

Predicted Data Modeling
Using
Data Profiling Techniques

T-4390



Developed By

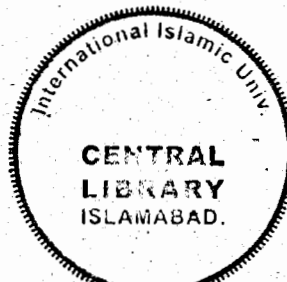
Fahd Bilal ur Rehman
Mata ur Rehman

Supervised By

Dr. Malik Sikandar Hayat Khiyal

Mr. Shahid Rauf

Department of Computer Science
International Islamic University,
Islamabad
(2007)



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

**Department of Computer Science
International Islamic University Islamabad**

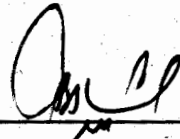
Date: 07-09-2007

Final Approval

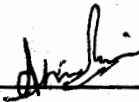
This is to certify that we have read the thesis submitted by **Fahd Bilal ur Rehman** 244-MS (CS)-F05 and **Mata ur Rehman** 246-MS (CS)-F05. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of MS in Computer Science.

Committee:

External Examiner
Dr. Arshad Ali Shahid
Professor,
Department of Computer Science,
National University of Emerging Sciences (FAST)
.H-11, Islamabad.

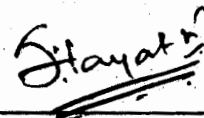


Internal Examiner
Mr. Asim Munir
Assistant Professor,
Department of Computer Science, FAS,
International Islamic University, Islamabad.

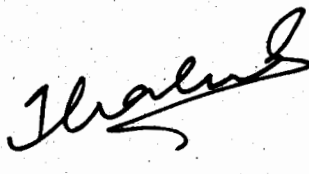


Supervisors

Dr. M. Sikandar Hayat Khiyal
House 1309, Gali 12, Sector I-10/2
Islamabad.



Mr. Shahid Rauf
Head IT Projects & Planning
CM Pak HQ
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)
&
To our Families

Declaration

We hereby declare that this Research "Predicted Data Modeling using Data Profiling" 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 & Mr. Shahid Rauf**. 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.



Fahd Bilal ur Rehman

244-CS/MS/F05



Mata ur Rehman

246-CS/MS/F05

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 and Mr. Shahid Rauf** for their 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.

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



Fahd Bilal ur Rehman

244-CS/MS/F05



Mata ur Rehman

246-CS/MS/F05

Project In Brief

Project Title:	Predicted Data Modeling using Data Profiling Technique
Undertaken By:	Fahd Bilal ur Rehman Mata ur Rehman
Supervised By:	Dr. Malik Sikandar Hayat Khiyal Mr. Shahid Rauf
Start Date:	01 September 2006
Completion Date:	July, 2007
Tools & Technologies	Visual Basic #.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

In data warehousing projects data feed comes from multiple sources and it is in the form of flat files where fields are separated by some delimiters, which does not show the model of an existing system and cause problems in understanding the existing system. Predicted Data modeling using Data profiling is intended to provide a logical data model from the provided business rule defined over a flat file data structure. Data profiling is used to discover the relationships across tables and validate the business rules with the help of statistics calculation.

It is a basic necessity for every data warehouse developer and designer to understand the existing system model. In the past lots of work has been done on data profiling but main focus was to address data quality issues and profiling techniques has not been used to predict data model. This research emphasize that logical data model can be predicted with the help of information and statistic provided by data profiling discovery techniques.

Concerned data warehouse department may have designers who can analyze the data and build the image of the existing system. Effectiveness of that model is based upon the experience of the designer. A trial and error approach is used to understand and construct logical model, which is infect a difficult job, requires higher cost in term of expertise and time.

TABLE OF CONTENTS

Chapter No	Contents	Page No
1.	Introduction	1
1.1	Flat File System	3
1.2	Data Staging	5
1.3	Data Profiling	6
	1.1.1 Outlier Detection	8
	1.1.2 Meta Data Validation	8
	1.1.3 Pattern Analysis	10
	1.1.4 Relationship Discovery	10
	1.1.5 Statistical Analysis	11
	1.1.6 Business-Rule Validation	12
2.	Literature Survey	14
2.1	Purpose of Logical Data Model	14
2.2	Limitations of Data in Flat File System	16
2.3	Capabilities of Data Profiling	20
2.4	Derivation of Logical Data Model	23
3.	Methodology	27
3.1	Requirement Analysis	29
3.2	Problem Definition	29
3.3	Use Case Analysis	30
	3.3.1 Use Case in expanded Format	30
	3.3.2 Use Case Diagram	30
3.4	Design	38
	3.4.1 System Design (Object Oriented Method)	39
	3.4.1.1 State Transition Diagram	40
	3.4.1.2 Sequence Diagram	41
	3.4.1.3 Class Diagram	43
	3.4.2 Data warehouse Architecture	45
3.5	Methodology	46
	3.5.1 Algorithm of Predicted LDM	47
	3.5.2 Mathematical Model of Algorithm	47
4.	Implementation	51
4.1	Simulation Software	51
	4.1.1 Declaration of library	52
	4.1.2 Loading Flat file in a Data set	52

Chapter No	Contents	Page No
	4.1.3 Primary Key Detection	54
	4.1.4 Relationship Discovery	55
	4.1.5 Business Rule Validation	56
	4.1.6 Business Rule Development	58
	4.1.7 Exception Handling	59
	4.1.8 Exiting from Application	59
5.	Results	60
	5.1 Main Screen	61
	5.2 Relationship Discovery Screen	62
	5.3 Show LDM Screen	63
	5.4 Business Rule Creation	64
	5.5 Business Rule Validation	65
	5.6 Manual Reduction	66
	5.7 Manual Addition of Column	67
	5.8 Final LDM	68
6.	Conclusion and Future Enhancements	69
	6.1 Conclusions	70
	6.2 Future Enhancement	70
	References and Bibliography	71

CHAPTER 1
INTRODUCTION

1. Introduction

Data feed of data warehousing or any other analytical system like ODS, ERP, CRM etc. may come from multiple sources. Format of the data from these sources are mostly in the form of flat files system or old non relational database systems. Flat file system usually have non standard and complex data design like it may contain fields that are separated by some delimiters. Relationship among tables in these systems is most likely missing so it may contain redundant and inconsistent entries. Data may also contain non standard and incorrect entries [18]. There is another important point to note that which companies are usually interested in developing data warehouse system? Probably those who have some huge volume of historical data and they want analytic reports from this historical data for their future business decisions and strategies. Usually these companies had got their automated MIS system developed in decade of 1980s. Till now data has been stored in their database systems. Data of these companies is stored in flat file or non relational data base systems. Many companies got replicas of the existing systems, redesigned and converted their database and data in relational database systems, but most of companies still stick with old systems because replication was not feasible and it was very expensive and time consuming.

Now many companies are having their data on database system that are complex in design. But when companies intend to design Data warehouse system from that flat file data,

most challenging issue is vision and understanding of this database design. Even if it is not a flat file system, old databases were not relational, database design of those system are very complex. This is another fact that designer of data warehouse and designer of the actual database are not same. Designer of data warehouse only get data in the form of flat files which doesn't include logical data model. Without logical model it is extremely difficult for Data warehouse designer to get vision and understanding the whole database [3,15]. Constructing logical data model directly from flat file is also extremely difficult and very expensive in term of time and expertise.

Our main challenge is to extract a logical data model from flat file system. Reason is to give abstract and logical view of database system to warehouse designer. This Logical Model of Database or abstraction of database will help data warehouse designer to understand the existing database design, as it is a basic necessity by every data warehouse developer and designer to understand the existing system model.

Luckily we are in the age where many techniques and technologies have been developed in different dimensions. To understand the data, data profiling contain bundle of techniques which help database administrator to get actual status of data and fix any inconsistency and inaccurate entries in database. So far profiling techniques are being used for data management task which mainly focuses on addressing data quality and integration issues [4]. Data profiling contains discovery techniques which give no of statistical information about structure, contents and relation exist in data. These report helps database manager to get clear idea about any inconsistency and anomaly exist in data. Before data is fed in data warehouse, its very important to make sure that data is accurate, otherwise information derived from data will not be accurate. This is a quality issue and a big concern for a company if they want to get correct report and vision about business. If data quality is compromised then correct decision making can be affected. But here our main idea is using profiling techniques to understand and predict the abstraction and logical model of existing flat file based database system. And basic idea is if profiling contains discovery information about data then it will be helpful in understanding the logical model. Some of techniques like relationship discovery are very helpful to construct LDM (Logical Data Model).

Now first we will focus getting understanding about flat file system, its structure and problems with flat file system regarding data. Then we will see what is profiling and what are different profiling techniques. Then we focus on how to get statistics or facts from profiling techniques about existing flat file system of or non relational database system to predict its logical data model.

1.1 Flat File System

A flat file system describes any of various means to encode a data model (most commonly a table) as a plain text file.

A flat file generally records one record per line. Fields may simply have a fixed width with padding, or may be delimited by whitespace, tabs, commas (CSV) or other characters. Extra formatting may be needed to avoid delimiter collision. There are no structural relationships. The data are "flat" as in a sheet of paper, in contrast to more complex models such as a relational database [18].

The classic example of a flat file system is a basic name-and-address list, where the database consists of a small, fixed number of fields: *Name*, *Address*, and *Phone Number*. Another example is a simple HTML table, consisting of rows and columns. This type of database is routinely encountered, although often not recognized as a database.

The following example shows the basic elements of a flat-file database. The data arrangement in flat files consists of a series of columns and rows organized into a tabular format. Like following specific example uses only one table. Here we are using data of an ISP as sample.

The columns include: *Login* (Customers login name); *Full Name* (Full name of the customer); *plan* (service tariff of customer), *PlanAmount* (default amount of plan), *Telephone*.

Here is an example textual representation of the described data:

Login	Full Name	Plan	PlanAmount	Telephone
hafkhan	hafkhan	24 hours access rs 1000	1000	
phma	Mirza Abdul Majid	open @ 10;1000	1973.99	5833868
wahmad	wahmad	24 hours access rs 1000	1000	
mirza66	Ayesha Kamar Waris	monthly expiration rs 300	300	7723754
drars	drars	24 hours access rs 1000	1000	
cdh	Wajid Ur Rehman	unlimited access	1750	6361493
hosshah	hosshah	24 hours access rs 1000	1000	
friend1	friend1	24 hours access rs 1000	1000	
mepco	Shahid Latif	24 hours access rs 2400	2400	7720814-6
jaleel	Jaleel Ur Rehman	summer package rs 15/ hr	300	5811952
simba1	Zeeshan Ahmad	monthly expiration rs 300	300	7841425

Above representation is using tab as field separator. Some flat file systems uses semi colon, colon, coma or other field separator like,

Login;Full Name;Plan;PlanAmount;Active;CurBalance;LastPaymentDate;Telephone

```

hafkhan;hafkhan;24 hours access rs 1000;1000;1;-5.33;3/31/2000 0:00:00;
phma;Mirza Abdul Majid;open @ 10;1000;1;1973.99;4/12/2003 0:00:00;5833868
wahmad;wahmad;24 hours access rs 1000;1000;1;28.33;4/8/2000 0:00:00;
mirza66;Ayesha Kamar Waris;monthly expiration rs 300;300;1;-3.0e-04;3/23/2003 0:00:00;7723754
drars;drars;24 hours access rs 1000;1000;1;0.00;4/12/2000 0:00:00;
cdh;Wajid Ur Rehman;unlimited access;1750;1;-10.35;3/19/2002 0:00:00;6361493
hosshah;hosshah;24 hours access rs 1000;1000;1;-3.00;4/14/2000 0:00:00;
friend1;friend1;24 hours access rs 1000;1000;1;-5.0e-04;4/17/2000 0:00:00;
mepco;Shahid Latif;24 hours access rs 2400;2400;1;2485.77;4/22/2003 0:00:00;7720814-6
jaleel;Jaleel Ur Rehman;summer package rs 15/ hr (open);300;1;121.74;8/23/2002 0:00:00;5811952
simba1;Zeeshan Ahmad;monthly expiration rs 300;300;1;0.00;11/28/2002 0:00:00;7841425
iciboch;Pervaiz Siddique;all day rs 3000;3000;1;1.94;8/21/2002 0:00:00;6363762
irshad92;irshad92;summer package rs 15/ hr (open);300;1;132.83;6/22/2002 0:00:00;
amirm;Amir Mir;summer package rs 10 / hr;100;1;0.00;4/24/2000 0:00:00;7410990

```



```
dawn1;dawn1;monthly expiration rs 300;300;1;0.00;12/19/2002 0:00:00;
rocbustr;Dr Azeem;24 hours access rs 1000;1000;1;8.0e-04;4/27/2000 0:00:00;5165721
suchet;suchet;24 hours access rs 360;360;1;1.4e-03;4/12/2002 0:00:00;
ibraheem;Muhammad Ibrahim;24 hours access rs 3200;3200;0;-1.44;4/29/2000 0:00:00;7831057
qci;Ejaz Yusuf;24 hours access rs 1000;1000;1;4.0e-04;11/3/2001 0:00:00;7841877
```

This type of data representation is quite standard for a flat-file system, although there are some additional considerations that are not readily apparent from the text: [18]

- Data types: each column in a database table such as the one above is ordinarily restricted to a specific data type. Such restrictions are usually established by convention, but not formally indicated unless the data is transferred to a relational database system.
- Separated columns: In the above example, individual columns are separated using whitespace characters. This is also called indentation or "fixed-width" data formatting. Another common convention is to separate columns using one or more delimiter characters. There are *many* different conventions for depicting data such as that above in text. (See e.g., Comma-separated values, Delimiter-separated values, Markup language, Programming language).

1.2 Data Staging

The data staging area is the data warehouse workbench. It is the place where raw data is brought in, cleaned, combined, archived, and eventually exported to one or more data marts. The purpose of the data staging area is to get data ready for loading into a presentation server (a relational DBMS or an OLAP engine). We assume that the data staging area is not a query service. In other words, any database that is used for querying is assumed to be physically downstream from the data staging area.

Perhaps we don't even realize we have a data staging area. Maybe our data just does a "touch and go" landing in between the legacy system and the presentation server. (That is an

airplane metaphor.) We bring the data in briefly, assign a surrogate key, check the records for consistency, and send them on to the DBMS loader that is the presentation database [17].

If the legacy data is already available in a relational database, then it may make sense to perform all the processing steps within the relational framework, especially if the source relational database and the eventual target presentation database are from the same vendor. This makes even more sense when the source database and the target database are on the same physical machine, or when there is a convenient high-speed link between them.

However, there are many variations on this theme, and in many cases it may not make sense to load the source data into a relational database. In the detailed descriptions of the processing steps, we will see that almost all the processing consists of sorting, followed by a single, sequential pass through either one or two tables. This simple processing paradigm does not need the power of a relational DBMS. In fact, in some cases, it may be a serious mistake to divert resources into loading the data into a relational database when what is needed is sequential flat-file processing [16].

Similarly, we will see that if the raw data is not in a normalized entity-relationship (ER) format, in many cases it does not pay to load it into an ER physical model simply to check data relationships. The most important data integrity steps involving the enforcement of one-to-one and one-to-many relationships can be performed, once again, with simple sorting and sequential processing. Keeping these thoughts in mind, lets tease apart as many of the data transformation steps as we can.

1.3. Data Profiling

Data profiling technique are very useful for the accuracy, consistancy and reliability of data. Data profiling contains analytical techniques on data for the purpose of developing a thorough knowledge of its content, structure and quality. Profiling is a process of developing information about data instead of information from data. Incorrect data can give incorrect

analytical report which may result to wrong decision and potential business loss in future. If data is incorrect or non standard, it will not be an asset for the organization.

Companies usually spend billions of dollars for implementing enterprise applications or integrating customer or product data, and industry estimates show these projects fail or go over budget. Beginning a data-driven initiative without first understanding the data is like fixing a car without understanding the problems inside the engine. To fix the engine, we have to understand the depth and breadth of the problem [10].

Similarly, data improvement efforts must start with an understanding of the integrity of the data. The first phase is data profiling also known as data discovery. With data profiling, we can:

- Discover the quality, characteristics and potential problems of information before beginning data-driven projects
- Drastically reduce the time and resources required to find problematic data
- Allow business analysts and data stewards to have more control on the maintenance and management of enterprise data
- Catalog and analyze metadata and discover metadata relationships [6]

Data profiling solutions automatically identify data quality issues in a variety of ways, including:

- Basic statistics, frequencies, ranges and outliers
- Identify multiple spellings of the same content
- Discover and validate data patterns and formats
- Numeric range analysis
- Identify and validate redundant data and primary/foreign key relationships across data sources
- Identify duplicate name and address and non-name and address information
- Validate data specific business rules within a single record or across sources [6]

Data profiling provides an analysis of the data problems we face. Data quality phase starts the process of building better data which contains following steps.

1.3.1 Outlier Detection

Profiling software provides frequency counts and outlier detection techniques that provide automated validation of data. By validating the data that we have - and finding data points that fall well outside of acceptable limits - we can save the immense cost typically spent on manual data validation. Frequency counts also limit the amount of business analyst fault detection required. In essence, these techniques highlight the data values that need further investigation. Outlier detection helps

- Gain insight into data values
- Identify data values that may be considered incorrect
- Drilldown to the data to make a more in-depth determination about the data [5]

For example, a database of customer information might have a number of valid state abbreviations. In many data sources, California is represented as "CA," "CA.," "Ca.," and "California." Non-standard representations complicate any future state-level analysis. Software technology contains rules to recognize these state entries, and the software allows us to consistently identify and contact specified individuals under each of these state abbreviations.[5]

1.3.2 Meta Data Validation

Profiling can scan any sort of data to determine its associated metadata - data that indicates the characteristics present within the data, such as data type, field length, whether the data should be unique, and whether a field can be missing or null.

A complete metadata analysis helps determine if the data matches the expectations of the developer when the data files were created. Has the data migrated from its initial intention over time? Has the purpose, meaning and content of the data been intentionally altered since it was first created? By answering these questions, it helps us make decisions about how to use the data moving forward.

METRIC NAME	METRIC VALUE
Data Type	CHAR
Primary Key Candidate	no
Unique Count	8513
Uniqueness	72.78
Pattern Count	5790
Minimum Value	#101 General Birthday...
Maximum Value	ZOO ANIMAL TUB
Minimum Length	5
Maximum Length	38
Null Count	1
Blank Count	0
Actual Type	string
Count	11698
Data Length	38 chars

Figure 1 [8]

When data and metadata disagree

At times, data and metadata do not match, causing far-reaching implications for our data quality and data integration efforts.

For example, consider a 10 million row field with a field length of 255 characters. If the longest data element in the data is 200 characters, the field length is longer than required, and we are wasting 550MB of disk space. Missing values in a field that should not have missing values can cause joins to fail and reports to yield erroneous results. The figure 1 [8] shows the types of information that a typical metadata report should contain.

1.3.3 Pattern Analysis

Pattern analysis is a technique typically used to determine whether the data values in a field are in the expected format. For example, some fields like a phone number or a product identifier have an expected pattern. Pattern analysis quickly validates that the data in a field is consistent across the data source - and meets our expectations.

PATTERN	COUNT	PERCENTAGE
999-999-9999	3166	96.73
(999)999-9999	42	1.28
(999) 999-9999	34	1.04
999 99 9999 999	20	0.61
999 999 9999	5	0.15
999-999-AAAA	2	0.06
9-999-999-9999	2	0.06
a	1	0.03
99 99 9999 999	1	0.03

Figure 2 [8]

1.3.4 Relationship Discovery

Relationship matching and discovery provide us with information about logistical pieces of data. Organizations maintain an enormous amount of data, such as customer data, supplier data, product data, operational and business intelligence data, financial and compliance data, and industry-specific data. Often, the data for any one of the data categories is spread across many data sources.

"Related" records can be multiple records in the same data file, records across data files or records across databases. With relationship discovery, It help us profile our data to answer the following questions:

- Are there potential key relationships across tables?

- If there is a primary/foreign key relationship, is it enforced?
- If there is an explicit or inferred key relationship, is there any orphaned data (data that does not have a primary key associated with it)?
- Are there duplicate records?

The figure 3 shows the results of a primary key/secondary key analysis, where two products listed in the sales data did not exist in the products table. Understanding relationships across data elements is the first step in consolidating, merging or matching information to get a single, best record of data. In discovering primary/secondary relationship, matching threshold is important. If relationship exist, it will give very high matching threshold like it must be more than 90 %, If relationship doesn't exist and matching threshold is more likely to be less then 5%.

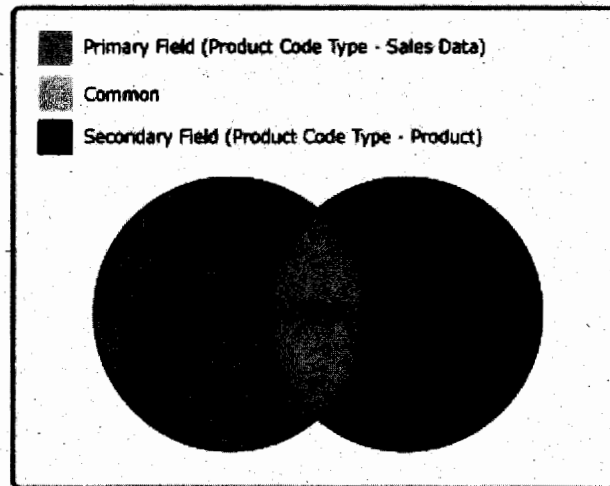


Figure 3 [8]

1.3.5 Statistical Analysis

Data profiling solutions give us a variety of statistical information, including minimum/maximum values, mean, median, mode and standard deviation, to help us assess the validity of our data.

Figure 4 shows statistical data about personal home loan values from a financial organization. Personal home loans normally range from \$20,000 to \$1,000,000. A loan database with incorrect loan amounts can lead to many problems, from poor analysis results to incorrect billing of the loan customer. Let's take a look at some basic statistics from a loan amount column in the loan database. Here it is calculating and giving statistics about the contents of this field, like data type, unique count, uniqueness percentage, maximum and minimum values, null counts, mean, median, mode and standard deviation etc. All these stats are helpful in discovering and removing any inconsistent and inaccurate entries in data.

METRIC NAME	METRIC VALUE
Data Type	double
Primary Key Candidate	no
Unique Count	1140
Uniqueness	70.11
Pattern Count	(not applicable)
Minimum Value	-223000 ←
Maximum Value	9999999 ←
Minimum Length	(not applicable)
Maximum Length	(not applicable)
Null Count	2 ←
Blank Count	(not applicable)
Actual Type	double
Count	1628
Data Length	53 bit
Mean	114348.170972
Median	4888499.5 ←
Mode	0
Non-Null Count	1626
Nullable	YES
Ordinal Position	7
Decimal Places	0
Standard Deviation	429438.361236 ←
Standard Error	10649.778281

Figure 4 [8]

1.3.6 Business Rule Validation

Every organization has business rules. Whether they are basic lookup rules or complex rules with detailed formulas, Data profiling technology provides both customizable

and “out-of-the-box” methods for business rule validation. Pre-built business rules may provide domain checking, range checking, look-up validation or specific formulas. In addition to the canned data profiling validation techniques,

It allows us to check on many basic business rules at the point of data entry and, potentially, recheck these rules on an ad-hoc basis. Problems that arise from lack of validation can be extensive, for example, over-paying expenses, running out of inventory, or undercounting revenue. [8]

Finally with all these techniques provided in data profiling actually helps in understanding and discovering any non standard and inconsistent data in source files and gives information about structure of data which can be matched with business rules. These data sources even can be flat files. So far profiling techniques has been used to help for data management and data quality purposes. With profiling tools and software, data management responsible can discover all non standard, inconsistent and wrong entries in data as he will be viewing different report and statistics about data and structure, provided by profiling which we have mentioned earlier. Our thought are that with the help of these statistics and report we can develop a methodology and techniques to predict and suggest Logical Data Model of the data source. That will help data warehouse designer to understand the abstraction of data very quickly. It will save his so much time and he will be able to focus and concentrate only on his ETL and design work.

CHAPTER 2
LITERATURE SURVEY

2. Literature Survey

Important part of research is Literature Survey, as scenarios cannot be understood without it and its hard to understand that what point the researchers have reached and what are the loopholes in the topic and what can be enhanced in that area. Important part of literature survey is literature review of existing research in the targeted problem area. First we have to discover that what has been done so far and what are the vacant areas which need to be discovered. We have gone through multiple research papers, articles and books about data warehouse, flat file systems, data profiling and data modeling. We find lots of information and facts about all these areas and got remarkable guidelines to got the solution of our target research area. First we focus on the purpose of Logical Data Model, as if we are deriving Logical Data Model from flat file system then how it will help data warehouse designer to get understanding and vision about business data.

2.1 Purpose of Logical Data Model

There has been written lots of material about data modeling, lot of research has been done about logical data model. But every one discussed Logical Data Model as first step to design the data base system. When we are designing a database system, first we

analyze the business need and requirement, then construct logical data model. Then we move to construct the physical data model. B. A. Carkenord[15] discussed these two types of model, Logical Data Model and Physical Data Model and stated that the importance of data modeling in context of answering following questions[15].

What is a logical data model?

Who uses the logical data model?

How is a logical data model different than a physical data model?

What happens if you don't build a logical data model?

B. A. Carkenord[15] stated that good quality data structure is critical to a long lasting, easy to maintain system. A logical data model is a graphical representation of the information requirements of a business area, although it is not a database. A logical data model is independent of a physical, data storage device. This is the key concept of the logical data model. Data is the most important part of an application system. A good, strong, accurate data structure allows application developers to design any processing, user interface, reporting, or statistical analysis ever needed. The Logical Data Model refers to a higher level of the business data. Once we know the business data requirements, we can normalize and implement the data. The Business Area Experts own the logical data model. Barbra A. Carkenord[15] describe their data requirements to the data modeler and review the models created. Barbra A. Carkenord[15] use the models for impact analysis of changes to business requirements. The most important reason to build a logical data model is to confirm the users and analysts understand of the business requirements to assure that the system developed satisfies the business need. Logical data modeling provides the analyst with a tool and technique to conduct analysis.

Here it is important to note that construction of Logical Data Modeling is the phase of understanding business requirements. Although, next phase of it is to construct or develop physical data model. In our scenario we already have data in flat file system, which is designed by some other Database designer. This data is for the purpose of construction of data warehouse. Flat file data may not have structures. And to construct the target warehouse system it is very important to understand business data, business needs and requirements for development of target warehouse. Now how to get understanding from data in flat file system is a big issue. Quick and better understanding

can be achieved if we have Logical model of the data. As Logical Data Model is the abstract design of Database. But here process is reversed, as we have unstructured physical data and business rules. And we have to construct a Logical Data Model as it will help to get quick and thorough understanding of business data and requirements.

After realizing the importance and necessity of Logical Data Model in our case, second important issue is to understand the facts about flat file system and its structure.

2.2 Limitations of data in Flat File System

Its important to know the difference between data in DBMS system and data in flat file system. Although every DBMS system provide facility to import data from flat file system, but that data will not be having any other structural information like relations, referencing and indexing. Even that data may not be normalized and contains many redundant and non standard entries. All data will be in form of plain text. Just field and record decimeter will help to import data from that flat file system into DBMS system.

Strictly, a flat file database should consist of nothing but data and delimiters. More broadly, the term refers to any database which exists in a single file in the form of rows and columns, with no relationships or links between records and fields except the table structure. A flat file generally contains one record per line. Fields may simply have a fixed width with padding, or may be delimited by whitespace, tabs, commas or other characters. Extra formatting may be needed to avoid delimiter collision. There are no structural relationships. The data are "flat" as in a sheet of paper, in contrast to more complex models such as a relational database [18].

In the 1980s, configurable flat-file database computer applications were popular on DOS and the Macintosh. These programs were designed to make it easy for individuals to design and use their own databases, and were almost on par with word processors and spreadsheets in popularity. Examples of flat-file database products were early versions of FileMaker and the shareware PC-File. Some of these offered limited relational capabilities, allowing some data to be shared between files.

Today, there are few programs designed to allow novices to create and use general-purpose flat file databases. This function is implemented in Microsoft Works (available only for some versions of Windows) and AppleWorks, sometimes named ClarisWorks (available for both Macintosh and Windows platforms). Over time, products like Borland's Paradox, and Microsoft's Access started offering some relational capabilities, as well as built-in programming languages. Database Management Systems (DBMS) like MySQL or Oracle generally require programmers to build applications.

Flat file databases are still used internally by many computer applications to store configuration data. Many applications allow users to store and retrieve their own information from flat files using a pre-defined set of fields. Examples are programs to manage collections of books or appointments. Some small "contact" (name-and-address) database implementations essentially use flat files [23].

With unstructured and non relational information in flat file system, understanding of business data from flat file system is a big challenging issue. Specially if design is very complex, there exists lots of no of tables, millions of records, lots of redundant data. Then the job of warehouse designer to understand the model of data is very very difficult. You never know under what requirements structure of data is designed of OLTP system. In 1980s eras when relational databases were not developed and developer don't have facility to query across multiple tables. They have to make logic to extract data across multiple table for their reports. Many of developer make an extra table for the report. This is some thing redundant information stored in separate table. Same way structure of data can be very complex and can have many standard and type. To work with so much diversity, it is a very difficult and challenging job to extract quality data from these files [18, 20].

Your data might come from a COBOL database on a EBCDIC system. If so, you might use a statement of the following form to specify a packed data column as an internal decimal data column:

```
column_name decimal_definition EXTERNAL "packed_definition"
```

When data files are created on an EBCDIC system, you also need to specify EBCDIC as the argument to the CODESET keyword as well. (The default CODESET is ANSI.) The following external table definition shows how you might create an external table to convert fixed-length COBOL data from an EBCDIC data source to Informix internal format and write all rejected records to a specified file. Because no mode is specified, the database server will try to load data with this external table into an internal RAW or OPERATIONAL table [18].

Many informal documents exist that describe the CSV (Comma Separated Value) file format. The basic rules of CSV file format are as follows:

CSV is a delimited data format that has fields/columns separated by the comma character and records/rows separated by newlines. Fields that contain a special character (comma, newline, or double quote), must be enclosed in double quotes. However, if a line contains a single entry which is the empty string, it may be enclosed in double quotes. If a field's value contains a double quote character it is escaped by placing another double quote character next to it. The CSV file format does not require a specific character encoding, byte order, or line terminator format [23].

- Each record is one line terminated by a line feed or a carriage return, however, line-breaks can be embedded.
- Fields are separated by commas.

1997,Mata,CS350

- Leading and trailing spaces or tabs, adjacent to commas, are trimmed.

1997, Mata , CS350

same as

1997,Mata,CS350

- Fields with embedded commas must be delimited with double-quote characters.

1997,Mata,CS350,"An IT Professional"

- Fields with embedded double-quote characters must be delimited with double-quote characters, and the embedded double-quote characters must be represented by a pair of double-quote characters.

1997,Mata,CS350,"An" "IT" " Professional"

- Fields with embedded line breaks must be delimited by double-quote characters.

1997,Mata,CS350,"Go get one now
they are going fast"

- Fields may always be delimited by double-quote characters, whether necessary or not.

"1997",Mata,CS350

- The first record in a csv file may contain column names in each of the fields.

Year,Name,Course

1997,Mata,CS350

2000,Salman,TE502

The CSV file format is very simple and supported by almost all spreadsheets and database management systems. Many programming languages have libraries available that support CSV files. Even modern software applications support CSV imports and/or exports because the format is so widely recognized. Many applications in fact allow .csv-named files to use any delimiter character [23].

But now new technologies like data profiling has been developed and evolve which provide all statistical analysis to discover variety of information about structure and contents of the data. Although purpose of profiling was to ensure data accuracy before loading it into warehouse system. But, basis of our research work is that this statistical information provided by data profiling is good enough to predict and derive logical model from this flat file data. This information will also help to ensure the health of Logical Model even if structural information is missing from data.

2.3 Capabilities of Data Profiling

Data profiling consists of discovery techniques through the data and then provide lots of statistical information which is useful for understanding any inconsistency and inaccuracy exists in data. Basically data profiling inspects data for errors, inconsistencies, redundancies and incomplete information. After analysing data through its contents and structure discovery techniques, it helps data manager and control and improve the health and quality of data [7].

W. W. Eckerson [10] describes data profiling as a set of discovery techniques needed to improve the quality in data. He says in his paper that organizations need to take right decisions for their business promotion. And correct decisions depend upon their knowledge about customer behavior and attractions. Historical data of the organization contains that information. Sales, marketing, customer support and other initiatives require a reliable source of data about customers, products and other entities. This data forms the basis of both operational decisions (Does this customer own a certain product?) and customer analytics (Which customers are potential opportunities for up-sell).

W. W. Eckerson [10] further states that organizations frequently ignore the quality of the underlying data, leading to poor decisions, bad strategies and insufficient customer service. Before an organization begins a new data-driven initiative – such as enterprise resource planning (ERP) or customer relationship management (CRM) – it is important to address issues of data quality within existing data sources. He proposed a five-phase process that can help companies analyze, improve and control corporate data. As technology currently exists that allows organizations to improve and consolidate corporate information. These five components: data profiling, data quality, data integration, data enrichment and data monitoring are best addressed through a single platform, providing a unified view of any type of data, including customer, product and supplier information.

W. W. Eckerson [10] states that Data profiling helps you determine the current state of your data and gives you an idea of the best ways to correct or reconcile your information assets. Profiling consists of three primary phases, each designed to analyze

your data in different ways to show you the problems with the content and structure of your existing data. During the data profiling phase, you'll answer some important questions about your data that will help you shape a targeted, effective data quality and data integration strategy [10].

- Structure discovery – Do the data patterns match expected patterns? Does your data match the corresponding metadata?
- Data discovery – Are the data values complete, accurate and unambiguous? Is the data standardized according to your established conventions?
- Relationship discovery – Does the data adhere to specified required key relationships across columns and tables? Are there inferred relationships across columns, tables or databases? Is there redundant data [10]?

So data profiling provides a blueprint of data problems. Using the statistical information, the data quality phase may start for the process of building better data. This phase helps to correct errors, standardize information across tables and validate information that is inconsistent and inaccurate. And data can be only valuable if contains correct information. Data profiling ensure the correctness of data.

Data profiling can also used to analyze the behavior of transactions. I. V. Cadez [2], said that in data mining application transactional data must exists and examples might include market basket data in retail commerce, telephone call records in telecommunications, and web logs of individual page-requests at websites. The technique of Profiling is used to analyze the behavior of the transactions. Simple profiling techniques such as histograms do not generalize well from large transactional data set. In his paper I. V. Cadez [2], investigate the application of probabilistic mixture models to automatically generate profiles from large volumes of transaction data. In effect, the mixture model represents each individual's behavior as a linear combination of "basis transactions." I. V. Cadez [2] evaluate several variations of the model on a large retail transaction data set and shows the proposed model provides improved predictive power over simpler histogram-based techniques

In his paper, I. V. Cadez [10] describes that large transaction data sets are common in data mining applications. Typically these data sets involve records of

transactions by multiple individuals, where a transaction consists of selecting or visiting among a set of items, e.g., a market basket of items purchased or a list of which web pages an individual visited during a specific session.

He [2] says that as we are interested in the problem of making inferences about individual behavior given transaction data from a large set of individuals over a period of time, so we focus on techniques for automatically using *profiles* for individuals from the transaction data. In his paper [2] a profile is considered to be a description or a model of an individual's transaction behavior. Finding profiles is a fundamental problem of increasing interest in data mining, across a range of transaction-related applications: retail cross-selling, web personalization, forecasting, and so forth. The main focus of I. V. Cadez [2] is to find customer behavior using a data profiling technique using the transactional data. Still all the work has been done in the transaction processing domain.

But most of the work about profiling has been done in the data quality domain. Most of the research about data profiling emphasizes that data profiling techniques help the data management team to improve the quality of data with the help of information provided by data profiling. And data quality is very important for an organization to make correct decisions about business.

R. Lerner [5] emphasizes on the data quality of analytical systems as they give information about customers. And customer data integration comes from different data sources. The major challenges faced by organizations is the need to create a single, accurate, consistent, and timely view of their customers. While many organizations have some sense of the value of such a view, many organizations fail to grasp the effect of not having an accurate, complete view of their customers. Surprisingly, more than a few organizations feel that they can operate efficiently without taking extra steps to obtain such a view of their customers.

However, there is an approach that is specifically designed to help organizations get a complete view of their customer i.e., customer data integration (CDI). CDI solution combines a customer data repository, a tightly integrated data quality solution, and a services-oriented architecture (SOA) [5].

R. Lerner [5] stated that the benefits of an effective CDI solution are obvious, especially in terms of having customer data that is dependable and that can provide the background for making better business decisions. However, before focusing on this task, an organization should also consider its needs and then determine an appropriate CDI solution or strategy. Moreover, the organization may want to consider regulatory compliance and other issues, which are necessary for accurate, consistent and timely customer data.

In this paper R. Lerner [5] has focused more over the accuracy of customer registration to avoid data duplication in a distributed environment. Data profiling is used to achieve the objective of data quality only.

2.4 Derivation of Logical Data Model

So far we have given enough focus on purpose of logical data model, flat file system and problem in manipulation of data exists in flat file system and capabilities of data profiling discovery information. We have objective to derive a logical data model on the basis of discovery information provided by data profiling. Work has been done in deriving Logical data model from COBOL file system.

H. M. Edward [1] describes that CASE tools can be used to derive logical data model from COBOL file system and data design can successfully be recovered. H. M. Edward wrote his paper in 1993. Before that most of the development in business was done in COBOL systems. Then new Rapid Application Development (RAD) tools came in software industry and replace old COBOL systems. Year 2000 compliance issues arose, and become important to avoid any business loss. Many organizations start thinking about converting there current applications into new RAD technologies. Big issue was the data of COBOL system.

H. M. Edward [1] proposed the methodology to construct a logical model of data exist in COBOL using CASE tools. He said that maintenance of existing software systems has been calculated to be very expensive like in excess of £1 billion in the UK alone. At least 25% of this outlay is on systems that are in need of replacement or major

modification. One way of reducing these costs is to provide a practical method for design recovery. The recovered and documented system design would then provide a path into some systems development method (via the use of CASE tools) thus assisting the development of a modified or replacement system. Such a method has been developed by the authors and is termed the Reverse Engineering into CASE Technology method (RECAST).

H. M. Edward [1] says that RECAST is a comprehensive method for reverse engineering COBOL systems. In his paper he considers in detail one key aspect of RECAST: the derivation of the logical data model of the system in terms of its entities, their relationships and their attributes. The RECAST method defines two sets of rules for transforming the physical data accessed by the system into an Entity-Relationship-Attribute (ERA) model. One set of rules is used for file-based data whilst the other considers IDMSX databases. In this paper this aspect of the method is put into context by first giving a brief overview of the full RECAST method and then by providing a more detailed explanation of how the components that relate to the system data model are derived.

H. M. Edward [1] says that Reverse Engineering into CASE Technology method (RECAST) takes the source code for an existing COBOL system and derives a no-loss representation of the system documented in the format of Structured Systems Analysis and Design Method (SSADM) documentation. One key element of the method is the abstraction of the system data and its representation as a Logical Data Model. The RECAST method considers how to derive the entities (with their relationships and attributes) both for systems accessing file-based data and for those accessing data from IDMSX databases. The functional specification of the system is derived primarily by considering the processing that affects these entities [1].

H. M. Edward [1] describe four stages of RECAST method,

Stage BU: Identification of the Business Users' Views

Stage LDM: Identification of the Logical Data Model

Stage SP: Identification of the System Processing

Stage MD: Identification of the Menus and Dialogues

Stage BU is based in the domain of the business user, whereas Stages LDM, SP and MD are based in the domain of the system source code.

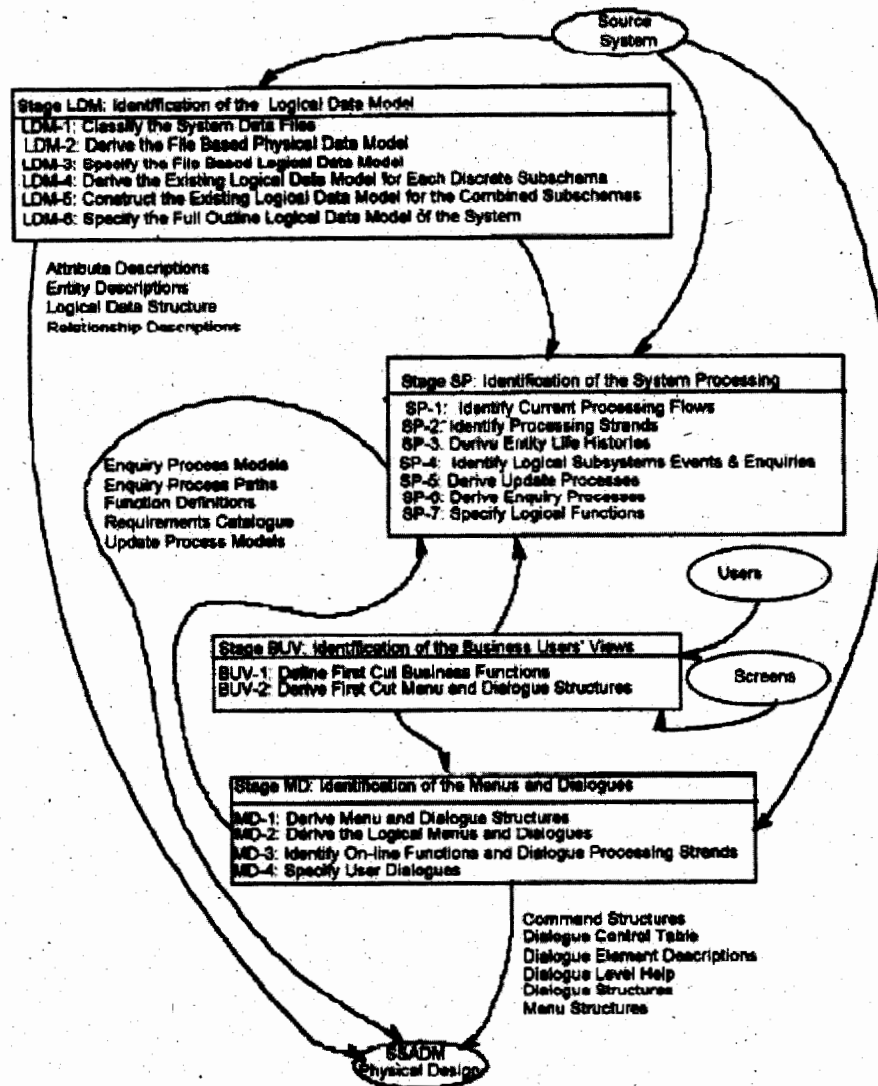


Fig 2-1 Stages, Steps and Products of RECAST [1]

Fig 2-1 describe all four stages of RECAST [1] method. Every stage has its sub stages. It important to not that this methodology does not take data in isolation for constructing Logical data model. For construction of logical model, it also taking input from Application interfaces, Menus and Dialogs and trying to under stand the application process and scenarios. Also it is taking Business user view of data as input through report screens and as well as identifying system processes. All input are used in CASE tools and then data model is derived. On big issue with this approach is that it is just focusing on COBOL system and not taking data in isolation.

But when we will be working on flat file system, we will only have data, not the application and system process associated with it. We even will not have knowledge about structure and relation of data. Also in the time RECAST method was proposed, techniques like data profiling on analytical data was not discovered. Now we have data profiling techniques, which analyze data thoroughly in many dimensions and provide very detail statistics about structure, contents and relationship in data. These statistics are very helpful in not only improving the quality of data but also in predicting the logical model of data.

CHAPTER 3
METHODOLOGY

3. Methodology

A research methodology defines what the activity of research is, how to proceed, how to measure progress, and what constitutes success. Different methodologies defined by distinct schools which wage religious wars against each other.

Methods are tools to be used. Don't let them use you. Don't fall for slogans that raise one above the others: "Research needs to be put on firm foundations;" "Philosophers just talk. " "You have to know what's computed before you ask how." To succeed at research, you have to be good at technical methods and you have to be suspicious of them. For instance, you should be able to prove theorems and you should harbor doubts about whether theorems prove anything.

The method section answers these two main questions:

1. How was the data collected or generated?
2. How was it analyzed?

In other words, it shows your reader how you obtained your results.

But why do you need to explain how you obtained your results?

- We need to know how the data was obtained because the method affects the results. For instance, if you are investigating users' perceptions of the efficiency of public transport in Bangkok, you will obtain different results if you use a multiple choice questionnaire than if you conduct interviews. Knowing how the data was collected helps the reader evaluate the validity and reliability of your results, and the conclusions you draw from them.
- Often there are different methods that we can use to investigate a research problem. Your methodology should make clear the reasons why you chose a particular method or procedure.
- The reader wants to know that the data was collected or generated in a way that is consistent with accepted practice in the field of study. For example, if you are using a questionnaire, readers need to know that it offered your respondents a reasonable range of answers to choose from (asking if the efficiency of public transport in Bangkok is "a. excellent, b. very good or c. good" would obviously not be acceptable as it does not allow respondents to give negative answers).
- The research methods must be appropriate to the objectives of the study. If you perform a case study of one commuter in order to investigate users' perceptions of the efficiency of public transport in Bangkok, your method is obviously unsuited to your objectives.
- The methodology should also discuss the problems that were anticipated and explain the steps taken to prevent them from occurring, and the problems that did occur and the ways their impact was minimized.
- In some cases, it is useful for other researchers to adapt or replicate your methodology, so often sufficient information is given to allow others to use the work. This is particularly the case when a new method had been developed, or an innovative adaptation used.

Some work is like science. You look at how people learn arithmetic, how the brain works, how kangaroos hop, and try to figure it out and make a testable theory. Some work is like engineering: you try to build a better problem solver or shape-from algorithm. Some work is like mathematics: you play with formalisms, try to understand their properties, hone them, prove things about them. Some work is example-driven, trying to explain specific phenomena. The best work combines all these and more.

3.1 Requirement 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 and expectations kept of software. In it problems are identified and then a possible solution is proposed.

3.2 Problem Definition

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

- Loading the flat file into the system which is used as an input.
- Detection of key on the basis of user option i.e. primary key or composite primary key.
- It is not necessary to develop the profiling Modules at the time of initial Development.
- Once all files have been loaded we will discover relationship among different entities using data profiling technique
- Rules creation module is always dependent on the data present in the system. And the data that is fed in the system will come from any existing OLTP.

TH-4389

- The LDM is constructed by applying the business rules. This module can be run according to the ease in such a way that it does not affect the performance of Operational System.

3.3 Use 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.

3.3.1 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:

3.3.1.1 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.

3.3.1.2 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	

3.3.1.3 File Loading

- a) Name: File Loading
- b) Actor: User
- c) Pre-Condition: Click load file button.
- d) Post Condition: File Loaded in a data set.
- e) Typical Course of Action:

Actor Action	System Response
1. Press load file Button	2. Acquiring File Path. 3. Load a File. 4. Display File..

- f) Alternate Course of Action:

Actor Action	System Response
No action	1a. Error Displays on screen. 2b. No File Loading.

3.3.1.4 Relationship Discovery

- a) Name: Execute Query
- b) Actor: User
- c) Pre-Condition: Files Loaded.
- d) Post Condition: Relationship discovered
- e) Typical Course of Action:

Actor Action	System Response
1. Press relationship discovery from menu	1. match every data table to all other data tables 2. Prompt against each relationship discovered 3. set relationship on the basis of user input 4. Relationship discovered.

- f) Alternate Course of Action:

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

3.3.1.5 Business Rules Creation

a) Name: Business Rule Creation

b) Actor: User

c) Pre-Condition: Files Loaded.

d) Post Condition:.

e) Typical Course of Action:

Actor Action	System Response
1. Select data table which needs changes on the basis of Business rules.	1. Select the type of the business rule 2. Create regular expression. 3. Execute expression. 4. Test Results displayed on Screen

f) Alternate Course of Action:

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

3.3.1.6 Apply Business Rules

- a) Name: Apply Business Rule
- b) Actor: User
- c) Pre-Condition: Business Rules created.
- d) Post Condition: Logical Data Model Construction.
- e) Typical Course of Action:

Actor Action	System Response
1. Select data table. 2. Click Apply Business Rules button	1. Entity Reduction is done. 2. Reduced entity will be displayed on Screen

- f) Alternate Course of Action:

Actor Action	System Response
	2b. Error message is displayed.

3.3.2 Use Case Diagram

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.

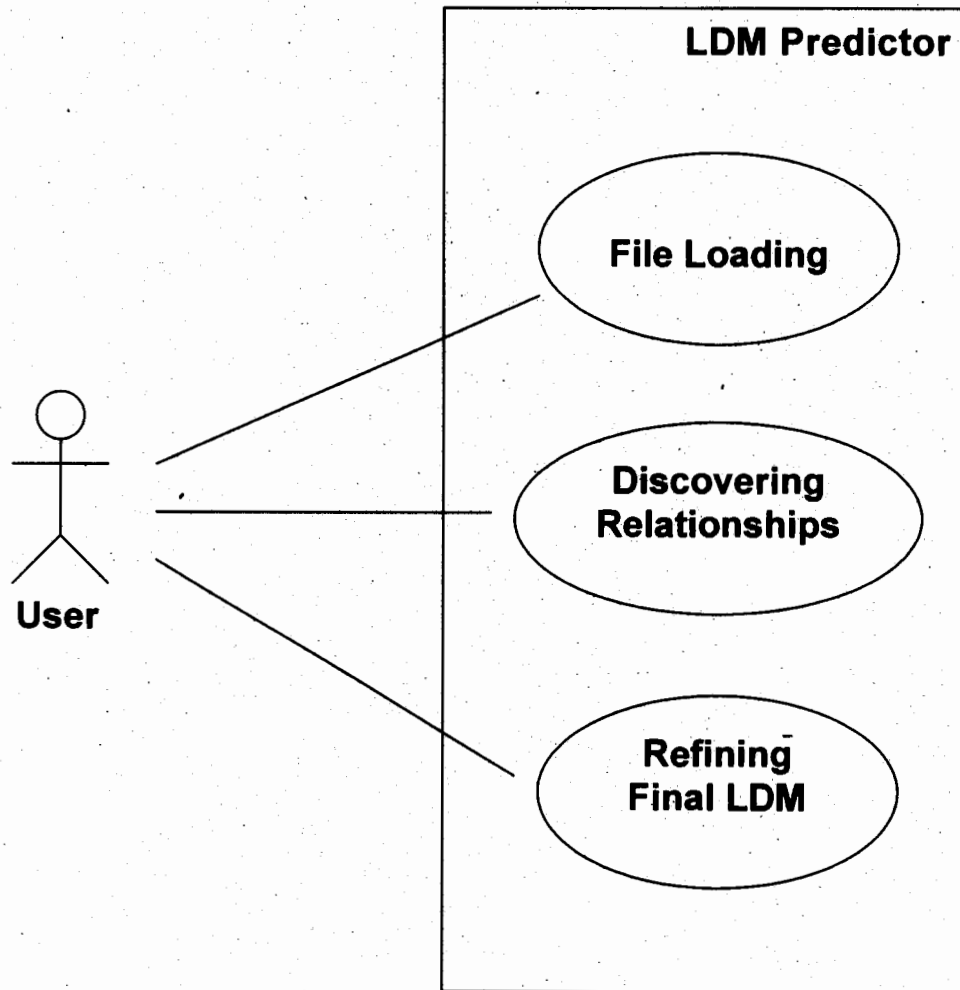


Fig 3-1 Use Case Diagram of LDM Predictor

3.4. Design

In this chapter we will discuss the System Design. A design is abstract solution in diagrammatic form of the problem. Basic proposed model of our solution is in following diagram.

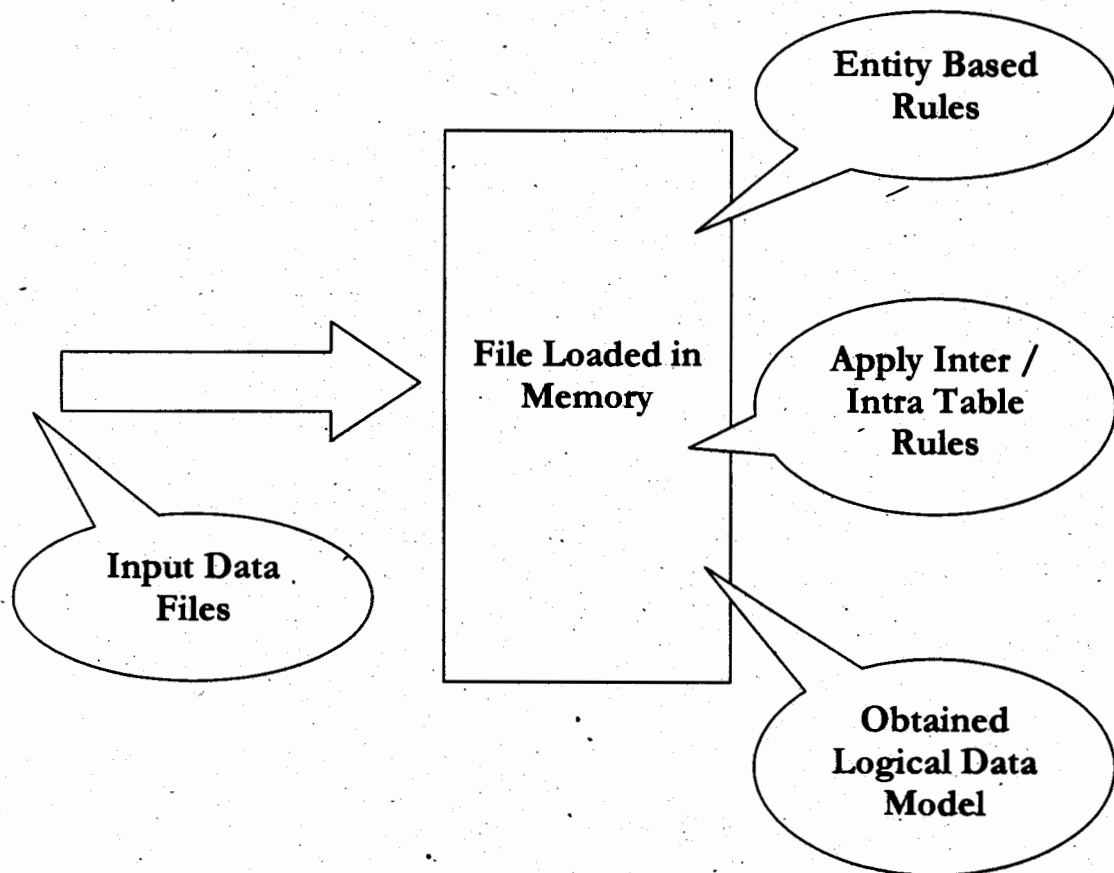


Figure 3-2 Proposed Model

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

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

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

3.4.1.1 State Transition Diagram

In Fig 3-3 diagram shows the state transition of the software module that will represent the time for queries and also will calculate the time differences.

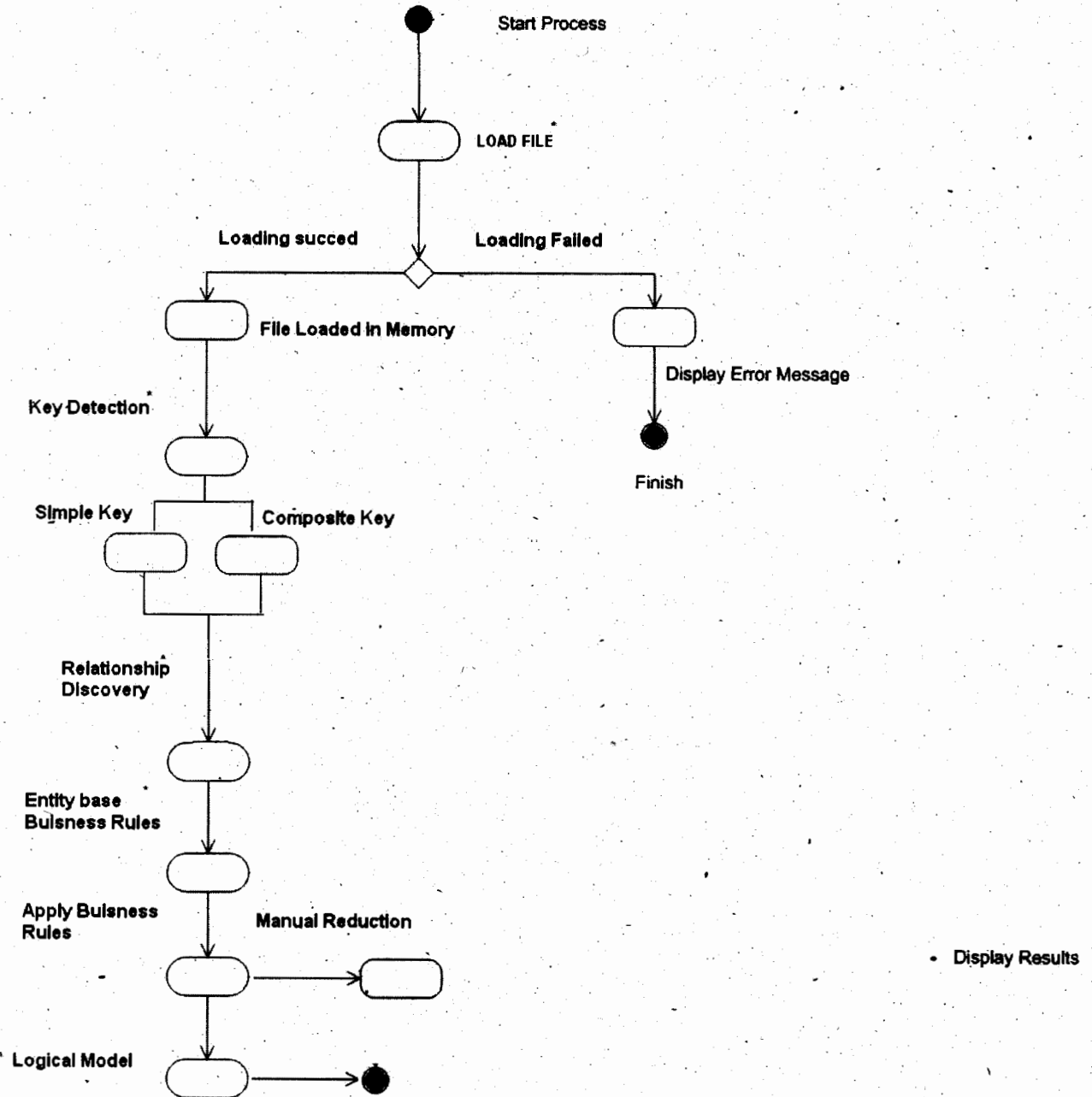


Fig 3-3 State Transition Diagram of Software Module

3.4.1.2 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 following Major events is shown below:

In Fig 3-4-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 3-4-b, the exit phase is shown.

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

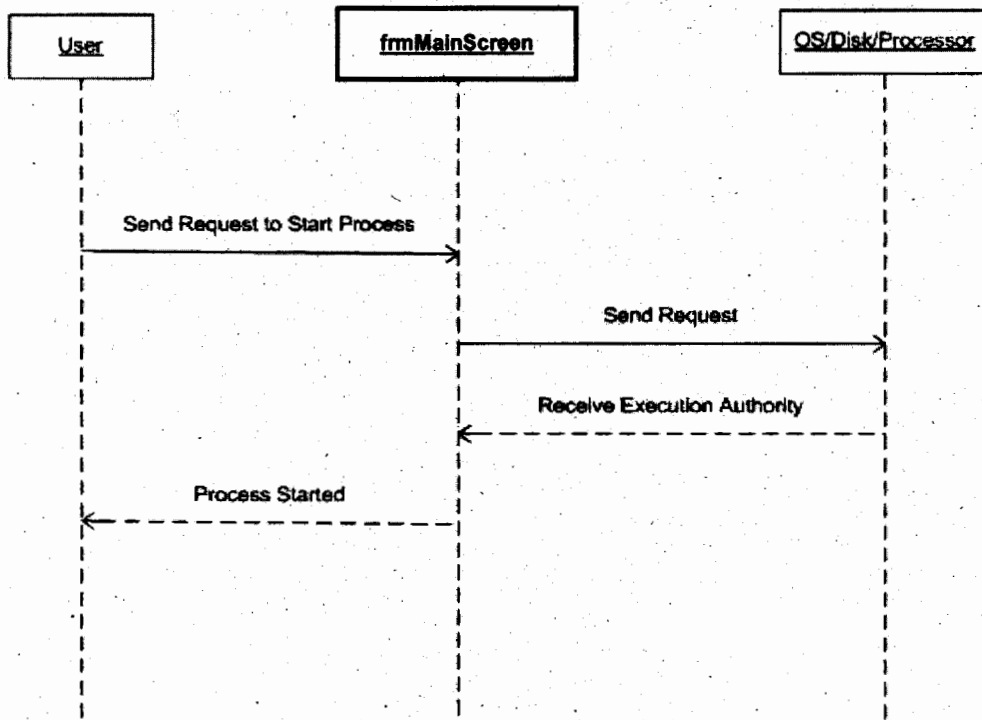


Figure 3.4-a: Start Process Sequence

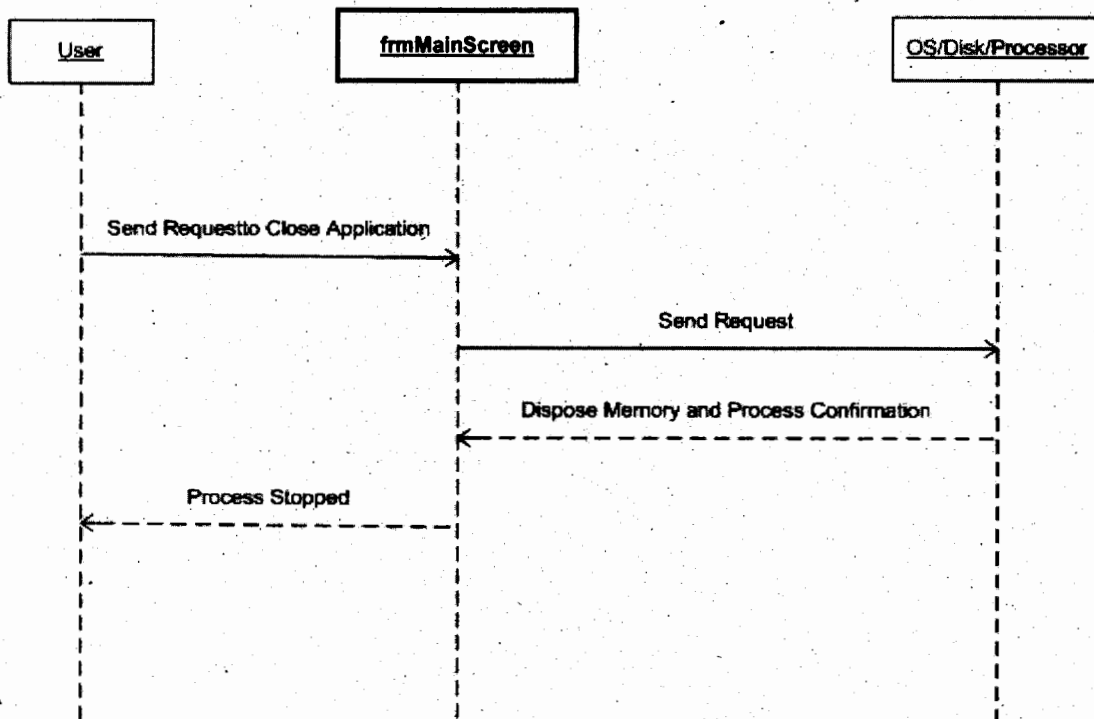


Figure 3.4-b: Stop Process Sequence

3.4.1.3 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 3-5 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.

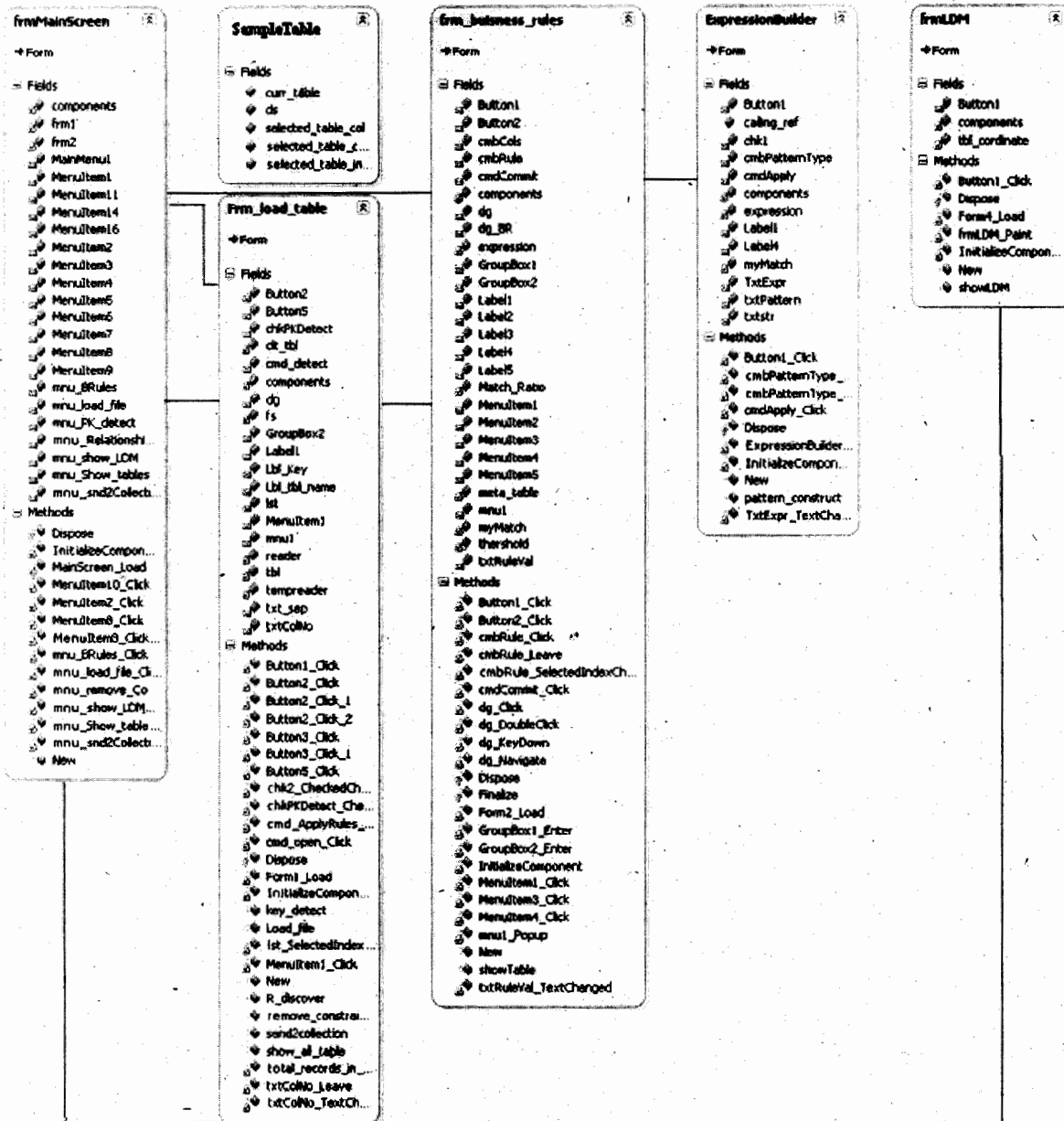


Fig 3-5 Class Diagram of Software Module

3.4.2 Data warehouse 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 3-6.

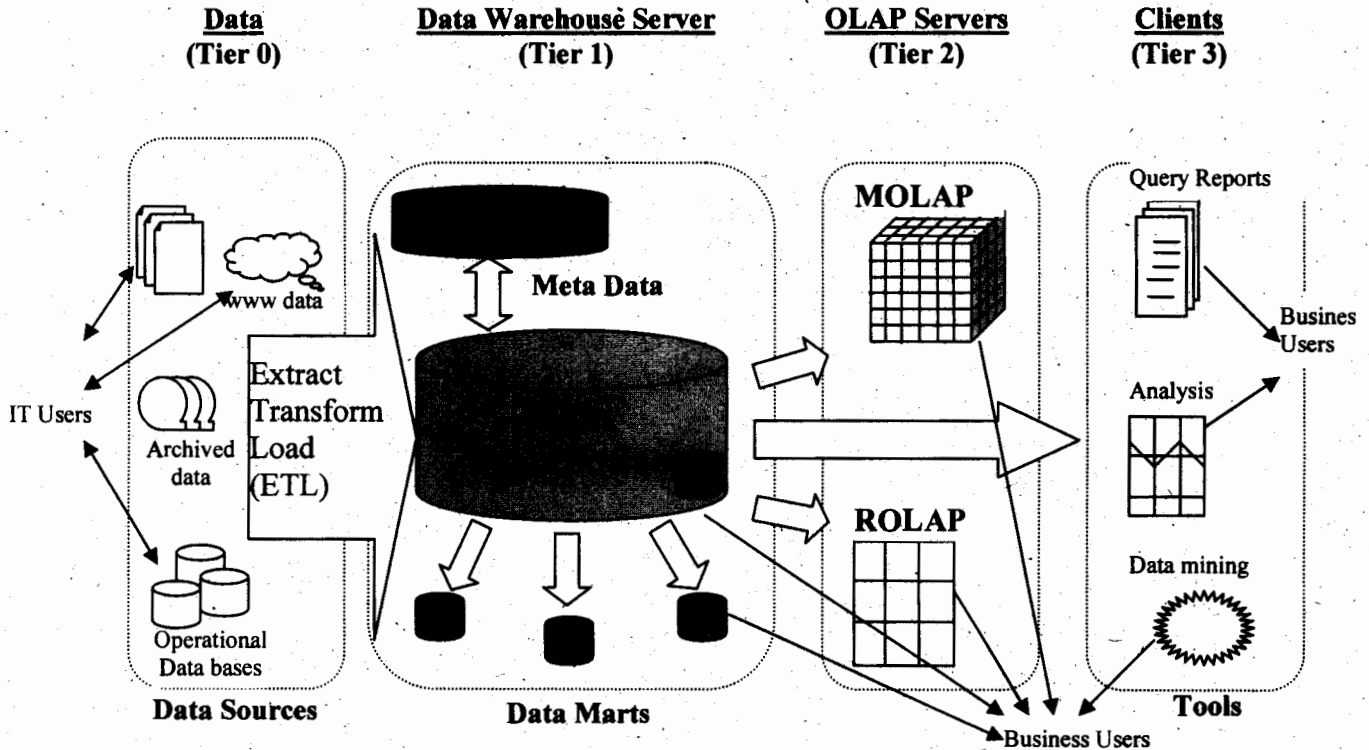


Fig 3-6: Architecture Design of DWH

3.5 Methodology

The main objective of this study is to establish a methodology for the construction of logical design from flat file. At present no reliable model exists for determining logical design from the flat file a proper optimization of these process as normally depends upon the combination of experience of the designer and expensive trials .The final design is dependant on business rules and statistics

Self developed methodology will be applied by which user will define business rule over flat file fields For example the Field "Age" has 3 characters and its data type is "Numeric" similarly user provide information about the "related field" of each entity and its business rules said Age Belongs to Employee only and Employee name has pattern (25Alphabets 25 Alphabets). In each entity with the help of business rules validation and data profiling we will identify the key field and other related field and build the entities and their relationships

Business Rules will be entity based that is complete rules for every entity is packaged together. Finally entity building is done in which we will identify relationship among entities on the basis of entity-to-entity business rules. All of the above mention process will be done using profile statistics. The name of this methodology will be predicted data modeling using data profiling. For our solution data profiling is a pre requisite for this study

3.5.1 Algorithm of Predicted LDM

- Step#1 Load all data files in the memory
- Step#2 select key constraints type
- Step#3 Apply key constraints
- Step#4 Relationship discovery using data profiling
- Step#5 Construction of Entity Based Business Rules
- Step#6 Apply Rules to construct LDM
- Step#7 Manual Reduction in entities if required

3.5.2 Mathematical Model of Algorithm

Let 'D' be the file set containing flat files

$$D = \{f_1, f_2, \dots, f_n\}$$

These files will be loaded in to Memory Tables

$$\text{Let } T_i \quad i=0..n$$

T_i contains Rows & column

$$T_i = R_j C_k \quad \begin{array}{l} i=0..n \\ j=0..n_1 \\ k = 0..n_2 \end{array}$$

Where R_j represent a tuple & C_k represent a data field

Let 'C' be the set of constraints

$$C = \{\text{Primary Key Constraints}\} \cup \{\text{Foreign Key Constraints}\}$$

For $y=1$ to n

 For $z=1$ to $T_y.\text{ColCount}$

 Begin

$(T_y.C_z = \text{unique}) \ \& \ (T_y.C_z \text{ Contains No Null value}) \rightarrow$ can be a Primary key

 Let K_i cols are detected with above criteria, where $K_i \in T_y.C_z$

where $z=1..Ty.colcount$

$K_i.ColCount=1 \rightarrow K_i$ is Primary key & expression be **PK**

Other wise

One of them is **PK** and remaining are Alternative keys

End

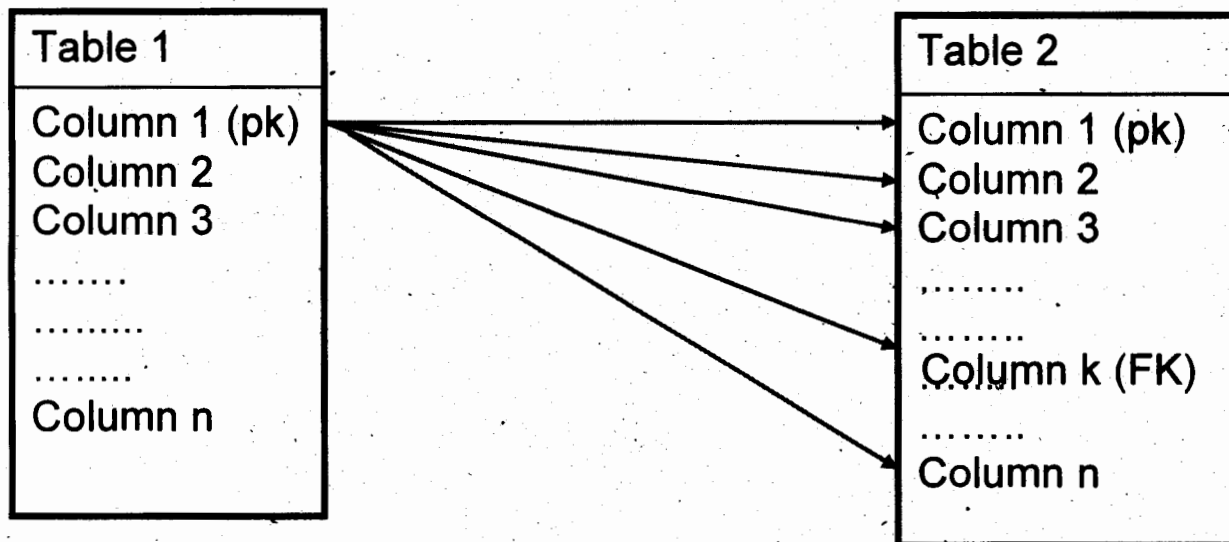


Fig: 3-7: Threshold Calculation by matching columns

Here If threshold of primary key of Table 1 with some Column k of Table2 is more then 95%, then it will strongly suggest it as relationship.

For L=1 to n

 For M= 1 to n

 For O= 1 to $T_M.ColCount$

 Begin

$Pk[L] \subseteq T_M.C_o \ \&\& \ T_M.C_o \subseteq Pk[L] \rightarrow$ Relationship Discovered from T_L to T_M

 It means $x \in PK [L] \rightarrow x \in T_M.C_o \ \&\& \ x \in T_M.C_o \rightarrow x \in PK [L]$

It will give 100 % threshold. Although this is ideal situation, here we are making prediction on the basis of threshold calculation. Where some orphan records can exist or even in master tables some record may exist which don't have any child.

Let above expression be $FK_{[M]}$

End

Entity Based Construction of regular expression

For $y=1$ to n

Begin

Construction of entity based Regular expressions which are used to define business rules/ patterns of characters. For any given set of characters Σ , a **regular expression over Σ** is defined by:

- The **empty string**, ϵ , which denotes a string of length zero, and means "take nothing from the input". It is most commonly used in conjunction with other regular expressions eg. to denote optionally.
- Any character in Σ may be used in a regular expression. For instance, if we write a as a regular expression, this means "take the letter a from the input"; ie. it denotes the (singleton) set of words $\{a\}$
- Any digit in Σ may be used in a regular expression. For instance, if we write 1 as a regular expression, this means "take the digit 1 from the input"; ie. it denotes the (singleton) set of words $\{1\}$
- The **Kleene closure** of a regular expression, denoted by $*$, indicates zero or more occurrences of that expression. Thus a^* is the (infinite) set $\{\epsilon, a, aa, aaa, \dots\}$ and means "take zero or more a from the input".

Let Pattern be some **RegX (Pattern)**

```
For z = 1 to Ty.ColCount
Begin
  For j = 1 to Ty.Rows
  Begin
    RegX (Pattern) = Ty.Cz Rj → pattern Match % increase for a
    column
  End
  Match = Thershold → Add column in a valid business rule collection
End
  Only select those column of ty which belongs to valid business rule
  collection
  Ty.Col ∈ Valid Business Rule
End
```

If required perform manual addition/reduction in the entity

Let 'R' be any regular expression which contain any type of expression may be a numeric or may be non numeric

CHAPTER 4
IMPLEMENTATION

4. Implementation

In this project we implemented the idea in VB.Net. We are supposed to generate the Logical Data Model from Flat files therefore we use GDI as well as developed procedures to populate the logical schema designed for the understanding of existing system.

4.1 Simulation Software

In order to show the results and compare the efficiency and performance simulation software of this project is developed in Microsoft Visual BASIC.Net. For profiling we use the SQLClient.dll library of the .Net frame work. The library provides instant and reliable connectivity with the database.

4.1.1 Declaration of Library

Using VB.net the following method is used for library declaration.

```
imports System;
imports System.Drawing
imports System.Collections
imports System.ComponentModel
imports System.Windows.Forms
imports System.Data
imports System.Data.SqlClient
Imports System.IO
```

4.1.2 Loading a flat File in a data set

```
Public Sub Load_file(ByVal txt_file As String)
    Dim local_tbl As New DataTable()
    Dim pk_cnst As UniqueConstraint

    Dim total_col = 0, rownum As Integer
    Dim curpos, nextpos As Integer
    Dim str As String

    Dim col As New ArrayList()
    Dim i, j As Integer
    fs = New FileStream(txt_file, FileMode.Open, FileAccess.Read)
    reader = New StreamReader(fs)
    rownum = 1
    curpos = 1
    tbl = local_tbl
    While (reader.Peek > -1)
        str = reader.ReadLine()
        nextpos = InStr(str, txt_sep.Text, CompareMethod.Text)
        col.Clear()

        While (nextpos <> 0)
            nextpos = InStr(str, txt_sep.Text, CompareMethod.Text)
            If (rownum = 1) Then
                total_col = total_col + 1
            End If
        End While
    End While
```

```

    If (nextpos <> 0) Then
        col.Add(Mid(str, curpos, nextpos - 1))
        str = Mid(str, nextpos + 1)
    Else
        col.Add(str)
    End If
End While

Try
    If (rownum = 1) Then 'schema building
        tbl.TableName = Mid(StrReverse(txt_file), 1,
InStr(StrReverse(txt_file), "\") - 1)
        tbl.TableName = StrReverse(tbl.TableName.ToString)
        Lbl_tbl_name.Text = tbl.TableName.ToString
        'MsgBox(tbl.TableName.ToString)

        For i = 1 To total_col
            Dim tbl_col As New DataColumn()
            tbl_col.Caption = col.Item(i - 1)
            tbl.Columns.Add(tbl_col)
            tbl_col.DataType = col.Item(i - 1).GetType
            tbl_col.ColumnName = col.Item(i - 1)
            'tbl_col.ColumnName = i - 1
        Next
    End If 'end schema building
Catch ex As Exception
    MsgBox("Unable to load the file because Duplicate
Column Name found", MsgBoxStyle.OKOnly)
    Exit Sub
End Try
'-----Row filling-----
/Dim row As DataRow
If (rownum > 1) Then 'skip 1st data row bcoz 1st row
=headings
    row = tbl.NewRow()
    For j = 0 To total_col - 1
        row(j) = col.Item(j)
    Next
    tbl.Rows.Add(row)
End If
'-----End row filling-----
rownum = rownum + 1
End While
dg.DataSource = tbl
tbl.AcceptChanges()
End Sub

```

4.1.3 Primary Key Detection

```

Public Sub key_detect()
    Dim j As Integer
    Dim pk(10) As DataColumn
    If (chkPKDetect.Checked = True) Then
        j = 0

        Dim temp As VariantType

        While (j < tbl.Columns.Count)
            Try
                pk(0) = tbl.Columns(j)
                tbl.PrimaryKey = pk
                temp = (MsgBox(" do you want  '" &
tbl.Columns(j).ColumnName & "' as Primary key ", MsgBoxStyle.OKCancel))
                j = j + 1
                If (temp = vbOK) Then
                    tbl.AcceptChanges()
                    chkPKDetect.Enabled = False 'reset all button
when new file is loaded
                    Lbl_Key.Text = " Key= " &
tbl.Columns(j).ColumnName.ToString
                    Exit While
                Else
                    tbl.RejectChanges()
                End If
            Catch ex As Exception

        End Try
        End While
    Else
        For j = 0 To lst.Items.Count - 1
            pk(j) = tbl.Columns((lst.Items(j)))
        Next
        Try
            tbl.PrimaryKey = pk
            tbl.AcceptChanges()
            MsgBox("CPK Successfully Created")
        Catch ex As Exception
            MsgBox(" Not a valid list for Composite Primary key")

        End Try
    End If
    -----End auto PK Detection-----
End Sub

```

4.1.4 Relationship Discovery

```

Public Sub R_discover()
    Dim i, j, k, kl As Integer

    Dim pk(10) As DataColumn
    Dim fkl(10) As DataColumn
    Dim fk As ForeignKeyConstraint

    Dim TblMaster, TblDetail As DataTable

    For i = 0 To ds.Tables.Count - 1
        TblMaster = ds.Tables.Item(i)
        pk = TblMaster.PrimaryKey()
        ReDim fkl(pk.Length - 1)
        For j = 0 To ds.Tables.Count - 1
            If (i <> j) Then
                TblDetail = ds.Tables.Item(j)
                For k = 0 To TblDetail.Columns.Count - 1
                    'find the no of length in case of CPK
                    For kl = 0 To pk.Length - 1
                        fkl(kl) = TblDetail.Columns(k + kl)

                        Next 'iterateviely select the next col until
relationship discovered or table ends

                Try
                    fk = New ForeignKeyConstraint("FK_" &
TblMaster.TableName.ToString, pk, fkl)
                    ds.Tables.Item(j).Constraints.Add(fk)
                    '-----Apply enforcement over the
dataset to ensure the fk use try catch
                    'clt_tbl.Item(j).AcceptChanges()
                    ds.EnforceConstraints = True

                    Dim str As String
                    str = "Relationship founded from Table ' "
& TblMaster.TableName.ToString & " ' To ' " &
ds.Tables.Item(j).TableName.ToString & " ' " & vbCrLf & " against
Following Cols "

                    For kl = 0 To fkl.Length - 1
                        str = str & " " &
(fkl(kl).ColumnName.ToString) & vbCrLf
                    Next
                    MsgBox(str)
                    If (MsgBox("do u want to apply this FK",
MsgBoxStyle.OKCancel)) = MsgBoxResult.Cancel Then

```

```

ds.Tables.Item(j).Constraints.Remove("FK_" &
TblMaster.TableName.ToString)
    End If

    Exit For
Catch ex As Exception
    'In case of invalid constraint
End Try
Next

End If

Next

Next

End Sub

```

4.1.5 Business Rules Validate

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim i, j, k As Integer
    Dim str As String
    Dim min_len, max_len As Integer
    'dg.DataSource = ds.Tables.Item(curr_table).Select(i)

    For k = 0 To meta_table.Rows.Count - 1

        expression = New Regex(meta_table.Rows(k).Item(2).ToString,
RegexOptions.IgnoreCase)

        For i = 0 To ds.Tables.Item(curr_table).Columns.Count - 1
            Match_Ratio = 0

            min_len =
Len(ds.Tables.Item(curr_table).Rows(0).Item(i).ToString)
            max_len =
Len(ds.Tables.Item(curr_table).Rows(0).Item(i).ToString)

            For j = 0 To ds.Tables.Item(curr_table).Rows.Count - 1
                'scan entire rows

                If
(Len(ds.Tables.Item(curr_table).Rows(0).Item(i).ToString) < min_len)
Then
                    min_len =
Len(ds.Tables.Item(curr_table).Rows(0).Item(i).ToString)
                End If
            Next j
        Next i
    Next k

```

```

        If
(Len(ds.Tables.Item(curr_table).Rows(0).Item(i).ToString) > max_len)
Then
            max_len =
Len(ds.Tables.Item(curr_table).Rows(0).Item(i).ToString)
        End If

        If (meta_table.Rows(k).Item(1).ToString <> "Max
Lenght") And (meta_table.Rows(k).Item(1).ToString <> "Min Lenght") Then
            For Each myMatch In
expression.Matches(ds.Tables.Item(curr_table).Rows(j).Item(i).ToString)
                If (Not (myMatch.ToString()) = "") Then
                    Match_Ratio = Match_Ratio + 1
                End If
                'output &= myMatch.ToString() & vbCrLf
            Next
        End If
    Next

    Match_Ratio = Match_Ratio /
ds.Tables.Item(curr_table).Rows.Count * 100
    If (Match_Ratio >= thershold) And
(meta_table.Rows(k).Item(1).ToString <> "Max Lenght") And
(meta_table.Rows(k).Item(1).ToString <> "Min Lenght") Then
        If (meta_table.Rows(k).Item(3).ToString) = "" Then
            meta_table.Rows(k).Item(3) = i
        Else
            meta_table.Rows(k).Item(3) =
meta_table.Rows(k).Item(3) & "," & i
        End If
    End If
    'write instructuons for len
    Try
        If (meta_table.Rows(k).Item(1).ToString = "Max
Lenght") And (max_len = CDec(meta_table.Rows(k).Item(2))) Then
            If (meta_table.Rows(k).Item(3).ToString) = ""
Then
                meta_table.Rows(k).Item(3) = i
            Else
                meta_table.Rows(k).Item(3) =
meta_table.Rows(k).Item(3) & "," & i
            End If
        End If
        'min len check
        If (meta_table.Rows(k).Item(1).ToString = "Min
Lenght") And (max_len = CDec(meta_table.Rows(k).Item(2))) Then
            If (meta_table.Rows(k).Item(3).ToString) = ""
Then
                meta_table.Rows(k).Item(3) = i
            Else
                meta_table.Rows(k).Item(3) =
meta_table.Rows(k).Item(3) & "," & i
            End If
        End If
    Catch ex As Exception

```



```

        'if col value is other than int exception
    End Try
Next
Next
End Sub

```

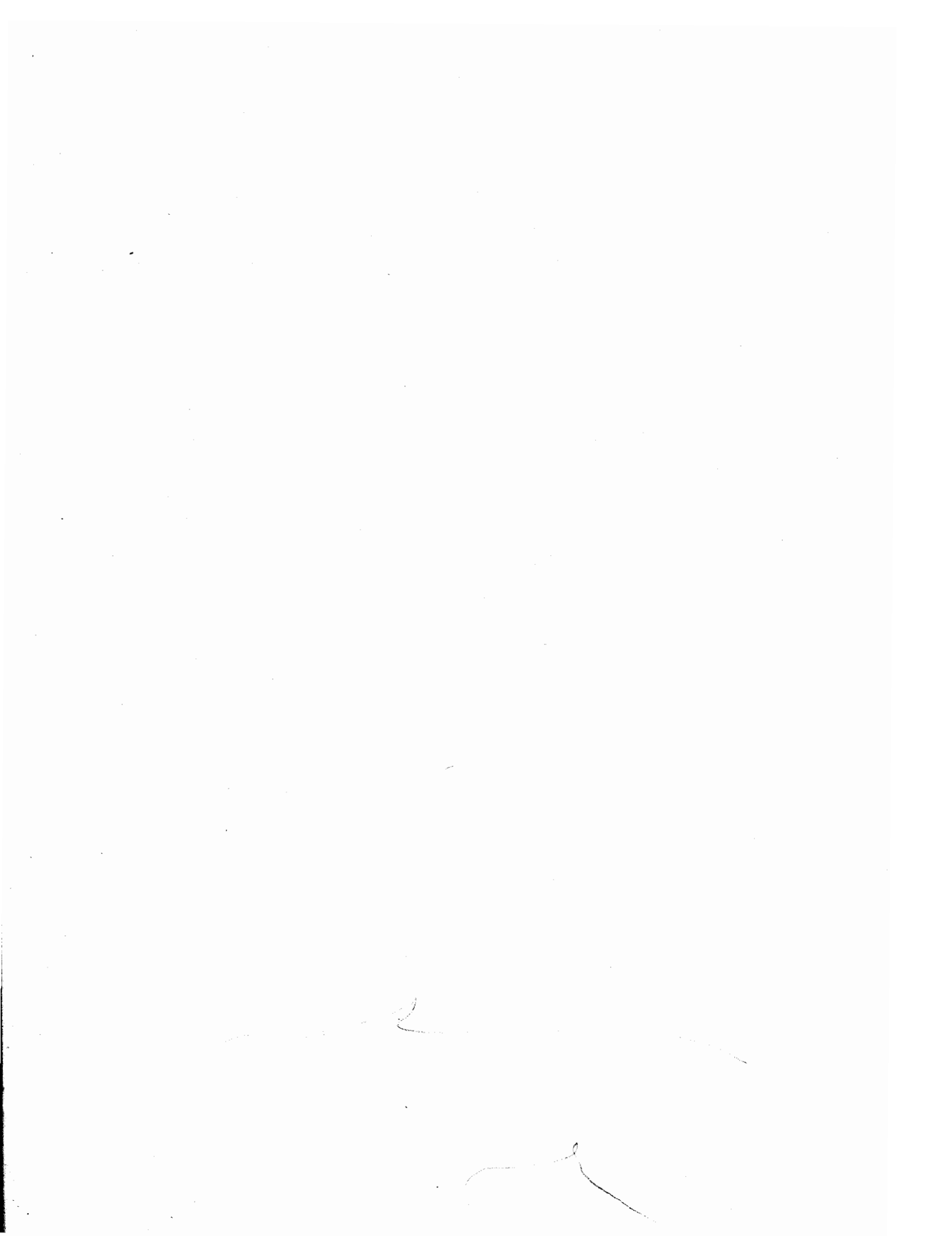
4.1.6 Business Rules Development

```

Sub pattern_construct()

    If (cmbPatternType.Text = "Any Single Character [other than New
Line]") Then
        TxtExpr.Text = TxtExpr.Text & "."
    End If
    If (cmbPatternType.Text = "Any Digit") Then
        TxtExpr.Text = TxtExpr.Text & "\d"
    End If
    If (cmbPatternType.Text = "Any Non Alphabet") Then
        TxtExpr.Text = TxtExpr.Text & "\W"
    End If
    If (cmbPatternType.Text = "Any Non_Digit") Then
        TxtExpr.Text = TxtExpr.Text & "\D"
    End If
    If (cmbPatternType.Text = "Any Alphabet") Then
        TxtExpr.Text = TxtExpr.Text & "\w"
    End If
    If (cmbPatternType.Text = "Any White Space[space, a tab, a
carriage return, a newline]") Then
        TxtExpr.Text = TxtExpr.Text & "\s"
    End If
    If (cmbPatternType.Text = "Any Non White Space[space, a tab, a
carriage return, a newline]") Then
        TxtExpr.Text = TxtExpr.Text & "\S"
    End If
    If (cmbPatternType.Text = "Find Repeated Occurances(Include
Empty)") Then
        TxtExpr.Text = TxtExpr.Text & "*"
    End If
    If (cmbPatternType.Text = "Find Repeated Occurances(Exclude
Empty)") Then
        TxtExpr.Text = TxtExpr.Text & "+"
    End If
    If (cmbPatternType.Text = "Find Zero or single Occurances")
Then
        TxtExpr.Text = TxtExpr.Text & "?"
    End If
End Sub

```



4.1.7 Exception Handling

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

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

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

4.1.8 Exiting from Application

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

CHAPTER 5

RESULTS

5. 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.

For testing our results and methodology, we got a four years old real time data from an Internet services related organization Pakistan Online. We got data in form of flat files. Data contained thousands of records. From our simulation software, it load this flat file data in memory, apply data profiling techniques, and on the basis of information provided by data profiling techniques, it first predict keys in relations. Than it predict relationship across the tables. While predicting relationship it also get help through business rules validation analysis. Which also help in reducing extra attributes from the tables. And in the end it try to construct a logical and abstract model of the data.

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

5.1 Main Screen

Test Run.exe Application is executed from the Application folder and following screen Fig 5.1 is displayed.

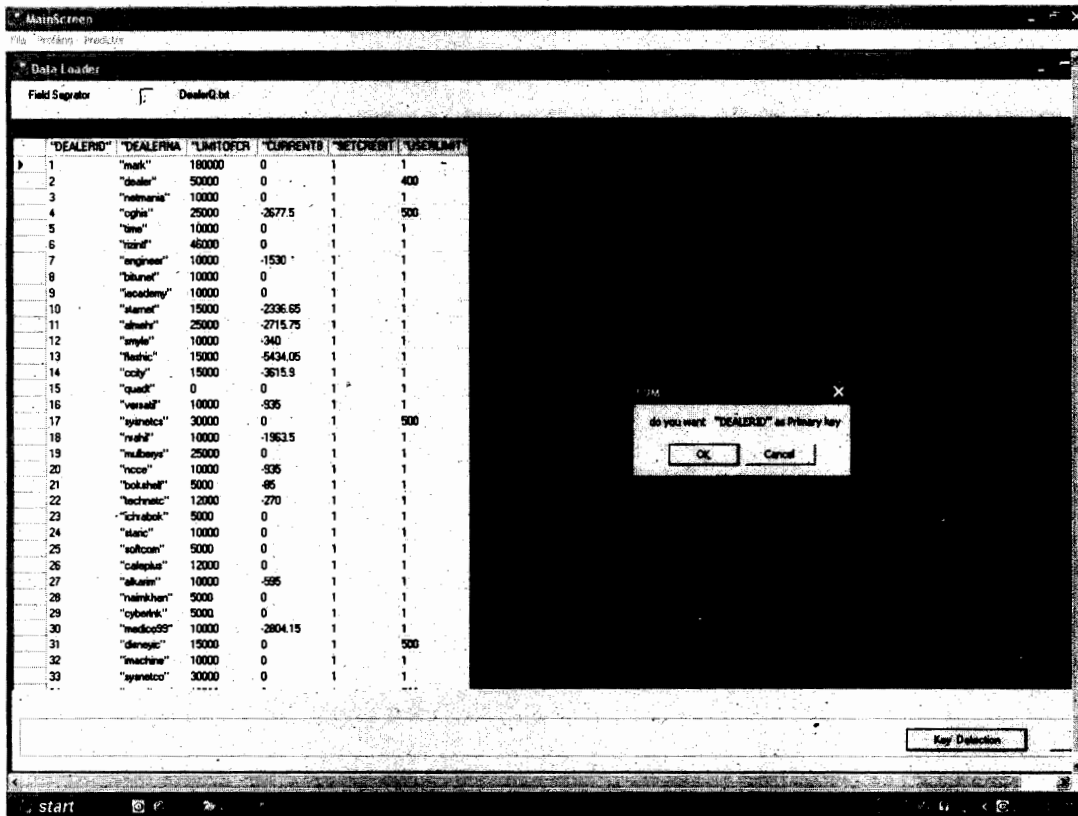


Fig 5.1 Data Loader Screen of Simulation Software

In this interface, it ask and load flat file in memory and transfer data into database table with field delimiter and record delimiter criteria. Then apply unique key constraint on each column using profiling techniques and on basis of information provided by profiling it predict primary key of the table first.

5.2 Relationship Discovery Screen

After file loading we will perform relationship discovery using data profiling technique. Process will start running and after completion of the process the following Fig 5.2 screen will be displayed.

The screenshot shows a software window titled "MainScreen" with a "Data loader" section. The "Field Separator" is set to "Comma" and the "Key" is "CUSTOMERID". A table displays customer data with columns: CUSTOMER, LOGINNAME, AEDNO, FNAME, MNAME, LNAME, ADDRESS, TEL, NIC, DATEACCD, and DEALERID. A pop-up dialog box is overlaid on the table, stating: "Relationship founded from Table 'DealerQ.tbl' To 'DealerPaymentQ.tbl' against Following Cols: 'DEALERID'". An "OK" button is visible in the dialog box. At the bottom right of the window, there is a "Key Detection" button.

CUSTOMER	LOGINNAME	AEDNO	FNAME	MNAME	LNAME	ADDRESS	TEL	NIC	DATEACCD	DEALERID
1	'halkhan'	1	'halkhan'						3/31/2000 10:0	
2	'ohna'	2	'Mica'	'Abdul'	'Majid'	'33 D New M	'5833668'		3/31/2000 10:0	
3	'wahmad'	3	'wahmad'						4/8/2000 10:10	
4	'mirza52'	4	'Aysha'	'Kama'	'Wase'	'66 G S II LA	'7723754'		4/10/2000 10:0	
5	'dasi'	5	'dasi'						4/12/2000 10:0	
6	'cdi'	6	'Wajid'	'U'	'Rehman'	'19 Block 2nd	'6361493'		4/14/2000 10:0	
7	'hosshah'	7	'hosshah'						4/14/2000 10:0	
8	'hend1'	8	'hend1'						4/17/2000 10:0	
10	'jabal'	10	'jabal'	'U'	'Rehman'	'43 Rehman	'5811952'	'273-85-0052'	4/21/2000 10:0	
11	'simba1'	11	'Zaashan'	'Ahmad'		'348 Hunez B	'7941425'		4/22/2000 10:0	
12	'New'	12	'New'						4/22/2000 10:0	
14	'shad52'	14	'shad52'						4/24/2000 10:0	
15	'aniam'	15	'Ans'		'Mi'	'Gulshan-e-R	'7410990'		4/24/2000 10:0	
16	'dawn1'	16	'dawn1'						4/25/2000 10:0	
17	'hocbati'	17	'Di'						4/25/2000 10:0	
18	'suchal'	18	'suchal'						4/25/2000 10:0	
19	'braheem'	19	'Muhammad'	'Ibrahim'					4/25/2000 10:0	
20	'qz'	20	'Ejaz'						4/25/2000 10:10	
21	'dpaly'	21	'dpaly'						4/25/2000 10:10	
22	'noc'	22	'noc'						4/25/2000 10:10	
23	'munar'	23	'munar'						5/10/2000 10:10	
24	'zac'	24	'zac'						5/6/2000 10:10	
25	'salma'	25	'Javed'		'Ahmad'	'28 UG Rakw	'6372317'		5/8/2000 10:10	
26	'newkhan'	26	'newkhan'						5/9/2000 10:10	
27	'bid'	27	'bid'						5/9/2000 10:10	
28	'tulase'	28	'Ali'		'Ahmad'	'33 B Agro S	'111707707'		5/10/2000 10:0	
29	'shakool'	29	'shakool'						5/11/2000 10:0	
31	'kasil'	31	'kasil'						5/15/2000 10:0	
33	'scd'	33	'Mudasa'		'Ali'	'465 I C 1 To	'5151729'		5/16/2000 10:0	
34	'mubasher'	34	'mubasher'						5/16/2000 10:0	
35	'loc'	35	'loc'						5/16/2000 10:0	
36	'narjee'	36	'Iqbal'	'A'	'Narjee'	'95 B Alid M	'6663629'		5/16/2000 10:0	
37	'zac'	37	'Khawaja'	'Asha'	'Saeed'	'1 A Durand	'6301164'	'265-88-1882'	5/16/2000 10:0	

Fig 5-2 Result Screen

Here discovery is done across the table. Data in the columns of two tables matched one by one. Profiling techniques give information about the threshold and this simulator predict any relationship exist among the tables by telling the threshold level.

5.3 Show LDM screen

After discovering the relationships among the tables, this simulator then predict LDM by indicating relationship among table and mentioned the primary/foreign key relationship exist among tables. This is initial step in constructing LDM.

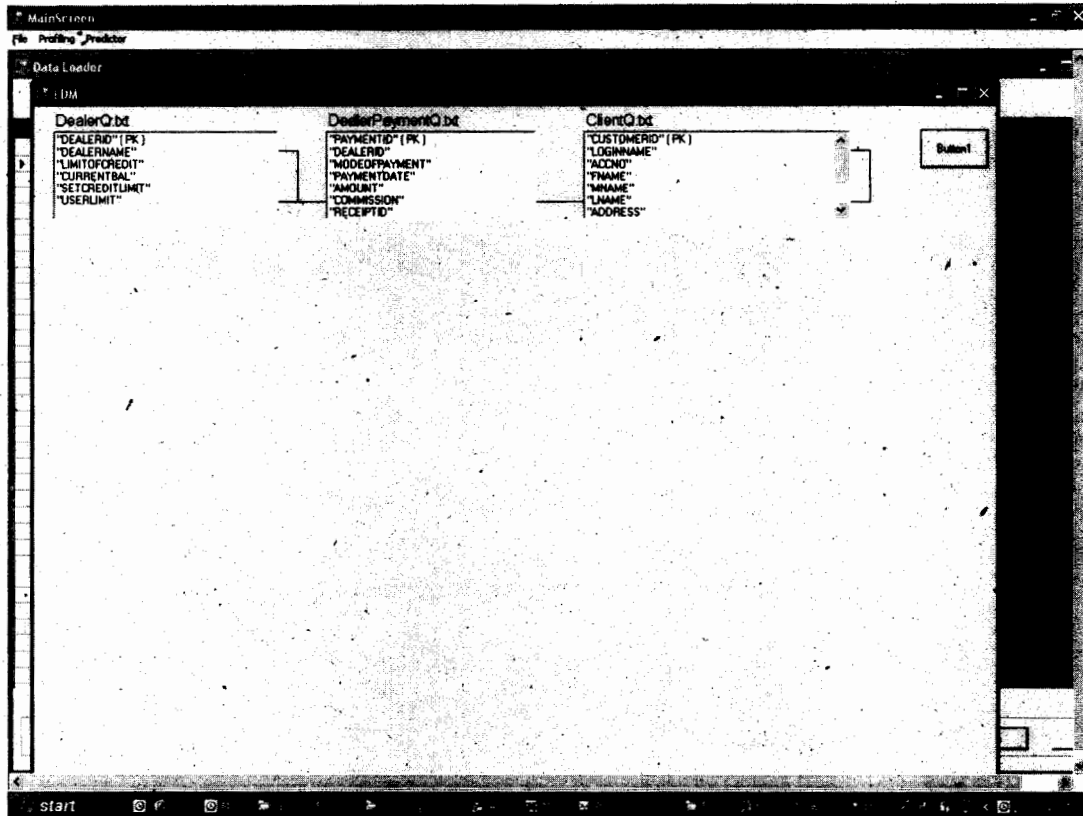


Fig 5-3 LDM after Relationship discovery

5.4 Business Rules Creation

Now next step is to refine initial LDM by applying business rules on the current data of different fields. Data Profiling techniques applied and with pattern analysis the all columns will be analyzed so that if those column meet the criteria specified in business rules. Here current interface provide facility to user to specify any business rules which can be applied.

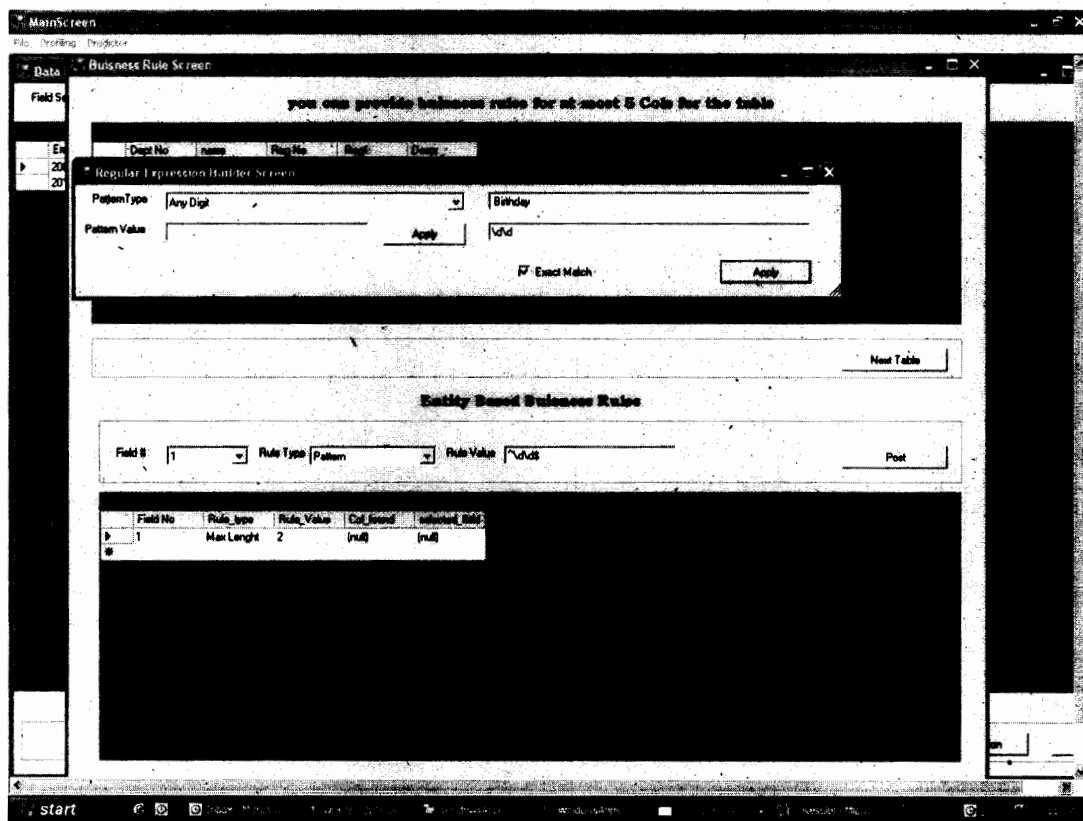


Fig 5-4 Business Rules Creation Screen

5.4 Business Rules Validation

Whatever business rules specified and provided in previous interface, those will be applied on different columns of data one by one to validate the business rules on existing data. Then validation related information will be provided using data profiling techniques.

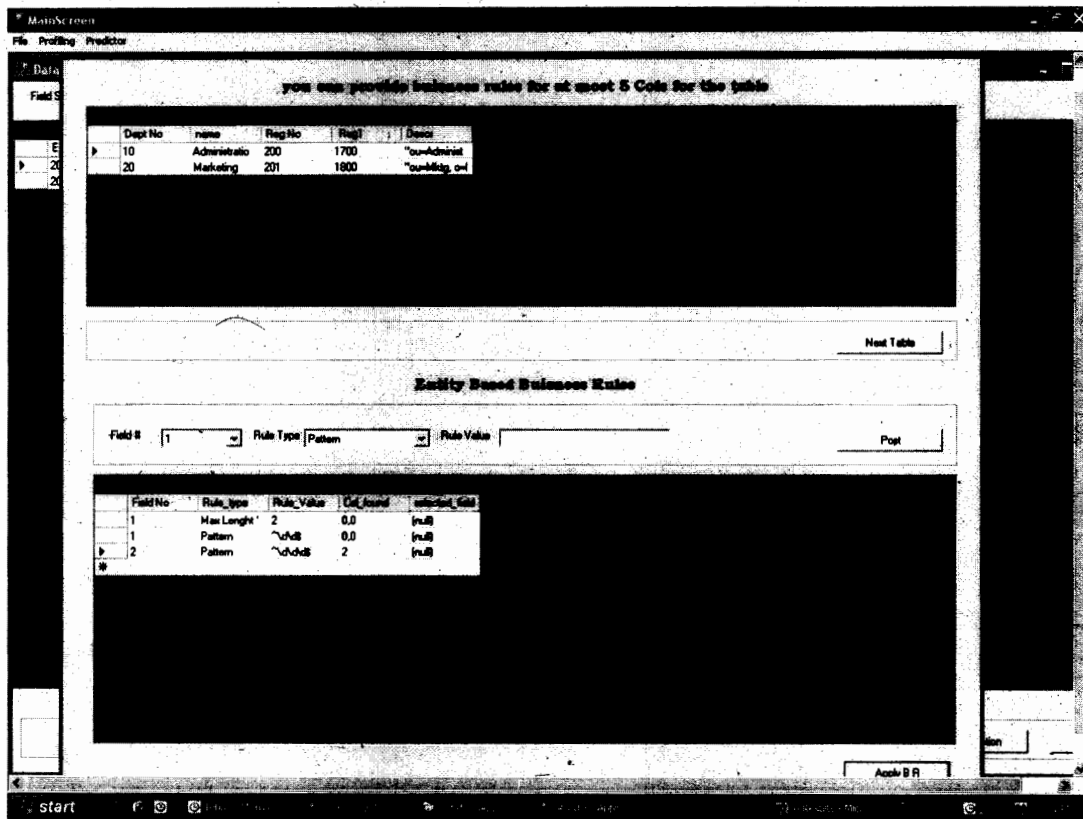


Fig 5-5-Buisness Rules Validate

5.6 Manual Reduction

On the basis of result of business rule validation provided, it will help data administrator to manually reduce the attributes which are not needed in tables.

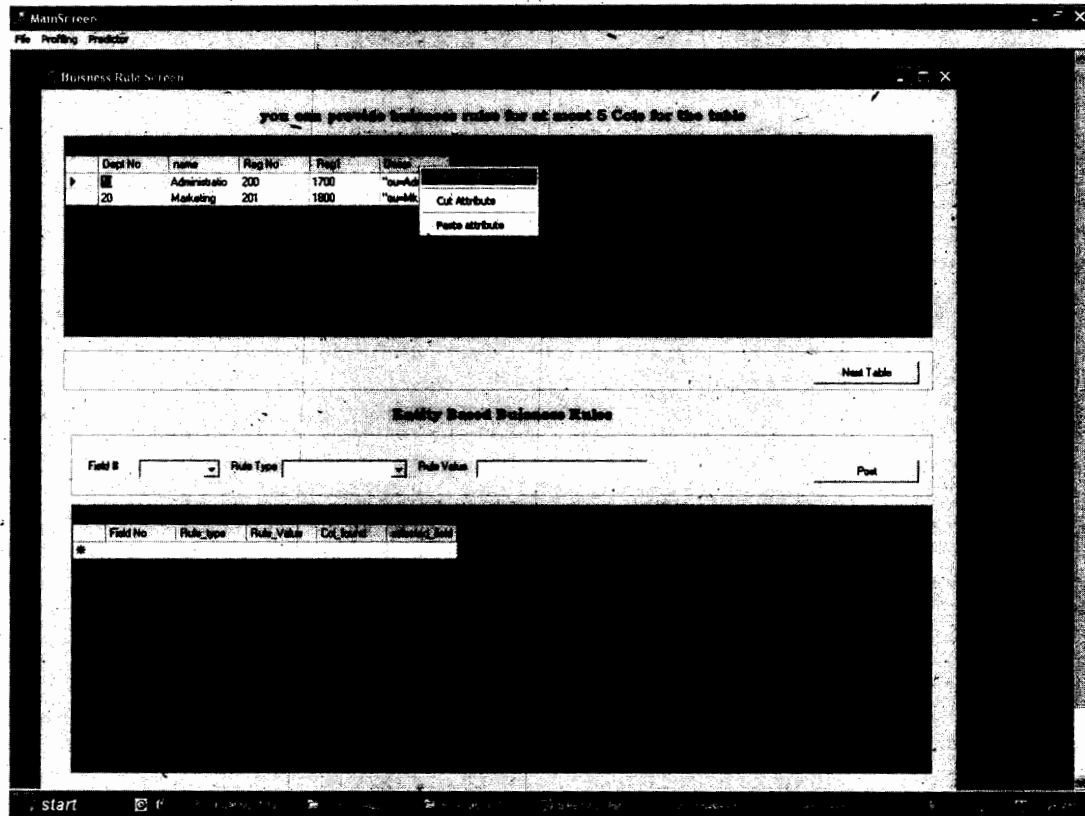


Fig 5-5- Manual Reduction

5.7 Manual Addition of Column

Same way on the basis of result of business rule validation provided, it will help data administrator to manually add attributes/columns to the tables.

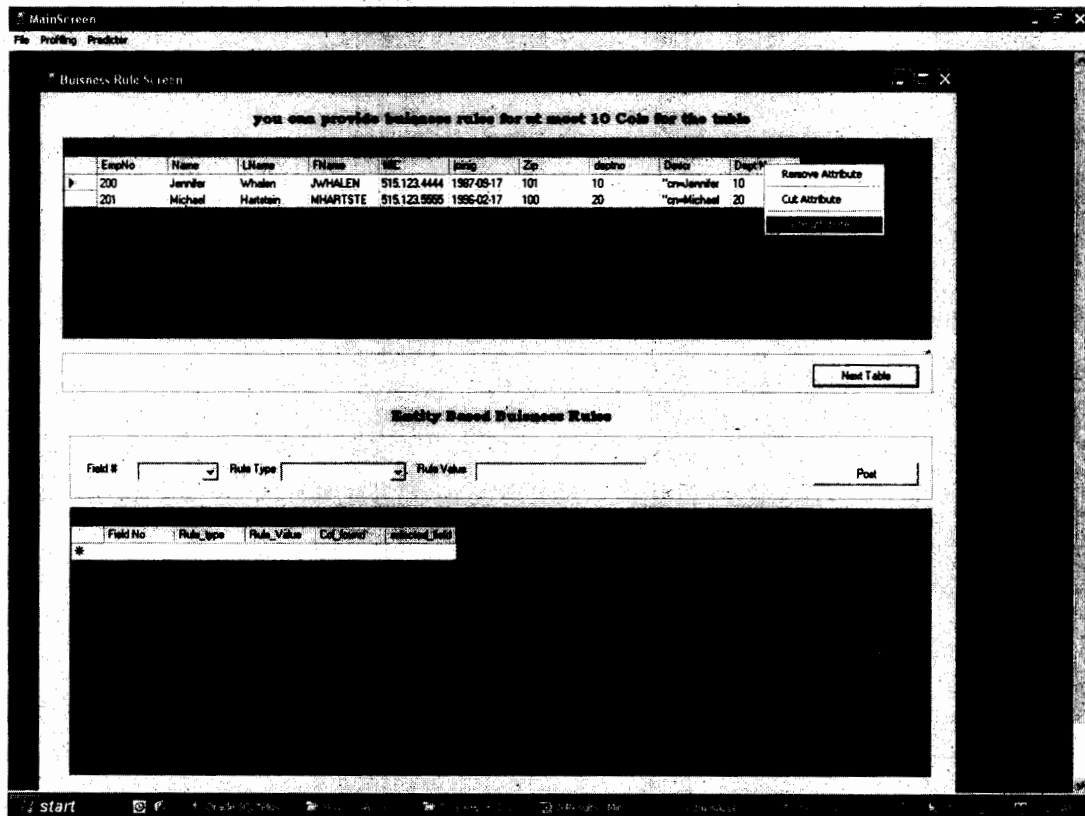


Fig 5-7-c Addition of a Column

5.8 Final LDM

Now every thing is ready for the final LDM. Primary keys has been identified, relationship across tables has been discovered and finally extra column has been removed and needed column has been added.

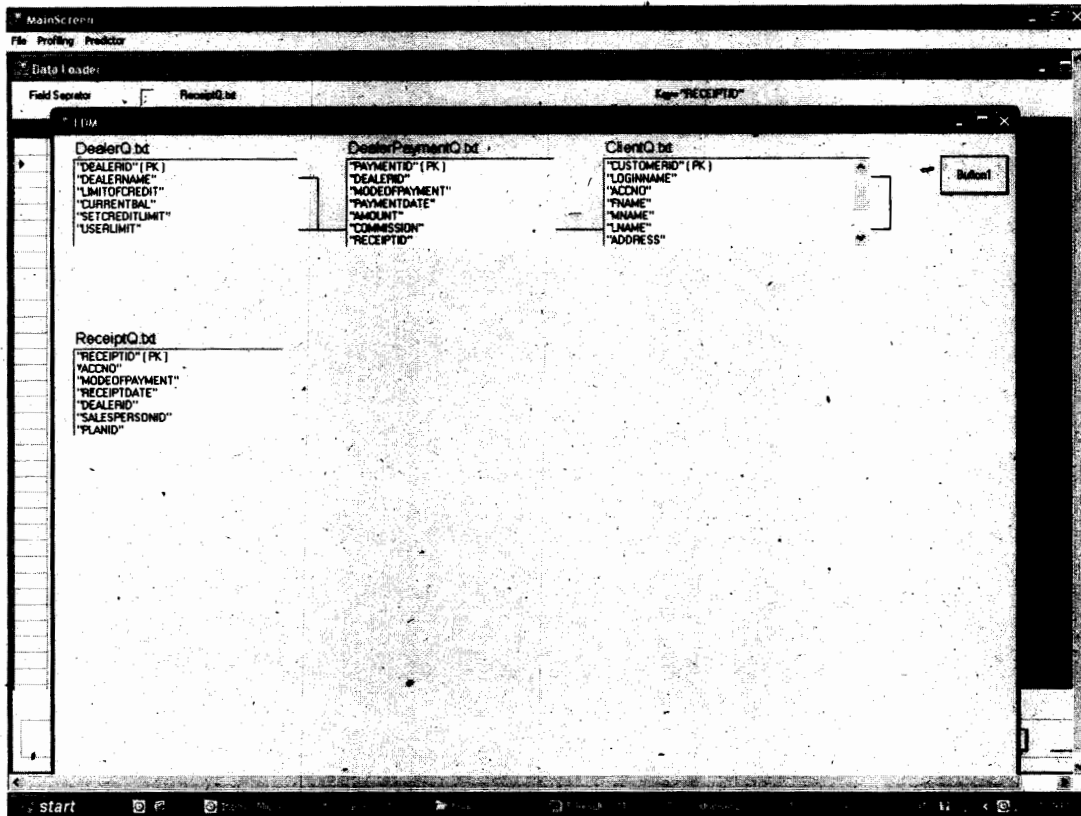


Fig 5-8 Final model

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6. Conclusion and Future Enhancements

Our main purpose was to construct a Logical Model from any data source and we take samples of data from unstructured flat files. With existence of structure, it is a very challenging job to discover structural information from data. Otherwise understanding of business data and data model is very difficult. Data profiling contains tremendous techniques to discover important information from data. It provides no of statistical information in many area existing in data. Although basic purpose of data profiling was to ensure quality and accuracy in data, but these statistical information are also very useful and may be use for other purposes like we have use these information for prediction of abstract model.

We have tested our methodology with simulation software which runs on the Pentium Machine with limited memory and slow processing speed. We used three years old sample data of an Internet service providing company Pakistan Online. It was a good enough data to test our methodology and research as it contains about 40 tables and thousands or in some cases millions of records in some tables. But we could only test this data with sample size of few thousands of records on simple Pentium machines. When we increased sample size to more than 30 thousands records, these machines took about 2 minutes to load the data and several minutes to derive statistical information using discovery techniques of data profiling, and result very slow processing by that time. Although better results can be obtained if the server machine is used with Xeon processor or Dual processing power system. But if samples size will be increased to millions then this algorithm will needs improvement in term of efficiency to run on even ordinary machines.

However as compared to the conventional Method like using CASE tools to derive Logical Model, this method shows very good results. These results come in seconds rather than spending several hours without the prior experience of the problem domain, when sample size was small. Also we had additional information through data profiling statistics, which are useful to verify the accuracy of our predicted LDM.

6.1 Conclusion

By applying the methodology we have achieved our target LDM prediction from flat file system. At present no reliable model exists for determining logical design from the flat file a proper optimization of these process as normally depends upon the combination of experience of the designer and expensive trials. The final design is dependant on business rules and statistics.

The project lies at the boundary of the practical industry problems and academic information analysis theory.

6.2 Future Enhancements

The research area is still open because we have only discussed the LDM Modeling for a relational database. There are some other areas that can be addressed like Object oriented Modeling by applying the same concept and efficiency of this algorithm if sample size of the data is very large. The information provided of data profiling can be more useful and value able if sample size will be very large. But in our algorithm, large sample size effect processing speed. Some code optimization techniques can be applied to make profiling techniques on large sample data more efficient.

REFERENCES AND BIBLIOGRAPHY

A. References and Bibliography

- [1] H. M. Edwards and M. Mum-o, Deriving a Logical Data Model for a System Using the RECAST Method. Proceedings of the Second Working Conference on Reverse Engineering 0-8186-7111-4, (1995).
- [2] I. V. Cadez, Breese J. S. and Heckerman D., Probabilistic Modeling of Transaction Data with Applications to Profiling, Visualization, and Prediction. Proceedings of the 1~th Conference on Uncertainty in Artificial Intelligence, San Francisco, (1998).
- [3] A. Kendapadi, M. Gandolfo and A. Shukla, BatchFlow: A Method And Notation To Visualize RDBMS Batch Jobs. ACM SIGSOFT Software Engineering Notes Volume 30 Number 3 Page 1, (2005).
- [4] J. E. Olsen, Data Quality: The Accuracy Dimension. Morgan Kaufmann Publishers, (2003).
- [5] R. Lerner, Using Data Qaulity Integration to Build a Single, Accurate and Consistent Customer View. A DataFlux White Paper, (2004).
- [6] R. Lerner, Data Profiling: The Diagnosis for Better Enterprise Information. A DataFlux White Paper, (2004).
- [7] D. Loshin, Master Data Management: Challenges to Success. A DataFlux White Paper, (2003).
- [8] W. W. Eckerson, Enterprise Data Management Maturity Model. A DataFlux White Paper, (2003).
- [9] D. Loshin, Global Paper Manufacturer Enhances Data Quality During Corporate-wide ERP Implementation. A DataFlux White Paper, (2005).

- [10] W. W. Eckerson, Data Profiling: Minimizing Risk in Data Management Projects. A DataFlux White Paper, (2005).
- [11] D. L. Moody and G. G. Shanks, What Makes a Good Data Model? Evaluating the Quality of Entity Relationships Models. Proc. 13th Intl. Conf. on Conceptual Modelling (E/R '94), pp. 94.111, (1994).
- [12] D. L. Moody and G. G. Shanks, Improving the quality of data models: empirical validation of a quality management framework. Information Systems, v.28 n.6, p.619-650, (2003).
- [13] D. L. Moody, Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data & Knowledge Engineering, v.55 n.3, p.243-276, (2005)
- [14] R. Kimball, Surprising Value of Data Profiling, Design Tips. A white paper at <http://www.kimballgroup.com>, (2004)
- [15] B. A. Carkenord Why Build A Logical Data Model – The Knowledge Exchange Company White Paper, (2001)
- [16] T. Connolly and C. Begg., Database Systems: A Practical Approach to Design, Implementation, and Management (3rd Edition). Addison Wesley, (2001)
- [17] A. Lou, The Essential Guide to Data Warehousing. Englewood Cliffs. Prentice Hall, (2000).
- [18] A. Ebert, Siemens A. G., Automatic Migration of Files into Relational Databases. Uwe Hohenstein &, Corporate Technology, (1998).
- [19] T. C. Redman, Data Quality For The Information Age. Artech House Boston, (1997).
- [20] B. Michael, Data Sharing Using a Common Data Architecture. John Wiley & Sons, (1994).

- [21] B. Michael, J. A. Berry and G. S. Linoff, *Mastering Data Mining*. John Wiley & Sons, (2000).
- [22] C. J. Date, *What Not How: The Business Rule Approach to Application Development*. Addison-Wesley, (2000).
- [23] E. Ahmad, M. Rusinkiewicz, and A. Sheth, *Management of Heterogeneous and Autonomous Database Systems*. Morgan Kaufmann, (1999).
- [24] H. Michael, *Beyond Reengineering*. HarperCollins Publishers, (1996).
- [25] H. Michael and J. Champy, *Reengineering the Corporation*. HarperCollins Publishers, (1993).
- [26] H. Michael, and S. A. Stanton, *The Reengineering Revolution*. HarperCollins Publishers, (1995).
- [27] M. David, *Building and Maintaining the Meta Data Repository*. Wiley Computer Publishing, (2000).
- [28] M. Howard, *Reengineering Legacy Software Systems*. Digital Press, (1998).
- [29] P. Dorian, *Data Preparation for Data Mining*. Morgan Kaufmann, (1999).
- [30] M. Reingruber and W. Gregory, *The Data Modeling Handbook: A Best Practice Approach to Building Quality Data Models*. John Wiley & Sons, (1994).
- [31] R. Ronald, *Business Rule Concepts*. Business Rule Solutions, (1998).
- [32] R. Ronald, *The Business Rule Book*, 2d ed. Business Rule Solutions, (1997).
- [33] A. Tannenbaum, *Metadata Solutions*. Addison Wesley, (2001).
- [34] J. A. Hoffer, *Modern Database Management (7th Edition)*. Prentice Hall, (2004).
- [35] T. J. Toby, *Database Modeling & Design (3rd Edition)*. Morgan Kaufmann (1999)

- [36] W.H. Inmon, **Building the Data Warehouse** (2nd Edition). Katherine Schowalter, (1996).
- [37] B. H. Michael, **Data Resource Quality: Turning Bad Habits into Good Practices**. Addison Wesley Longman, (2000).
- [38] T. C. Redman, **Data Quality: The Field Guide**. Boston Digital Press, (2001).