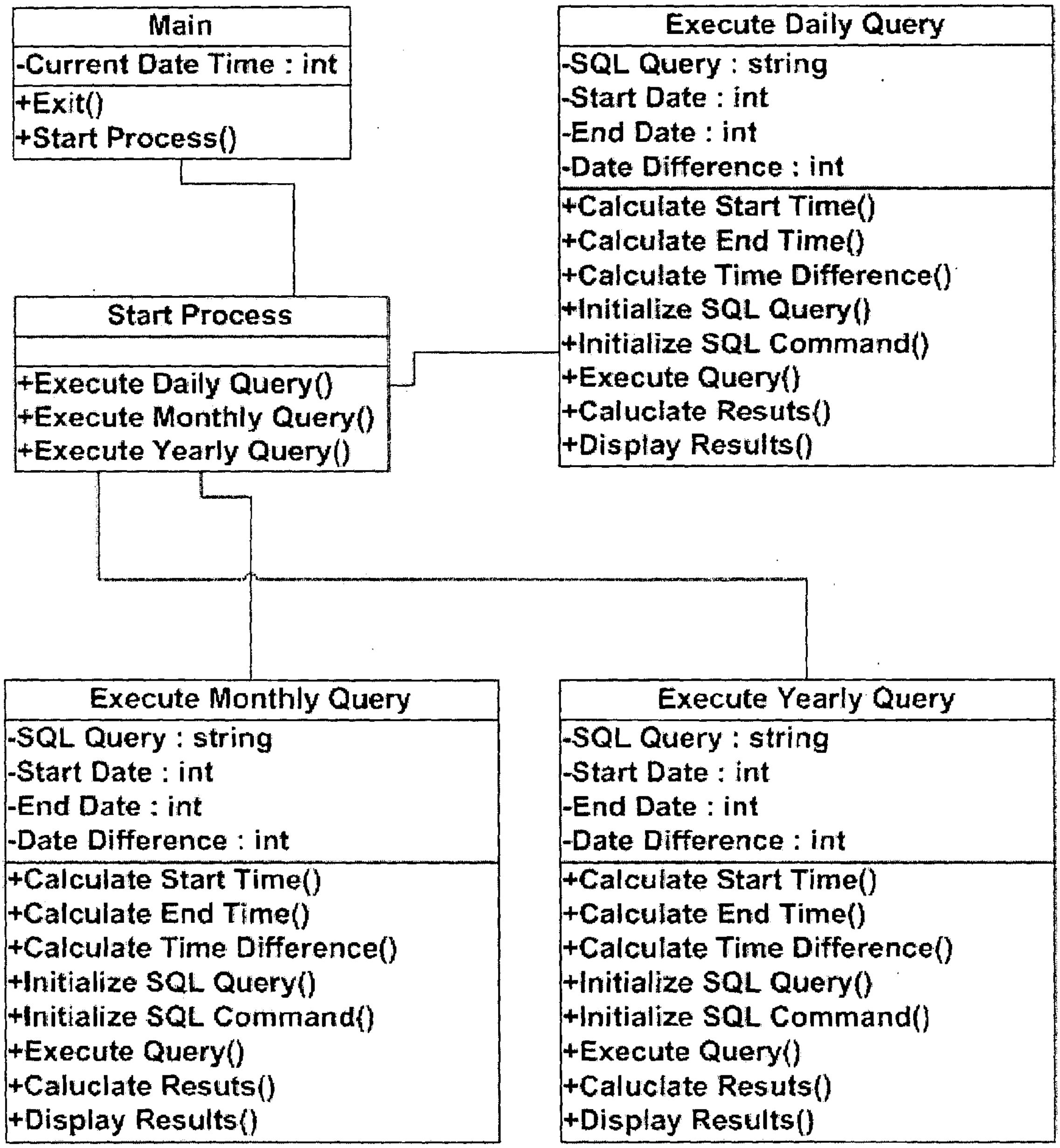


Fig 4-1 Class Diagram of Software Module

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.

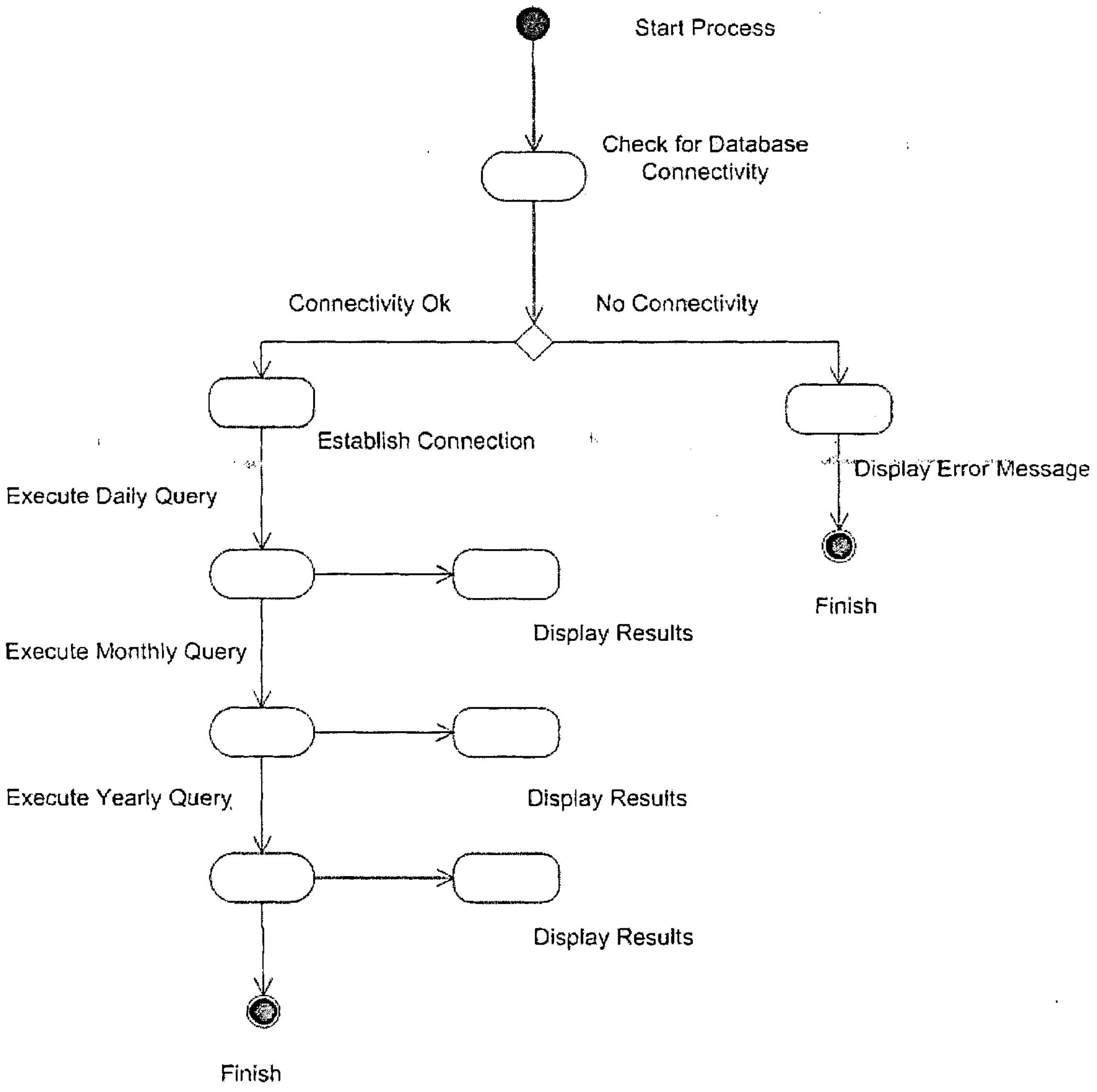


Fig 4-2 State Transition Diagram of Software Module

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.

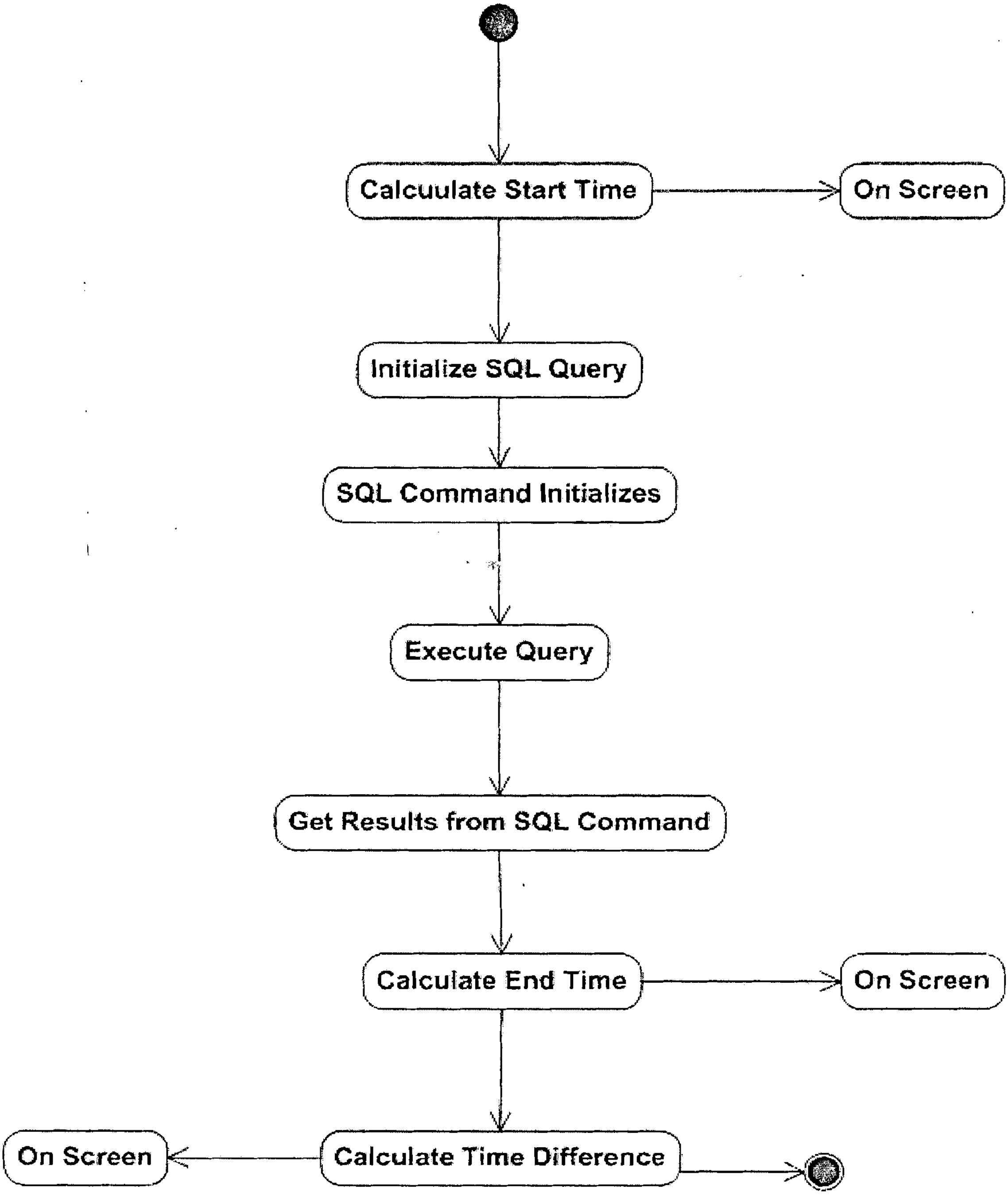


Fig 4-3 Detailed State Transition Diagram of Query Processing

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.

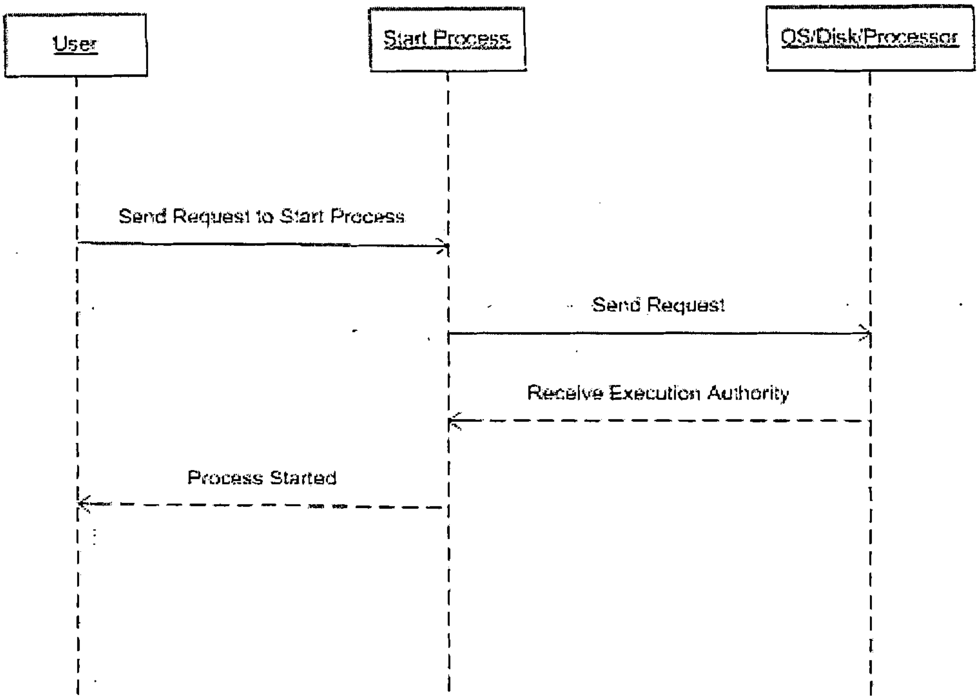


Fig 4.4-a: Start Process Sequence Diagram

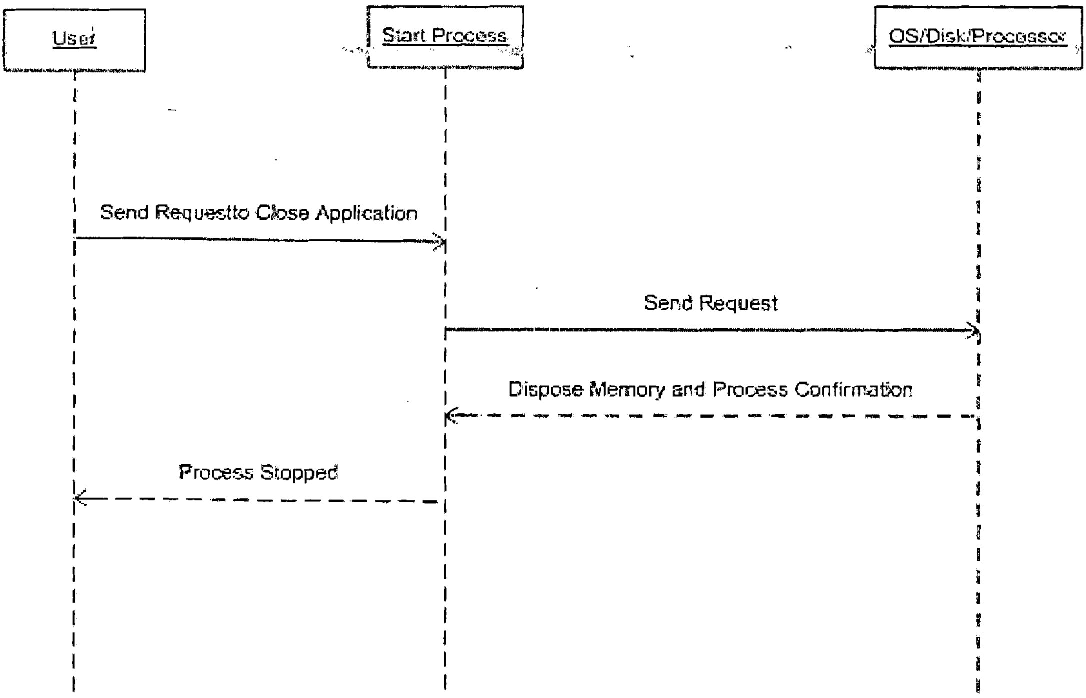

User
Start Process
OS/Disk/Processor
Send Requestto Close Application
Send Request
Dispose Memory and Process Confirmation
Process Stopped

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.

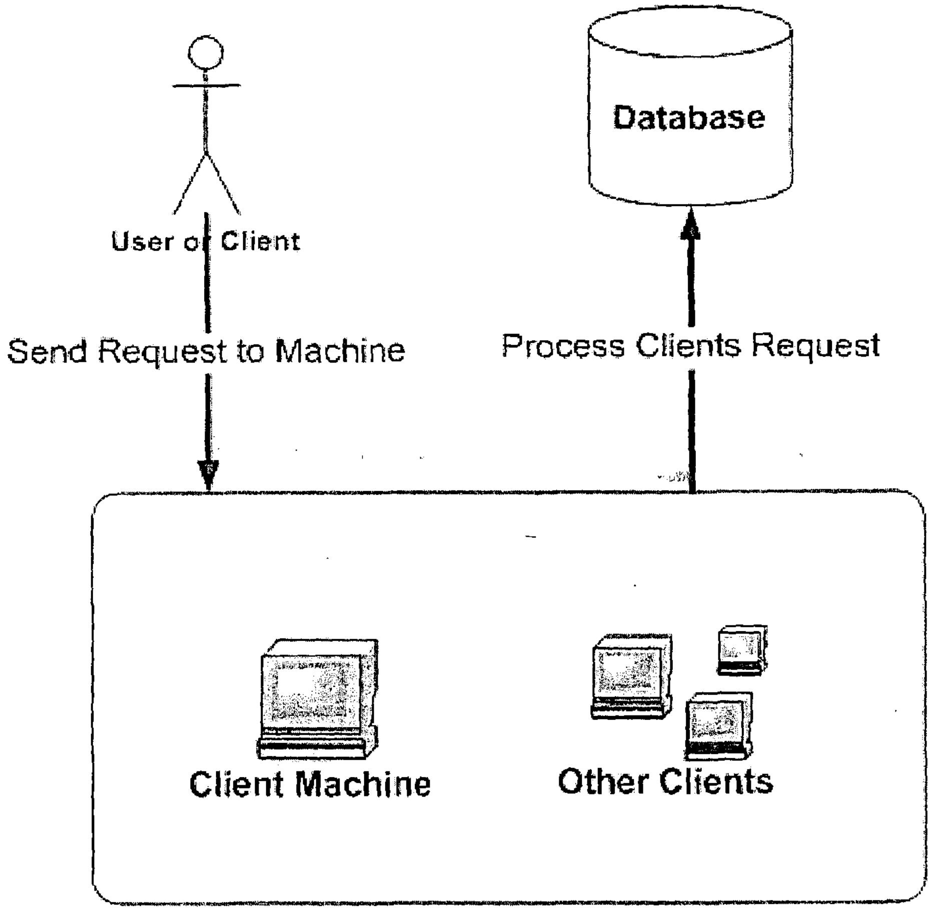


Fig 4.5: Architecture Design of OLTP

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.

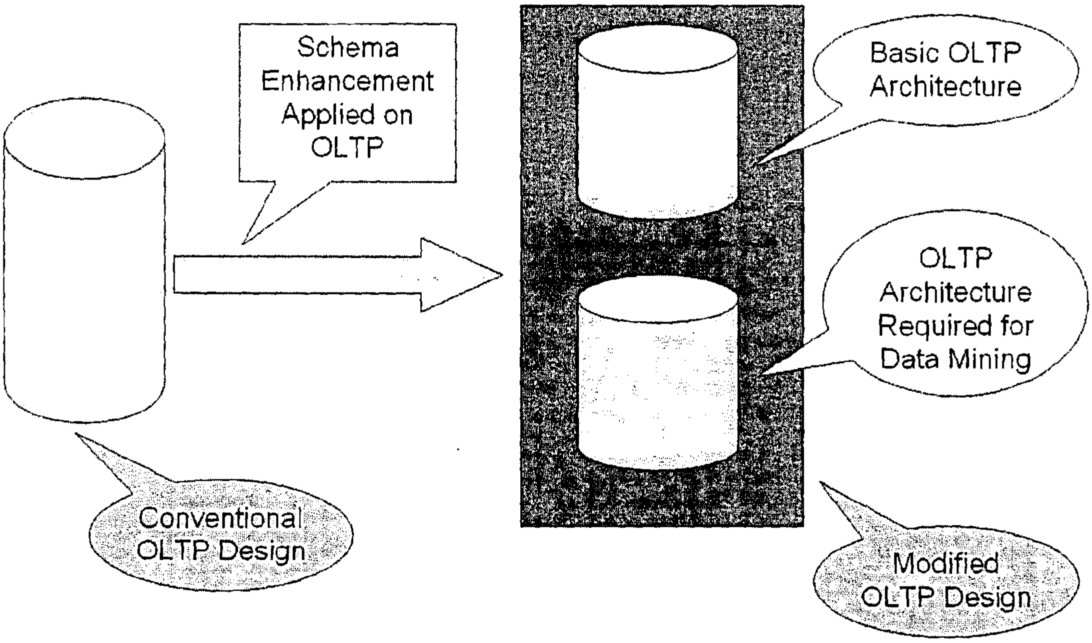

Schema Enhancement Applied on OLTP
Basic OLTP Architecture
OLTP Architecture Required for Data Mining
Conventional OLTP Design
Modified OLTP Design

Fig 4.6 Proposed Schema Enhancement Model

4.5 Case Study:

In order to support this idea we have taken the case study of one module of National Database Registration Authority (NADRA, Pakistan).

4.5.1 Data Collection System:

The Schema (ERD) of that module is given in Fig 4.7. This is the ERD of Data Collection 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 effect the efficiency of the operational system specially when analysis is required on data of many years.

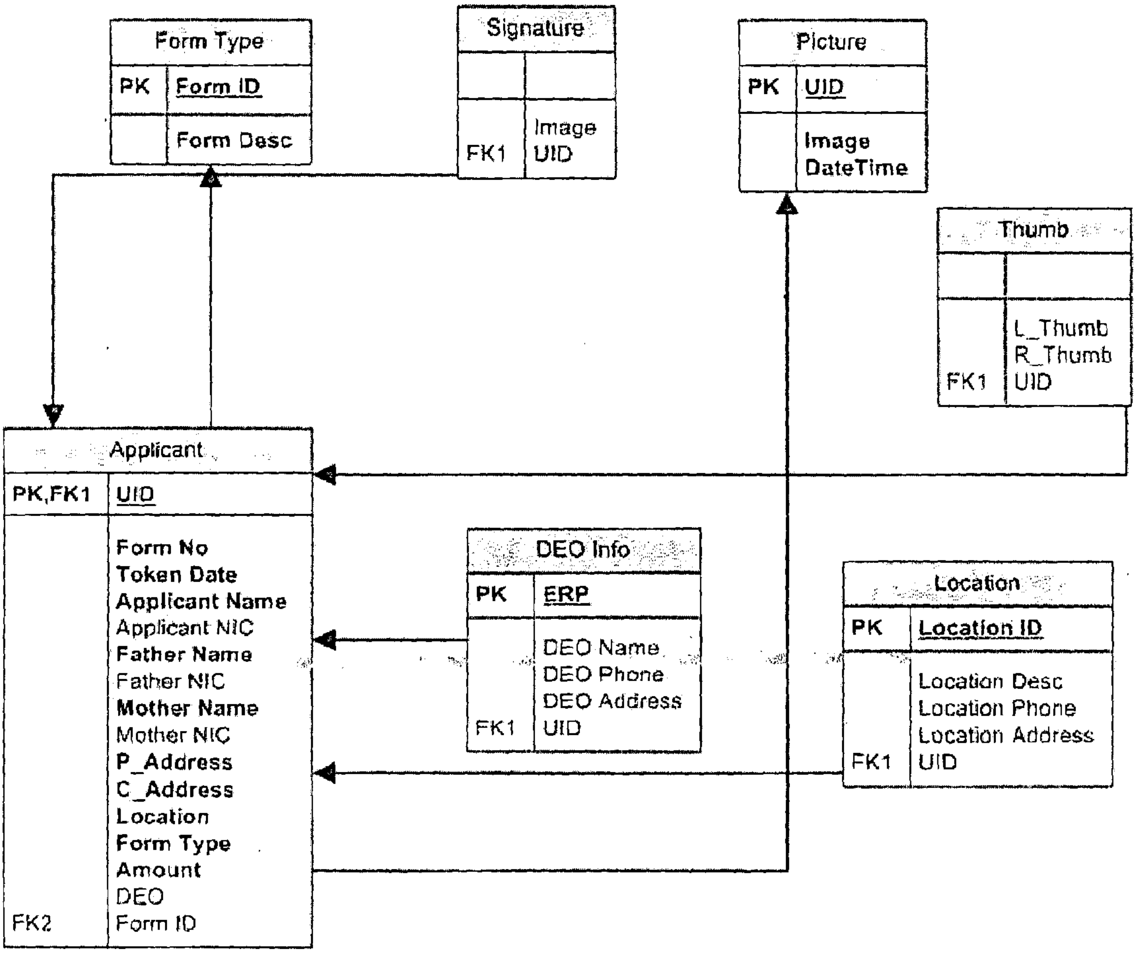


Fig 4.7: ERD of Operational System of one Module of NADRA.

4.5.2 Enhance Schema Model for Data Collection 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.

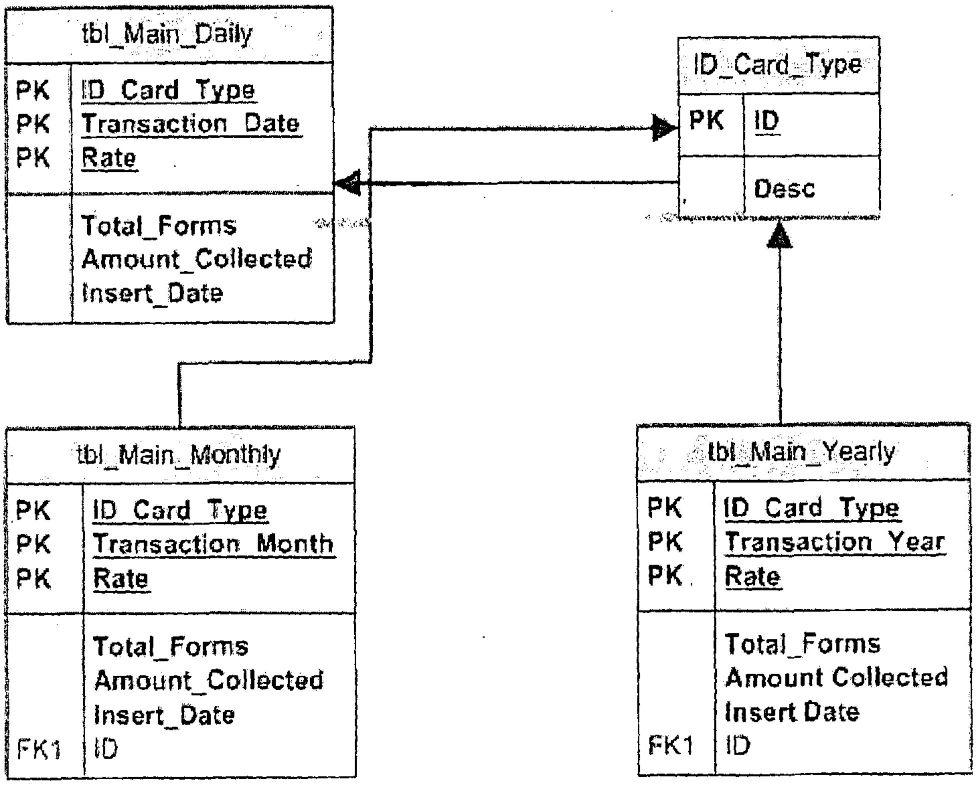
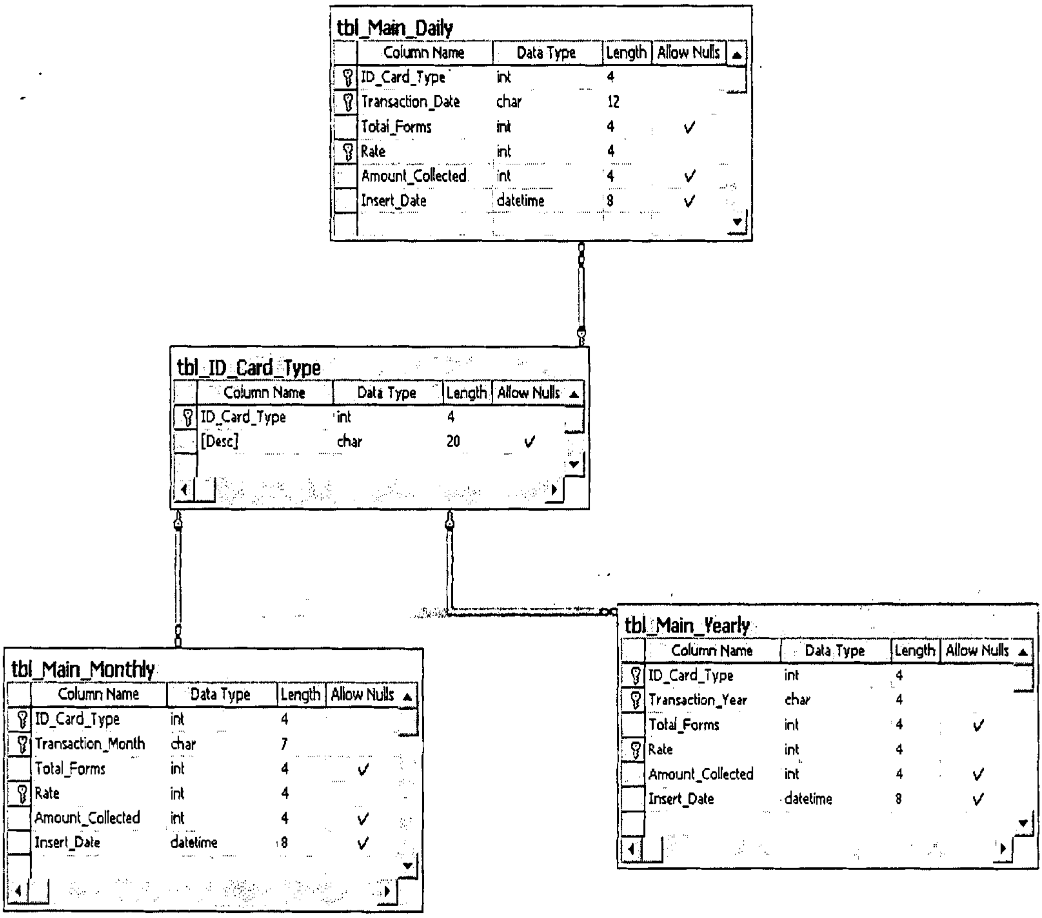


Fig 4.8 Proposed Enhanced Schema

NADRA is using SQL Server as DBMS so the schema of enhanced schema is shown in SQL-Server view is given in figure 4.9.



tbl_Main_Daily

	Column Name	Data Type	Length	Allow Nulls
🔑	ID_Card_Type	int	4	
🔑	Transaction_Date	char	12	
	Total_Forms	int	4	✓
🔑	Rate	int	4	
	Amount_Collected	int	4	✓
	Insert_Date	datetime	8	✓

tbl_ID_Card_Type

	Column Name	Data Type	Length	Allow Nulls
🔑	ID_Card_Type	int	4	
	[Desc]	char	20	✓

tbl_Main_Monthly

	Column Name	Data Type	Length	Allow Nulls
🔑	ID_Card_Type	int	4	
🔑	Transaction_Month	char	7	
	Total_Forms	int	4	✓
🔑	Rate	int	4	
	Amount_Collected	int	4	✓
	Insert_Date	datetime	8	✓

tbl_Main_Yearly

	Column Name	Data Type	Length	Allow Nulls
🔑	ID_Card_Type	int	4	
🔑	Transaction_Year	char	4	
	Total_Forms	int	4	✓
🔑	Rate	int	4	
	Amount_Collected	int	4	✓
	Insert_Date	datetime	8	✓

Fig 4-9: SQL-Server view of Enhanced Schema

CHAPTER 5

IMPLEMENTATION

5. Implementation

In this project we implemented the idea in C#.Net. We are supposed to populate the enhanced schema therefore we use triggers as well as developed procedures to populate the new schema designed for data mining.

5.1 Procedures:

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

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

We have written a procedure that loads the available (previous) data for the first time into the enhanced schema relations when enhanced schema relations are created. Following code (query) describes the query that loads data into the relation:

```
----------------------------CNIC Normal
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amou
nt_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=1
and totalamount=75
group by needofcard,convert(char(12),issue_date_time,105),totalamount
```

```
----------------------------CNIC Urgent
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=2
and needofcard=1
and totalamount=180
group by needofcard,convert(char(12),issue_date_time,105),totalamount

----------------------------CRC
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=2
and totalamount=50
group by needofcard,convert(char(12),issue_date_time,105),totalamount
```

```
---------------------------Modification
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=3
and totalamount=140
group by needofcard,convert(char(12),issue_date_time,105),totalamount

---------------------------Duplicate
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=4
and totalamount=140
group by needofcard,convert(char(12),issue_date_time,105),totalamount
```

```
---------------------------Office Mistake
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=6
and totalamount=0
group by needofcard,convert(char(12),issue_date_time,105),totalamount

---------------------------NICOP
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=7
and totalamount=900
group by needofcard,convert(char(12),issue_date_time,105),totalamount
---------------------------
```

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

Following code describes the 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:

```
---------------------------Variable Declarion
declare @mydate1 char(24),@mydate2 char(24)
set @mydate1='2004-12-01 00:00:00.000'
set @mydate2='2004-12-31 23:59:59.999'


---------------------------CNIC Normal
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=1
and totalamount=75
and issue_date_time between @mydate1 and @mydate2
group by needofcard,convert(char(12),issue_date_time,105),totalamount
```

```
----------------------------CNIC Urgent
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amou
nt_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=2
and needofcard=1
and totalamount=180
and issue_date_time between @mydate1 and @mydate2
group by needofcard,convert(char(12),issue_date_time,105),totalamount


----------------------------CRC
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amou
nt_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=2
and totalamount=50
and issue_date_time between @mydate1 and @mydate2
group by needofcard,convert(char(12),issue_date_time,105),totalamount
```

```
----------------------------Modification
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=3
and totalamount=140
and issue_date_time between @mydate1 and @mydate2
group by needofcard,convert(char(12),issue_date_time,105),totalamount

----------------------------Duplicate
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=4
and totalamount=140
and issue_date_time between @mydate1 and @mydate2
group by needofcard,convert(char(12),issue_date_time,105),totalamount
```

```
----------------------------Office Mistake
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=6
and totalamount=0
and issue_date_time between @mydate1 and @mydate2
group by needofcard,convert(char(12),issue_date_time,105),totalamount

----------------------------NICOP
insert into
oltp_analysis..tbl_main_daily(ID_Card_Type,Transaction_Date,Total_Forms,Rate,Amount_Collected)
select needofcard as "ID Card Type",convert(char(12),issue_date_time,105) as
"Transaction Date",count(*) as "Total Forms",totalamount as "Rate",(count(*) *
totalamount) as "Amount Collected"
from applicant
where form_status <> 'W'
and delivery_status=1
and needofcard=7
and totalamount=900
and issue_date_time between @mydate1 and @mydate2
group by needofcard,convert(char(12),issue_date_time,105),totalamount
```

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

Following is the code that describes the queries which calculate the monthly and yearly data from the data stored in the enhanced schema:

```
---------------------------CNIC Normal Monthly
insert into
oltp_analysis..tbl_main_monthly(ID_Card_Type,Transaction_Date,Total_Forms,Rate,A
mount_Collected)
select id_card_type as "ID Card Type",substring(transaction_date,4,7) as "Transaction
Month",sum(total_forms) as "Total Forms",rate as "Rate",sum(amount_collected) as
"Amount Collected"
from tbl_main_daily
where substring(transaction_date,4,2)='12' and substring(transaction_date,7,4)='2004'
group by id_card_type,substring(transaction_date,4,7),rate

---------------------------CNIC Normal Yearly
insert into
oltp_analysis..tbl_main_monthly(ID_Card_Type,Transaction_Date,Total_Forms,Rate,A
mount_Collected)
select id_card_type as "ID Card Type",substring(transaction_month,4,4) as "Transaction
Year",sum(total_forms) as "Total Forms",rate as "Rate",sum(amount_collected) as
"Amount Collected"
from tbl_main_monthly
where substring(transaction_month,4,4)='2004'
group by id_card_type,substring(transaction_month,4,4),rate
```

5.2 Simulation Software

In order to show the results and compare the efficiency and performance simulation software of this project is developed in Microsoft Visual C#.Net. SQL Server 2000 is used to store the Database of the OLTP and enhanced schema. For connectivity we use the SQLCLient.dll library of the .Net frame work. The library provides instant and reliable connectivity with the SQL 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.SqlClient;
```

5.2.2 Timer Component to Display the Current Date and Time

Timer Component is used to display the current date and time value for reference purpose and to compare the query time of the simulator.

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

5.2.3 SQL Server Connectivity

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

```
System.Data.SqlClient.SqlConnection con = new
                              System.Data.SqlClient.SqlConnection ();
con.ConnectionString = "server=SRC-SWD-LAPTOP;uid=sa;pwd=;initial
                        catalog=FTRC_Backup;Connection
                        Timeout=0;integrated security=SSPI;";
con.Open();
```

5.2.4 Calculation of Daily Data Query Time

```
label2.Text  =DateTime.Now.ToLongTimeString ();
label2.Refresh() ;
mysql="select needofcard ,convert(char(12),issue_date_time,105)
      ,count(*) ,totalamount ,(count(*) * totalamount) from applicant
      where form_status <> 'W' and delivery_status=1 and needofcard=1
      and totalamount=75 and issue_date_time between '2004-12-01
      00:00:00.000' and '2004-12-01 23:59:59.999' group by
      needofcard,convert(char(12),issue_date_time,105),totalamount ";
      cmd = new SqlCommand(mysql, con);
      cmd.CommandTimeout =0;
      cmd.ExecuteNonQuery();
label3.Text  =DateTime.Now.ToLongTimeString ();
label3.Refresh();

date1=DateTime.Parse(label2.Text);
date2=DateTime.Parse(label3.Text);
ts=date2-date1;
label6.Text =ts.ToString();
label6.Refresh ();

label4.Text  =DateTime.Now.ToLongTimeString ();
label4.Refresh();
```

```
mysql="select id_card_type,transaction_date,total_forms ,rate
      ,amount_collected  from oltp_analysis..tbl_main_daily where
      transaction_date='01-12-2004' and id_card_type=1 and rate=75";
cmd = new SqlCommand(mysql, con);
cmd.CommandTimeout =0;
cmd.ExecuteNonQuery();
label5.Text  =DateTime.Now.ToLongTimeString ();
label5.Refresh();

date1=DateTime.Parse(label4.Text);
date2=DateTime.Parse(label5.Text);
ts=date2-date1;
label7.Text =ts.ToString();
label7.Refresh ();
```

5.2.5 Calculation of Monthly Data Query Time

```
label8.Text  =DateTime.Now.ToLongTimeString ();
label8.Refresh() ;
mysql="select
      needofcard,substring(convert(char(12),issue_date_time,105),4,7)
      ,count(*) ,totalamount ,(count(*) * totalamount) from applicant
      where form_status <> 'W' and delivery_status=1 and needofcard=1
      and totalamount=75 and issue_date_time between '2004-12-01
      00:00:00.000' and '2004-12-31 23:59:59.999' group by
      needofcard,substring(convert(char(12),issue_date_time,105),4,7),
      totalamount";
cmd = new SqlCommand(mysql, con);
cmd.CommandTimeout =0;
cmd.ExecuteNonQuery();
label9.Text  =DateTime.Now.ToLongTimeString ();
label9.Refresh();

date1=DateTime.Parse(label8.Text);
date2=DateTime.Parse(label9.Text);
ts=date2-date1;
label12.Text =ts.ToString();
```

```
label12.Refresh ();

label10.Text  =DateTime.Now.ToLongTimeString ();
label10.Refresh();
mysql="select id_card_type ,transaction_month ,total_forms ,rate
      ,amount_collected from oltp_analysis..tbl_main_monthly where
      transaction_month='12-2004' and id_card_type=1 and rate=75";
cmd = new SqlCommand(mysql, con);
cmd.CommandTimeout =0;
cmd.ExecuteNonQuery();
label11.Text  =DateTime.Now.ToLongTimeString ();
label11.Refresh();

date1=DateTime.Parse(label10.Text);
date2=DateTime.Parse(label11.Text);
ts=date2-date1;
label13.Text =ts.ToString();
label13.Refresh ();
```

5.2.6 Calculation of Yearly Data Query Time

```
label14.Text  =DateTime.Now.ToLongTimeString ();
label14.Refresh() ;
mysql="select needofcard
      ,substring(convert(char(12),issue_date_time,105),7,4) ,count(*)
      ,totalamount ,(count(*) * totalamount) from applicant where
      form_status <> 'W' and delivery_status=1 and needofcard=1 and
      totalamount=75 and issue_date_time between '2004-01-01
      00:00:00.000' and '2004-12-31 23:59:59.999' group by
      needofcard,substring(convert(char(12),issue_date_time,105),7,4),
      totalamount";
cmd = new SqlCommand(mysql, con);
cmd.CommandTimeout =0;
cmd.ExecuteNonQuery();
```

```
label15.Text  =DateTime.Now.ToLongTimeString ();
label15.Refresh();

date1=DateTime.Parse(label14.Text);
date2=DateTime.Parse(label15.Text);
ts=date2-date1;
label18.Text =ts.ToString();
label18.Refresh ();



label16.Text  =DateTime.Now.ToLongTimeString ();
label16.Refresh();
mysql="select id_card_type ,transaction_year ,total_forms,rate
      ,amount_collected from oltp_analysis..tbl_main_yearly where
      transaction_year='2004' and id_card_type=1 and rate=75";
cmd = new SqlCommand(mysql, con);
cmd.CommandTimeout =0;
cmd.ExecuteNonQuery();
label17.Text  =DateTime.Now.ToLongTimeString ();
label17.Refresh();


date1=DateTime.Parse(label16.Text);
date2=DateTime.Parse(label17.Text);
ts=date2-date1;
label19.Text =ts.ToString();
label19.Refresh ();
```

5.2.7 Exception Handling

Exception Handling is applied on the code for abnormal termination. It provides the actual error with detail description whenever the code returns and error during execution.

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

      catch (Exception ee)

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

5.2.8 Exiting from Application

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

CHAPTER 6

RESULTS

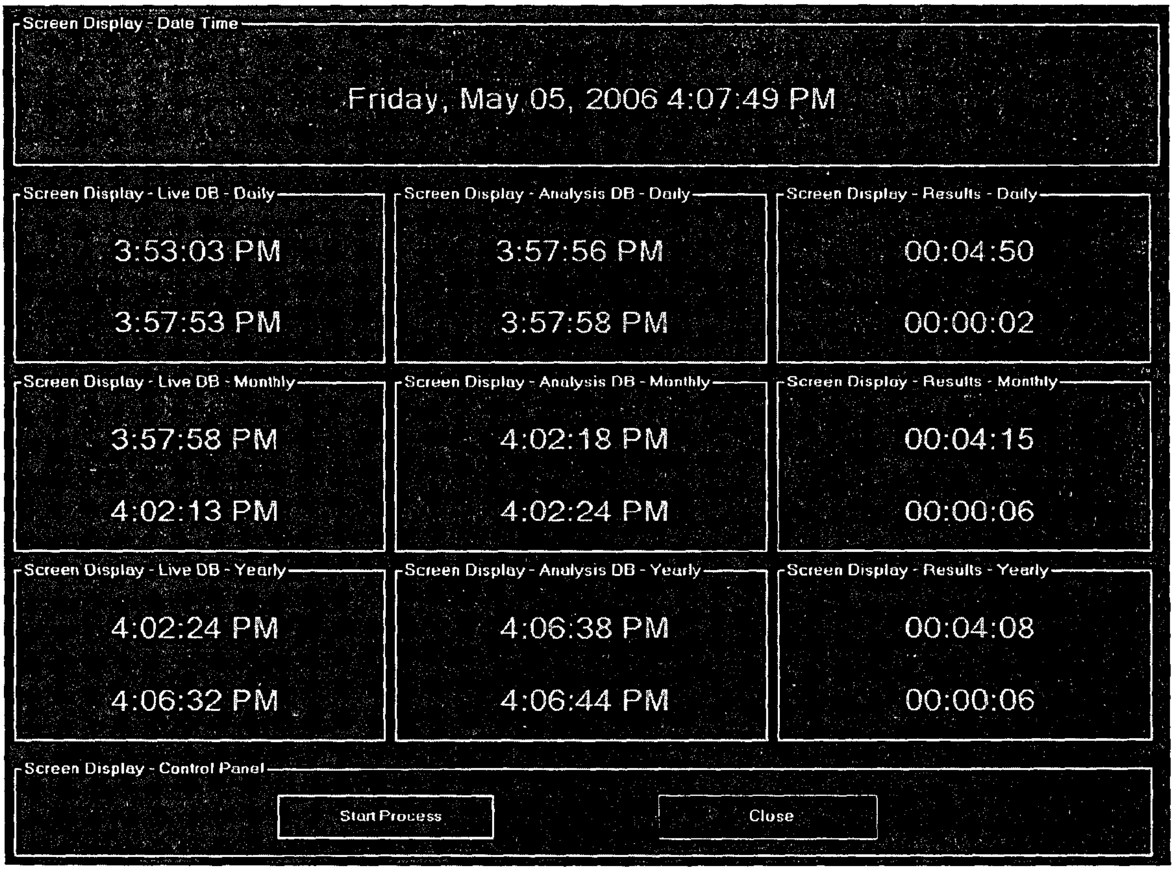
6. Results

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 and used the simulation software to calculate the time for each query and also its comparison.

The results the displayed on the screen and calculated comparison are also displayed.

6.1 Result Screen

After Start Process is clicked the process will start running and after completion of the process the following Fig 6.1 screen will be displayed.



Screen Display - Date Time

Friday, May 05, 2006 4:07:49 PM

Screen Display - Live DB - Daily

3:53:03 PM

3:57:53 PM

Screen Display - Analysis DB - Daily

3:57:56 PM

3:57:58 PM

Screen Display - Results - Daily

00:04:50

00:00:02

Screen Display - Live DB - Monthly

3:57:58 PM

4:02:13 PM

Screen Display - Analysis DB - Monthly

4:02:18 PM

4:02:24 PM

Screen Display - Results - Monthly

00:04:15

00:00:06

Screen Display - Live DB - Yearly

4:02:24 PM

4:06:32 PM

Screen Display - Analysis DB - Yearly

4:06:38 PM

4:06:44 PM

Screen Display - Results - Yearly

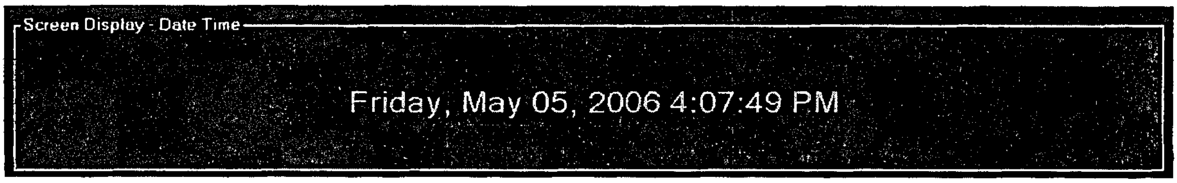
00:04:08

00:00:06

Screen Display - Control Panel

Fig 6-1 Result Screen

Fig 6-2 shows the current date time for reference purpose.

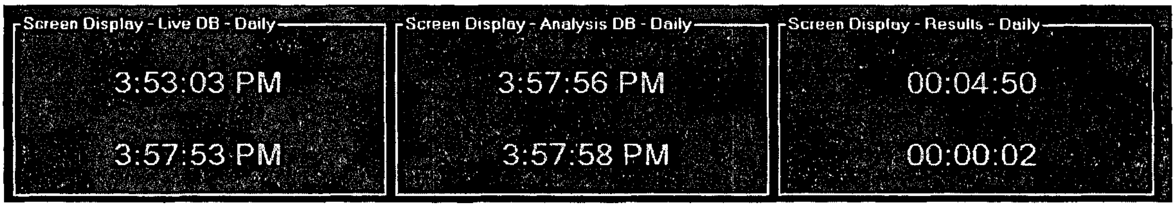


Screen Display - Date Time

Friday, May 05, 2006 4:07:49 PM

Fig 6-2 Date and Time Panel

Fig 6-3 shows times for Daily data query.



Screen Display - Live DB - Daily

3:53:03 PM

3:57:53 PM

Screen Display - Analysis DB - Daily

3:57:56 PM

3:57:58 PM

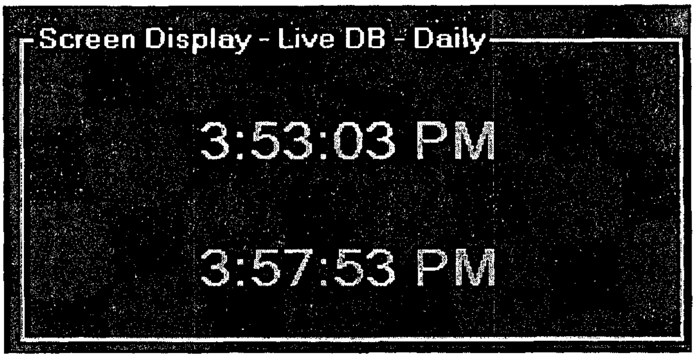
Screen Display - Results - Daily

00:04:50

00:00:02

Fig 6-3 Time Panel for Daily data Query

Fig 6-3-a shows the starting time of query for daily data.



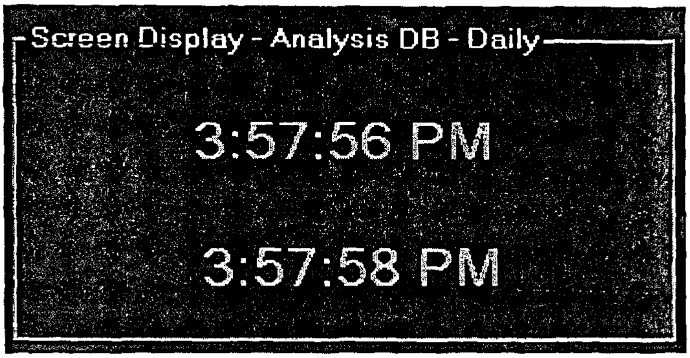
Screen Display - Live DB - Daily

3:53:03 PM

3:57:53 PM

Fig 6-3-a Starting Time Panel

Fig 6-3-b shows the ending time of query.



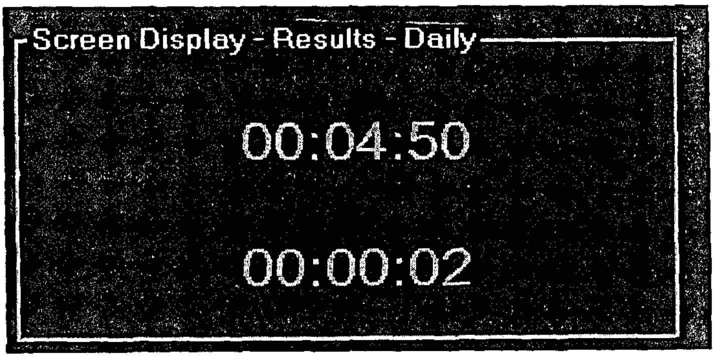
Screen Display - Analysis DB - Daily

3:57:56 PM

3:57:58 PM

Fig 6-3-b Ending Time Panel

Fig 6-3-c shows the difference of time for query in OLTP and enhance schema



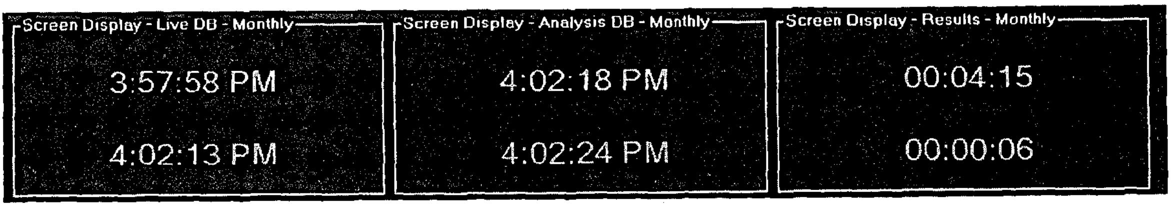
Screen Display - Results - Daily

00:04:50

00:00:02

Fig 6-3-c Time Difference Panel

Fig 6-4 shows the times for monthly data query



Screen Display - Live DB - Monthly

3:57:58 PM

4:02:13 PM

Screen Display - Analysis DB - Monthly

4:02:18 PM

4:02:24 PM

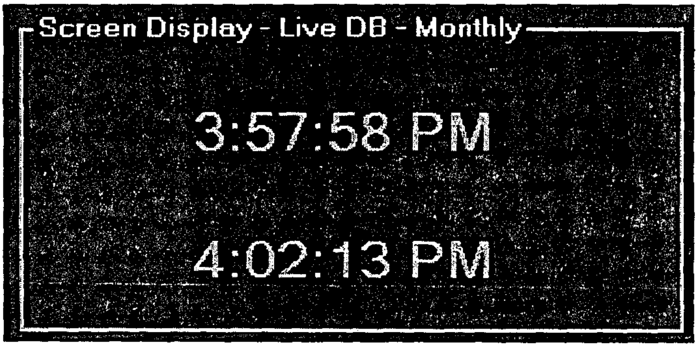
Screen Display - Results - Monthly

00:04:15

00:00:06

Fig 6-4 Monthly Data Query Time Panel

Fig 6-4-a shows the starting time of a query for monthly data.



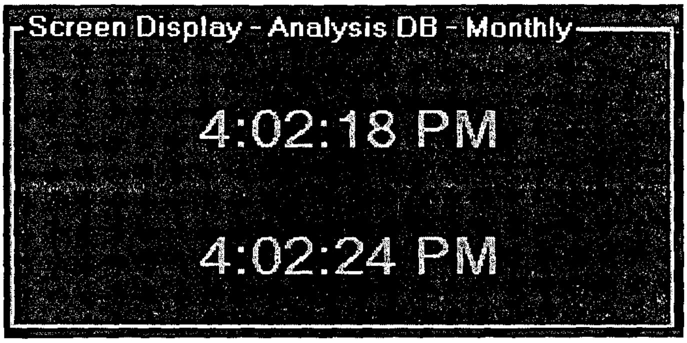
Screen Display - Live DB - Monthly

3:57:58 PM

4:02:13 PM

Fig 6-4-a Starting Time for Monthly Data Query Panel

Fig 6-4-b shows the starting time of a query for monthly data.



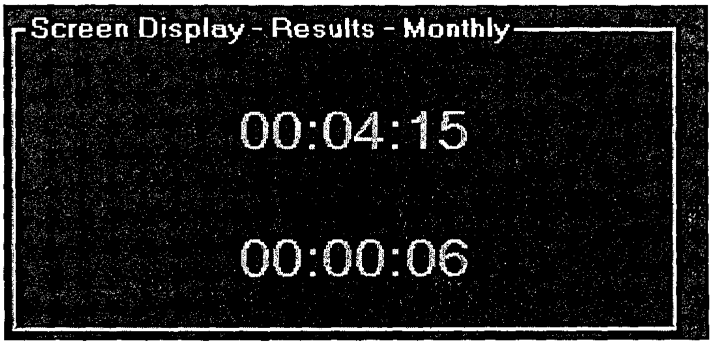
Screen Display - Analysis DB - Monthly

4:02:18 PM

4:02:24 PM

Fig 6-4-b Ending time for Monthly Data Query Panel

Fig 6-4-c shows starting time of query for monthly data



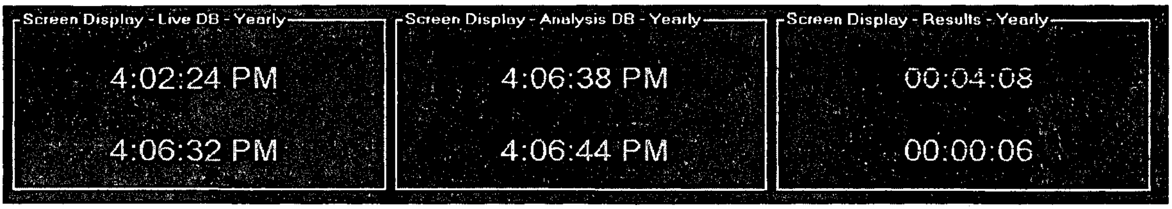
Screen Display - Results - Monthly

00:04:15

00:00:06

Fig 6-4-c Result Panel

Fig 6-5 shows the times for Yearly Data Query.



Screen Display - Live DB - Yearly

4:02:24 PM

4:06:32 PM

Screen Display - Analysis DB - Yearly

4:06:38 PM

4:06:44 PM

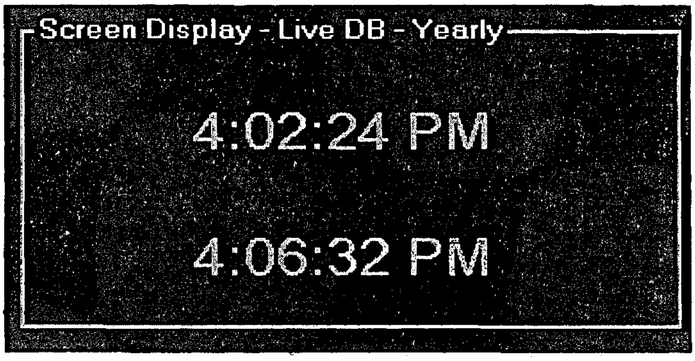
Screen Display - Results - Yearly

00:04:08

00:00:06

Fig 6-5 Time Panel for Yearly Data

Fig 6-5-a shows the starting time of yearly query.



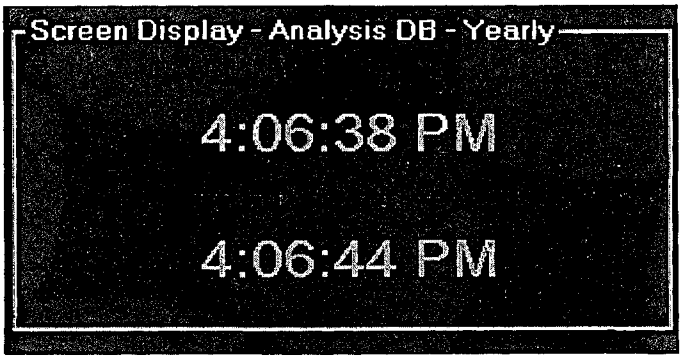
Screen Display - Live DB - Yearly

4:02:24 PM

4:06:32 PM

Fig 6-5-a Query Starting Time for yearly data

Fig 6-5-b shows the Ending time of query.



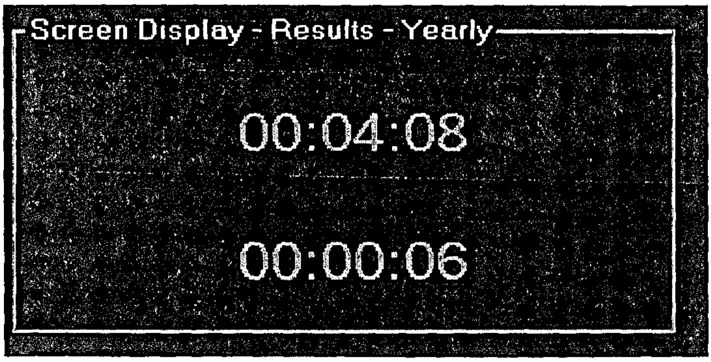
Screen Display - Analysis DB - Yearly

4:06:38 PM

4:06:44 PM

Fig 6-5-b Ending Time Panel for Yearly Data

Fig 6-5-c shows the difference of time of queries for yearly data.



Screen Display - Results - Yearly

00:04:08

00:00:06

Fig 6-5-c Time Comparison Panel for Yearly data

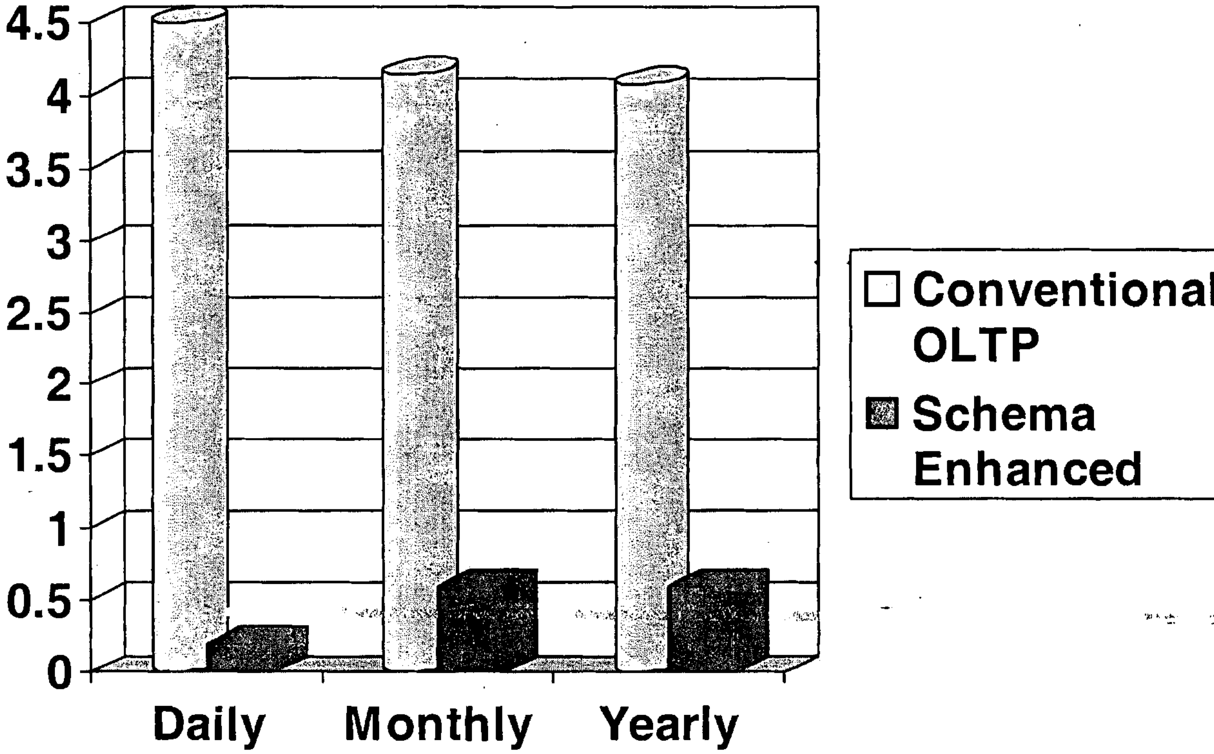
Fig 6-6 shows the Start Process and Close option buttons.

Screen Display - Control Panel
Start Process
Close

Fig 6-6 Start Process and Close Button Panel.

6.2 Comparison Graph

The graph shown in Graph 6-1 shows the comparison of test run.



Graph 6-1 Comparison Graph

6.3 Sample Size for Test

Record Set	Live Database	19968
	Backup Database	4043648
	Schema Enhancement DB	4319

Table 6-1 Sample Size for Test

6.4 Comparison Sheet on Conventional OLTP

System Name	Daily [1]	Monthly [2]	Yearly [3]
	Full Scan	Full Scan	Full Scan
1.6 GHz Centrino Mobile	6 min 30 sec	5 min 3 sec	4 min 42 sec
800 MHz Intel Original	2 min 24 sec	2 min 12 sec	2 min 17 sec
Xeon 3.20 GHz	37 sec	21 sec	22 sec
1.70 GHz Intel Original	3 min 55 sec	4 min 26 sec	4 min 40 sec

Table 6-2 Comparison Sheet on Different Machines for Conventional OLTP

6.5 Comparison Sheet on Schema Enhancement

System Name	Daily [11]		Monthly [12]		Yearly [13]	
	One Time [111]	Table Scan [112]	One Time [113]	Table Scan [114]	One Time [115]	Table Scan [116]
1.6 GHz Centrino Mobile	1 min 15 sec	4 sec	3 sec	5 sec	3 sec	4 sec
800 MHz Intel Original	18 sec	less than 1 sec	2 sec	less than 1 sec	2 sec	1 sec
Xeon 3.20 GHz	5 sec	less than 1 sec	2 sec	less than 1 sec	2 sec	less than 1 sec
1.70 GHz Intel Original	16 sec	3 sec	3 sec	3 sec	3 sec	3 sec

Table 6-3 Comparison Sheet on Different Machines for Schema Enhancement

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7. Conclusion and Future Enhancements

The simulation software is run on the Pentium Machine with limited memory and slow processing speed. Better results can be obtained if the server machine is used with Xeon processor or Dual processing power system. However as compared to the OLTP system the Schema Enhancement Method shows a very good result for analysis purpose. The results come in seconds rather than minutes.

7.1 Conclusion

By applying the Schema enhancement method we have achieved our target i.e., OLTP system with the support of Mining schema and our system performs much better then the Conventional system. The Simulation Process also shows the same results and proves our research. The Performance of the system increases and query time is very less as compared to the conventional OLTP.

We have also checked the performance of our model on different machines and results are shown in chapter 6 in the form of graph / table. After that we have produced the research paper titled "Performance efficient mining on an OLTP System using Schema Enhancement Method" refer to Publication which has been accepted for publication in Euro Journal.

7.2 Future Enhancements

The research area is still open because we have only discussed the performance aspect of the Schema Method. There are some other areas that can be addressed like Storage efficiency and Cost factors of the Schema Method.

Storage Factor and Cost factor of the Schema Enhancement Method can also make a lot of research area for the new researchers.

REFERENCES AND BIBLIOGRAPHY

References and Bibliography

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

2. Publications

[1]. 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)

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

[3]. Jiawei Han, Yongjian Fu, Yue Huang, Yandong Cai and Nick Cercone.
"DBLearn: A System Prototype for Knowledge Discovery in Relational Databases." , ACM SIGMOD Record, Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data SIGMOD '94, Volume 23 issue 2.

[4]. Jiawei Han, Jenny Y. Chiang, Sonny Chee, Jianping Chen and Qing Chen.
"DBMiner: A System for Data Mining in Relational Databases and Data Warehouses", ACM Proceedings of he 1997 conference of the center for advanced studies on collaborative research, November 1997.

A. User Manual:

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

A-1 Main Screen

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

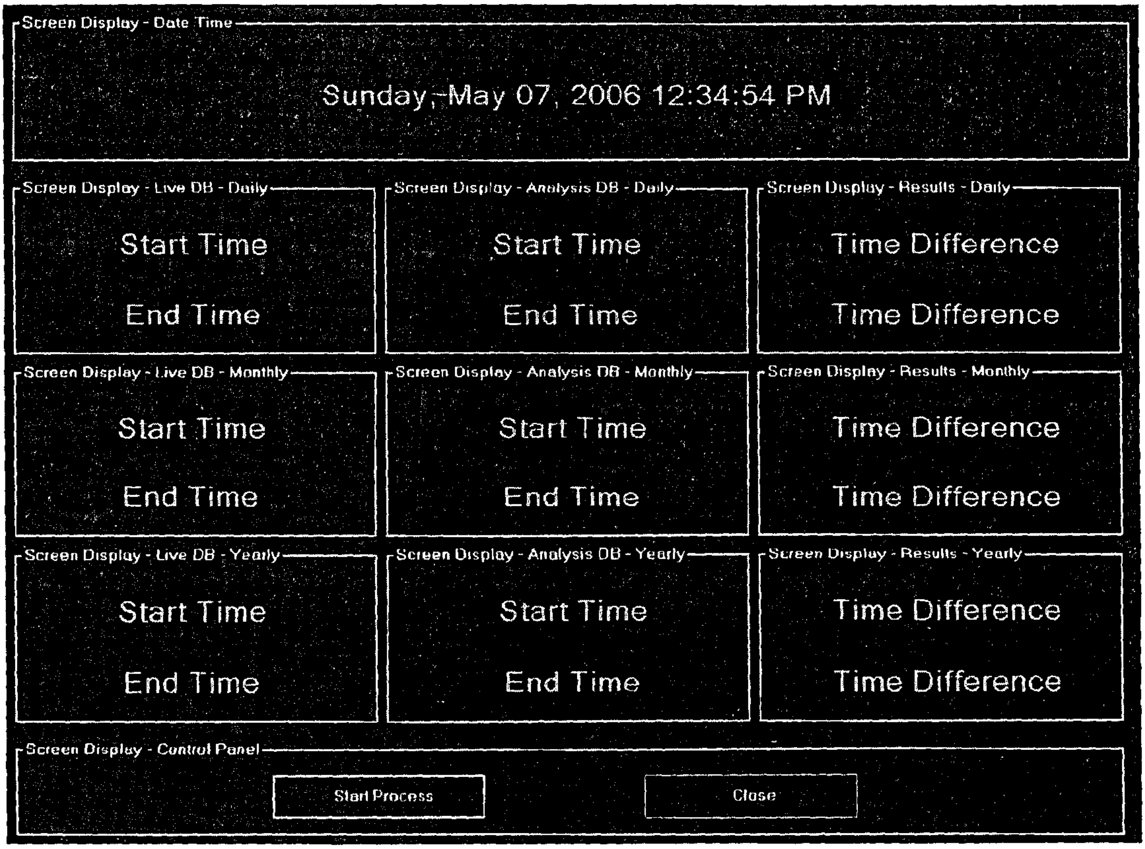

Screen Display - Date Time
Sunday, May 07, 2006 12:34:54 PM
Screen Display - Live DB - Daily
Start Time
End Time
Screen Display - Analysis DB - Daily
Start Time
End Time
Screen Display - Results - Daily
Time Difference
Time Difference
Screen Display - Live DB - Monthly
Start Time
End Time
Screen Display - Analysis DB - Monthly
Start Time
End Time
Screen Display - Results - Monthly
Time Difference
Time Difference
Screen Display - Live DB - Yearly
Start Time
End Time
Screen Display - Analysis DB - Yearly
Start Time
End Time
Screen Display - Results - Yearly
Time Difference
Time Difference
Screen Display - Control Panel
Start Process
Close

Fig A.1 Main Screen of Simulation Software

APPENDIX A

USER MANUAL

In Fig A.2 Current Date and Time is displayed for calculating the comparison results.

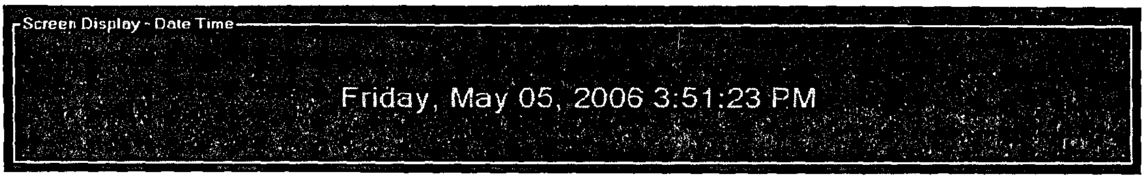


Fig A.2 Date and Time of Query Panel

In Fig A.3 the Labels will display the result by querying the data of one day from OLTP and Enhanced Schema (Analysis) Databases.

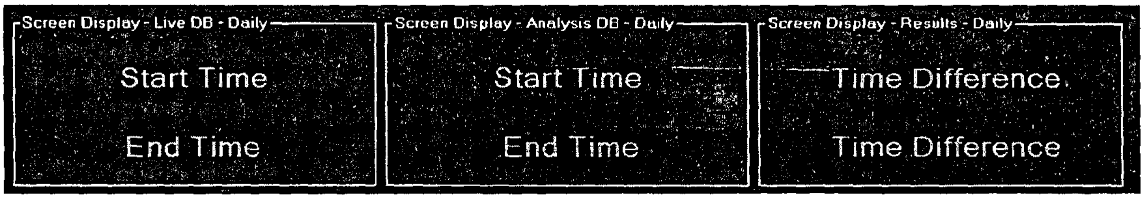

Screen Display - Live DB - Daily
Start Time
End Time
Screen Display - Analysis DB - Daily
Start Time
End Time
Screen Display - Results - Daily
Time Difference
Time Difference

Fig A.3 Comparison Results for daily data Panel

In Fig A.4 Labels will display the result by querying the data of one month from OLTP and enhanced schema (Analysis) Databases.

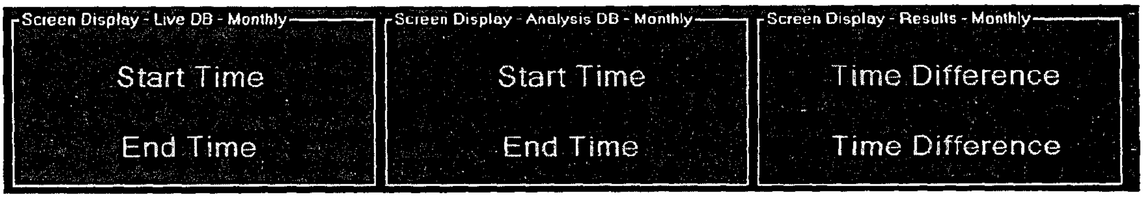

Screen Display - Live DB - Monthly
Start Time
End Time
Screen Display - Analysis DB - Monthly
Start Time
End Time
Screen Display - Results - Monthly
Time Difference
Time Difference

Fig A.4 Comparison Results for monthly data Panel

In Fig A.5 Labels will display the result by querying the data of one year from OLTP and enhanced schema (Analysis) Databases.

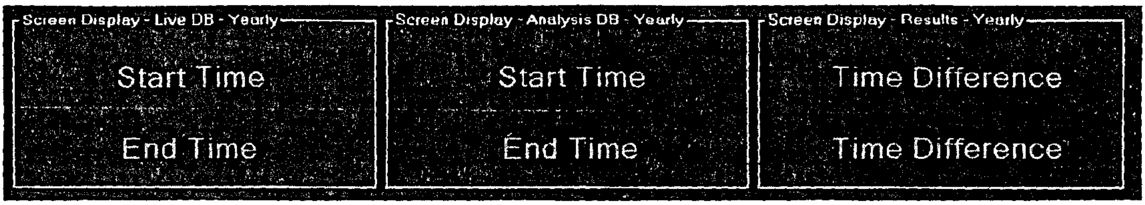

Screen Display - Live DB - Yearly
Start Time
End Time
Screen Display - Analysis DB - Yearly
Start Time
End Time
Screen Display - Results - Yearly
Time Difference
Time Difference

Fig A.5 Comparison Result for daily data Panel

Start Process will initiate the execution of process while close will exit the application as shown in Fig A.6.

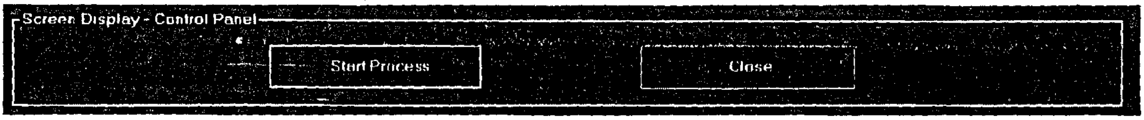


Fig A.6 Start and Close Panel

B. Data Models

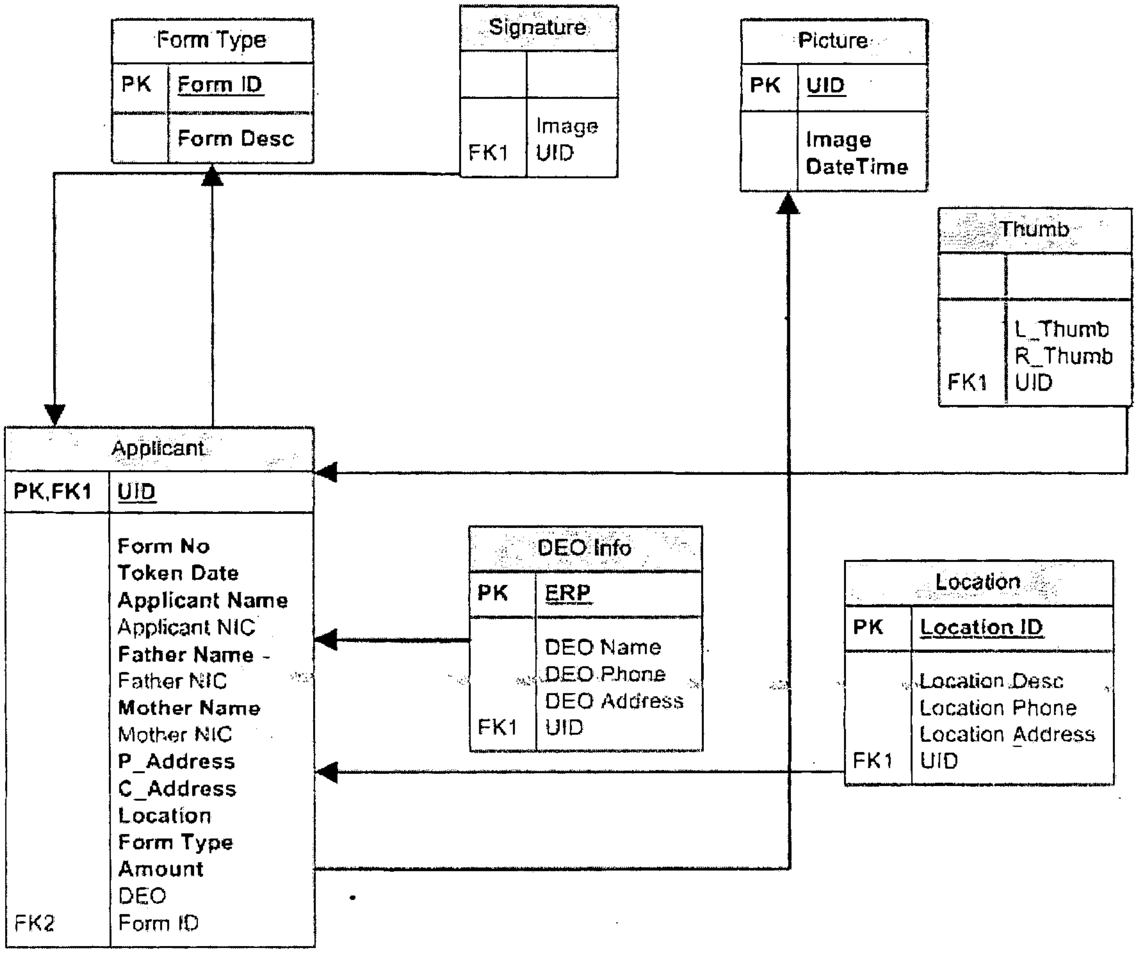


Fig B-1 ERD of Operational System

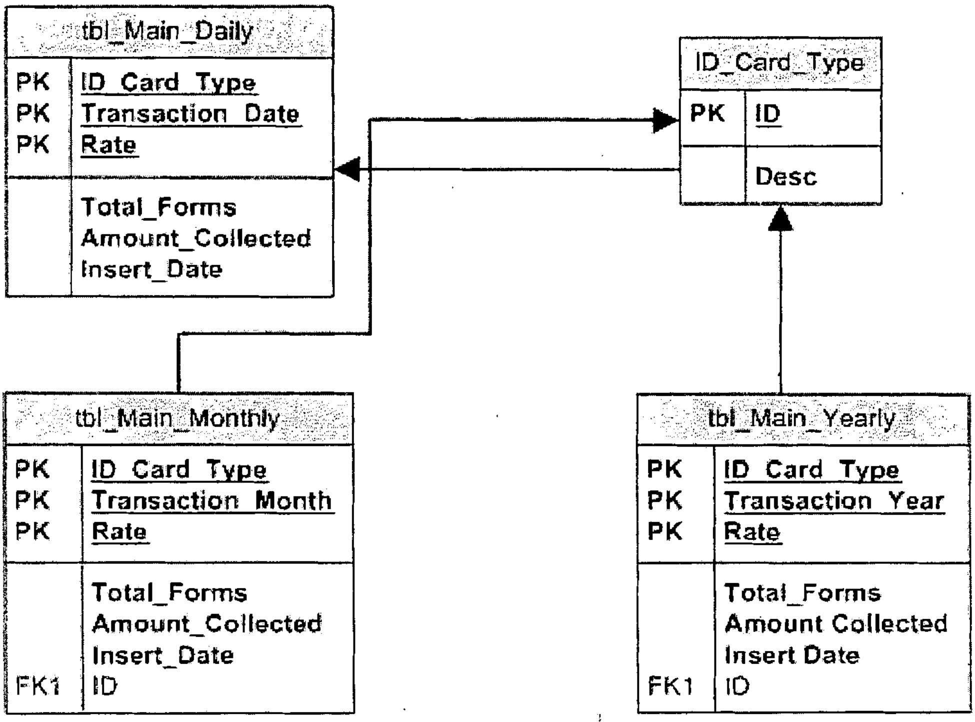


Fig B-2 ERD of Enhanced Schema of Mining Model

PUBLICATION

Performance Efficient Mining on an OLTP System using Schema Enhancement Method

Mr. Muhammad Hasan Rasheed
Department of Computer Science
International Islamic University Islamabad, Pakistan

Mr. Muhammad Imran Saeed
Department of Computer Science
International Islamic University Islamabad, Pakistan

Dr. M. Sikandar Hayat Khiyal
Department of Computer Science
International Islamic University Islamabad, Pakistan

Abstract

Mining on an OLTP System is possible but the cost of running Mining Module on a Conventional OLTP Architecture is not practical. The response time decreases and the Normal Functioning of the System disturb.

Schema Enhancement Method is proposed to overcome these disadvantages in OLTP System. After Modifying Conventional OLTP to Schema Enhanced OLTP Structure, the mining module runs very fast as compare to the Conventional OLTP.

The idea is proposed because installation of DSS Solution is not possible for all the users. The cost and resource factors are very large in DSS Solution. Schema Enhancement Method take advantage over the Conventional OLTP is that it supports the DSS (to some extent) and the cost of the System is also low. This solution is the combination on OLTP (for running Real Time business) and DSS (for running Analysis/Mining module) together.

Performance Efficient System takes advantages of storing the Summary Data inside the actual Schema of the OLTP and the Query Time for Mining the Data decrease.

Keywords

Online Transaction Processing, Online Analytical Processing, Decision Support System, National database and Registration Authority, NADRA Swift Registration Center, Automated Finger Identification System, Microsoft SQL Server 2000 Professional Edition.

1. Introduction

The demand of OLTP environment is increasing day by day in the developing countries like Pakistan. The Analysis Module is an important part of any Software Architecture that tells the owner about the growth of the business and the future demands. The basic difference between OLTP architecture and OLAP is that the OLTP works in transaction bases processing that supports very limited amount of analysis. Although Analysis is possible but it is very time consuming and resource hungry and it is not possible during the real time working of the Software. OLAP is actually designed to meet the demands and needs of the Decision making or DSS (Decision Support Systems). However the OLAP architecture is very expensive and takes a lot of Human Resource and Hardware to install for the organization. Similarly to run the OLAP System Technical persons are required like Database Administrators and System Administrators. This much cost is not possible for a Small level business that is running on the total Human resource of 5-6 people and have limited resources of Hardware and not very Technical persons therefore there is a need for a System that is running on an OLTP architecture and supports Mining but the cost of the system is not exceeding the boundaries of the small level company or shop.

The idea of Mining on an OLTP system has been in research for many years. Some scholars have supported the idea but the implementation has not been done by them and it was made open to design such a system that should be an OLTP but have some support of Mining. There is of course not possible to have a intelligent Mining that is present in a OLAP or DSS or Warehouse Solutions however the idea is to make such a system or to modify the existing system that will make minimum changes in the architecture of the OLTP system.

In year 2000, Erik Riedel, Christos Faloutsos, Gregory R. Ganger and David F. Nagle publish a research paper "**Data mining on an OLTP system (nearly) for free**" presents a new idea of making the system intelligent and strong that helps mining on a OLTP system. The word "**nearly for free**" is an important part of the title which means that the 100% results will not be returned from this idea however by scheduling the disk requests that takes advantage of the ability of high-level functions to operate directly at individual disk drives and thus the Mining data that will be required in future by the Software will be fetch from the disk head before it is required.

2. Schema Enhancement Method

We present the idea of Schema Enhancement for OLTP Systems. The basic OLTP architecture will remain the same and software will work fine with this new enhancement. This modification is only required by those Systems willing to include the Mining Module in the Systems.

Figure 2-A shows the Conventional OLTP System. A Database Server and a client machine that access the Database Server. OLTP (Online Transaction Processing) runs on real time environment and Server machine accepts the requests from the clients and process the queries. If we run complex queries the system down time will increase and response time will decrease resulting in bad image on the company.

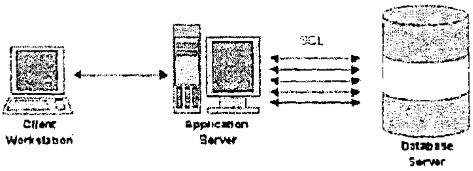


Figure 2-A

In figure 2-B OLAP or DSS system architecture is shown.

This system is designed to run complex queries. High level server machines and Technical staff are present to handle high level queries. Here the response is very fast but buying such a system is not affordable and similarly running cost of these systems is very high.

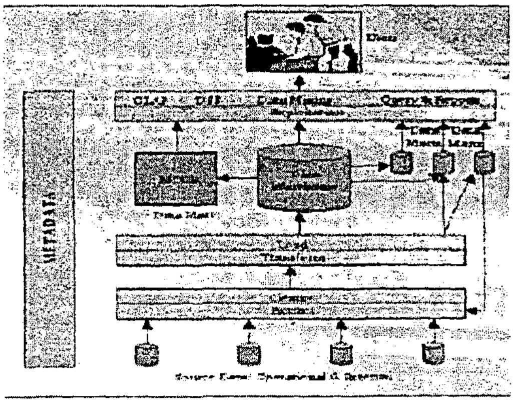


Figure 2-B

In figure 2-C Schema Enhancement Method is shown.

The system is an OLTP based system. During the conversion process, which will run at the time of System Design or any other time of the System, a new Mining supported Schema will be included in the system main architecture. This Schema will be the main source of Mining and the Mining data will come from the actual Database.

The data for Mining Module will come either using the DBMS build in trigger process or user developed software that will feed the Mining Module according to the requirements of the Company. Incase of new requirements the same procedure will be repeated once for the old data and regular for the new data.

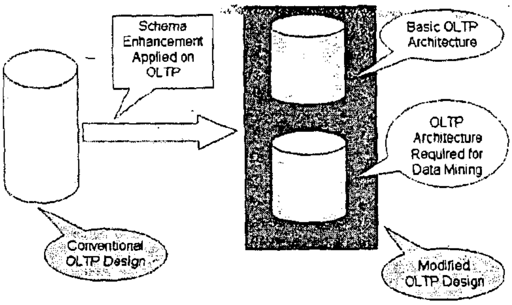

Schema Enhancement Applied on OLTP
Basic OLTP Architecture
OLTP Architecture Required for Data Mining
Conventional OLTP Design
Modified OLTP Design

Figure 2-C

The normal Relational DBMS process will apply on this system:

- Analysis
- Requirement Gathering
- Design of ERD
- Normalization Process
- System Design
- Database Structure
- Implementation and Testing

However in Requirement Gathering and System Design, new phases will be included for supporting Mining Module. The new process will be:

- Analysis
- **Requirement Gathering of Mining**
- Design of ERD
- Normalization Process
- System Design
- **System Design for Mining Module**
- Database Structure
- Implementation and Testing

3. Case Study (NADRA Data Entry Project)

National Database and Registration Authority (NADRA) a Government Registration Organization working under the Ministry of Interior, Government of Pakistan with its Headquarters in Islamabad, Pakistan. This IT Organization is responsible for the registrations of Pakistani Citizens living in Pakistan and Abroad. It has a Dataware House of operating capacity of 20 Tera bytes of Space with latest NCR Machines. The Dataware house is designed to work in extreme environments, running heavy queries over the Data of approx. 5 crore citizens at present.

NADRA Swift Registration Center (NSRC) is the registration office working in every Tehsil of Pakistan with total of approx 300 Centers in the country. The basic concept is to process the applicant record from the NSRCs, collect the data from the whole country, Consolidate and Load in the Central Dataware house after applying business rules to the Data including Facial and AFIS (Automated Finger Identification System) checks.

The working environment of the NSRC is SQL Server 2K with Windows 2K Server and a 2-Tier Software Application working in a Local Network between Clients (more than one) and Server (only one). There is a huge amount of Data present in the NSRCs after 2-3 working years making it difficult to run even small queries at Local level. The server response time decreases and the normal working of the whole system stops.

Schema Enhancement is applied on the existing System without making any changes in the Registration Software.

Sample Size:

Record Set	Live Database	19968
	Backup Database	4043648
	Schema Enhancement DB	4319

Table 3-A

Conventional OLTP:

System Name	Daily [1]	Monthly [2]	Yearly [3]
	Full Scan	Full Scan	Full Scan
1.6 GHz Centrino Mobile	6 min 30 sec	5 min 3 sec	4 min 42 sec
800 MHz Intel Original	2 min 24 sec	2 min 12 sec	2 min 17 sec
Xeon 3.20 GHz	37 sec	21 sec	22 sec
1.70 GHz Intel Original	3 min 55 sec	4 min 26 sec	4 min 40 sec

Table 3-B

Schema Enhancement Method:

System Name	Daily [11]		Monthly [12]		Yearly [13]	
	One Time [111]	Table Scan [112]	One Time [113]	Table Scan [114]	One Time [115]	Table Scan [116]
1.6 GHz Centrino Mobile	1 min 15 sec	4 sec	3 sec	5 sec	3 sec	4 sec
800 MHz Intel Original	18 sec	less than 1 sec	2 sec	less than 1 sec	2 sec	1 sec
Xeon 3.20 GHz	5 sec	less than 1 sec	2 sec	less than 1 sec	2 sec	less than 1 sec
1.70 GHz Intel Original	16 sec	3 sec	3 sec	3 sec	3 sec	3 sec

Table 3-C

Note:

All the values are taken in real time environment (Working Environment)

[1]: Time taken on conventional OLTP to get Daily Statistics
[2]: Time taken on conventional OLTP to get Monthly Statistics
[3]: Time taken on conventional OLTP to get Yearly Statistics
[11]: Time taken on Schema Enhancement Method to get Daily Statistics
[12]: Time taken on Schema Enhancement Method to get Monthly Statistics
[13]: Time taken on Schema Enhancement Method to get Yearly Statistics
[111]: Time taken to Shift Daily Data to OLTP Analysis DB
[112]: Time taken to Query Daily Data from OLTP Analysis DB
[113]: Time taken to shift Monthly Data from OLTP Analysis Daily Table to OLTP Analysis Monthly Table
[114]: Time taken to query Monthly Data from OLTP Analysis DB
[115]: Time taken to shift Yearly data from OLTP Analysis Monthly Table to OLTP Analysis Yearly Table
[116]: Time taken to query Yearly Data from OLTP Analysis DB

4. Performance Efficient Schema Enhancement Method

Performance Efficient System is the demand of every business of the 21st century. Schema Enhancement method as compared to the conventional OLTP is Performance Effective Solution. The Query Time Decrease and the efficiency of the System increase by this method. As from the table 3-B, it is clear that the Query Time for running Daily Report is approx 6 minutes however in Schema Enhancement Method the Total time is approx 1 min 15 sec (for transferring the Data from Live DB to the Summary Tables) and 4 sec (for Query the Summary Table to get the final results). The Transfer times decrease for the Monthly and Yearly Data because when the Summary Data is present in the Tables then the Monthly and Yearly Summary Data is generated from the Daily Summary Data instead of querying the Full Tables.

5. Conclusions

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.

6. Acknowledgment

We are thankful to National Database and Registration Authority 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.

7. References

[1]. 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)

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

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

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