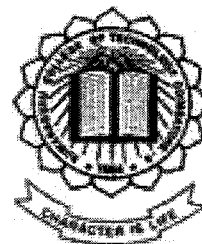




P-2288



HIDING SENSITIVE ASSOCIATION RULES

By

O. SASIKALA

(Registration Number: 71205621040)

Of

KUMARAGURU COLLEGE OF TECHNOLOGY

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

*In partial fulfillment of the requirements
for the award of the degree*

of

MASTER OF COMPUTER APPLICATIONS

ANNA UNIVERSITY
CHENNAI 600 025

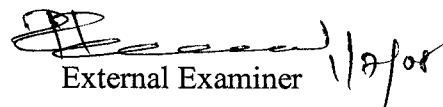
July 2008

BONAFIDE CERTIFICATE

Certified that this project report titled “**Hiding Sensitive Association Rules**” is the bonafide work of Ms. **O.SASIKALA** (Registration Number: **71205621040**) who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**Supervisor****Head of the Department**

Submitted to Project and Viva Examination held on 1.7.2008

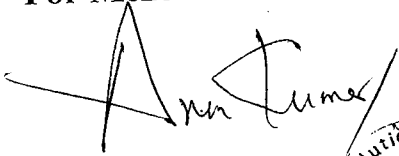
**Internal Examiner****External Examiner**

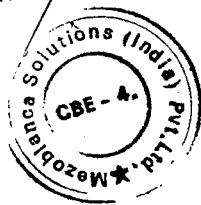
MEZOBLANCA SOLUTIONS

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Miss. O.SASIKALA** (Reg no: 71205621040), final year student of **M.C.A** from **KUMARAGURU COLLEGE OF TECHNOLOGY** has been given an opportunity to do her final year project entitled "**HIDING SENSITIVE ASSOCIATION RULES**" using **VB.NET** technology in our Organization from December 2007 to May 2008

For Mezoblanca Solutions (India) Pvt Ltd,


DIRECTOR.



Mezoblanca Solutions (India) Pvt. Ltd.,
204, PSG Step Software Park II, Peelamedu, Coimbatore - 641 004. India.
Tele : (91) (422) 4363316, 4363318

ABSTRACT

Data mining techniques have been widely used in various applications. However, the misuse of these techniques may lead to the disclosure of sensitive information. Researchers have recently made efforts at hiding sensitive association rules.

The Hiding of Sensitive Rules is done by using Modification Schemes. Nevertheless, undesired side effects, e.g., non sensitive rules falsely hidden and spurious rules falsely generated may be produced in the rule hiding process.

In this system, our approach strategically modifies a few transactions in the transaction database to decrease the supports or confidences of sensitive rules without producing the side effects. Since the correlation among rules can make it impossible to achieve this goal, we propose heuristic methods for increasing the number of hidden sensitive rules and reducing the number of modified entries.

The experimental results show the effectiveness of our approach, i.e., undesired side effects are avoided in the rule hiding process. The results also report that in most cases, all the sensitive rules are hidden without spurious rules falsely generated. Moreover, the good scalability of our approach in terms of database size and the influence of the correlation among rules on rule hiding are observed.

ACKNOWLEDGEMENT

I would like to express my gratitude to our principal **Dr. Joseph V. Thanikal, Kumaraguru College of Technology, Coimbatore** for the successful completion of the project.

I like to take this opportunity to show my sincere thanks to **Dr. M. Gururajan, Head of the Department, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore** for encouraging me during the project.

I wish to express my thanks to **Mrs V.Geetha, Assistant Professor , Department of Computer Applications, Kumaraguru College of Technology, Coimbatore** for her guidance and support given to me throughout the project.

I very much thankful to my guide **Mr. M.Manikantan, Senior Lecturer, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore** for his valuable guidance and support throughout the project.

I take this fine occasion to thank the software organization **“MEZOBLANCA SOLUTIONS India (Pvt) Ltd, Coimbatore”** for giving me a project. I also extend my thanks to external guide **Mr. M.Arun Kumar** for his valuable guidance.

At last I express my heartfelt thanks to my parents, my brother and my friends for the help rendered by them.

O.SASIKALA

CONTENTS

TITLE	PAGE NO
Abstract	iv
List of Figures	viii
List of Tables	ix
1. Introduction	
1.1 Organization Profile	1
1.2 Abstract	3
2. System Analysis	
2.1 Existing System	4
2.2 Proposed System	5
2.3 User Interface Requirements	6
3. Development Environment	
3.1 H/W Environment	10
3.2 S/W Environment	10
3.3 Software Overview	11
4. System Design	
4.1 Data Model	14
4.1.1 Table Relationship	14
4.2 Process Model	15
4.2.1 Use Case Diagram	15
4.2.2 Sequence Diagram	18
4.2.3 Activity Diagram	21
4.3 Database Design	22

5.	Testing	
	5.1 System Testing	24
	5.2 Unit Testing	25
	5.3 Integration Testing	25
	5.4 Validation Testing	26
6.	Performance and Limitations	
	6.1 Conclusion	27
	6.2 Limitations of Proposed System	27
	6.3 Future Enhancements	28
7.	Appendices	
	7.1 Sample Screens	29
8.	References	41

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1.1	Table Relationship	14
4.2.1.1	Use-Case: Main diagram	15
4.2.1.2	Use-Case: Mining rules	16
4.2.1.3	Use-Case: Selecting modification scheme	16
4.2.1.4	Use-Case: Indexing	17
4.2.2.1	Sequence Diagram: Mining rules	18
4.2.2.2	Sequence Diagram: selecting modification scheme	19
4.2.2.3	Sequence Diagram: Indexing	20
4.2.3	Activity Diagram	21

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
4.3.1	Main	22
4.3.2	Items	22
4.3.3	Bit vector	22
4.3.4	Primenos	23
4.3.5	Encoding	23
4.3.6	Converted	23
4.3.7	Transactions	23

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 ORGANIZATION PROFILE

The idea of providing quality software products at a competitive price gave birth to MEZOBLANCA SOLUTIONS India (Pvt) Ltd. MEZOBLANCA works as a part of any organization seeking cost-effective solutions. MEZOBLANCA is blessed with a team of software professionals in order to achieve its mission. MEZOBLANCANS work towards managing successful software delivery.

MEZOBLANCA SOLUTIONS India (Pvt) Ltd is a leading solution provider for software based applications established in 2003. The Company has been promoted by some highly experienced professionals dedicated to provide total IT solutions under one roof.

Mezoblanca team provides top quality software solutions with an innovative view. Our highly spirited Mezoblancans focus on meeting the complications of the present business with prodigious ease.

Mission

To help individuals enhance their creativity and build Knowledge that creates opportunities, thus deliver a result that maximizes customer's return on their existing investments, Affordable cost, High quality products, Effective marketing time.

Vision

To evolve as a product company by optimally using the resource thus to provide competitive products that are affordable.

Products

- CRM
- Trading, Accounts, Production process softwares
- Networking Products
- Medical Imaging Solutions
- Web Based Training Tools
- Project for Hospitality Industry, Gear Industry, Bearing Industry, Textile Industry

Services

- Software Development
- Web Designing
- Software Training
- Software development consulting
- Research and Development

Company Background

Business Since: 2003

No. of Employees: 1-15

Entity Type: Corporation

Contact Information

Company Address

204,PSG STEP Software ParkII,

Peelamedu, Coimbatore-4, Tamilnadu, India.

Company Phone: +914224363316

Sales Phone: +914224363318

Company Web Site: <http://www.mezoblanca.com>

Company email: info@mezoblanca.com

1.2 ABSTRACT

This project is aimed at improving the privacy preserving features of the DataMining results sharing and we utilize approach that strategically modifies a few transactions in the transaction database to decrease the supports or confidences of sensitive rules without producing the side effects. Since the correlation among rules can make it impossible to achieve this goal.

All the sensitive rules are hidden without spurious rules falsely generated. Moreover, the good scalability of our approach in terms of database size and the influence of the correlation among rules on rule hiding are observed.

SYSTEM ANALYSIS

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

There are frequent instances that we need to share our transaction database with 3rd parties for varieties of reason like to study the consumer's behavior, to have better layout of the products in the sales shelves/counter, accounting and product promotion. So, there is a growing demand for these types of transaction databases for Super stores, large trading corporations, Banks and other financials services and trading organizations.

The existing methodology of sharing the database after hiding the sensitive rules is widely followed. Very simple and unconventional methods are followed by the IT managers to hide the few sensitive data or data patterns before it is available for the sharing.

Though many organizations follow more robust and proven algorithms like Apriori, FP (Frequent Pattern) algorithms, there is no provisions or checks that are followed to ensure no undesired side effects are raised in the process of hiding the sensitive rules.

Some of the undesired Side Effects may be 1) Weak Rules (Rules which are not sensitive to the organization) may be falsely hidden. 2) Spurious Rules that may have been raised etc.

2.1.1 Disadvantages of the existing system

- The sampling approach used in the existing system suffers from the following drawbacks
 - An effective approach is not used

- Constraints are not checked
- Leads to side effects
- A method that affects the least number of transactions and helps to hide the most number of sensitive rule is not given the priority.
- Lacks in performance , particularly in a large database
- Uncertainty would be large if correlation among rules are not considered.

2.2 PROPOSED SYSTEM

In order to safeguard the genuineness of the transaction database that are to be studied and analyzed , we need to make sure the undesired side effect – 1) Weak Rules (Rules which are not sensitive to the organization) may be falsely hidden. 2) Spurious Rules that may have been raised etc., does not happen.

In order to avoid the side effects of the association rule hiding, we follow the constraint checking process before we confirm any hiding of sensitive association rule. It can be interesting to discover the full set of rules that will be falsely hidden or generated as the side effects after rule hiding.

Efficient mechanisms are required to speed up the rule hiding process for large databases. Another issue is the fast recognition of sensitive rules that cannot be hidden according to the user-specified constraint. An ideal goal is to build a system that can aid the database administrator to find the sensitive rules for hiding.

The other issue is to remove the threshold assumption. A rule hiding approach should be robust no matter how the adversary looks into the modified database, e.g., using a lower MCT to reveal the hidden sensitive rules. The challenge is to take into account both the above attacks and the undesired side effects.

2.2.1 Advantages of proposed system

The modifying approach used in the proposed system contains the following advantages

- An effective approach is used
- Constraints such as
 - Not to hide a strong rule
 - Not to generate another strong rule
 - fail to hide a strong rule
 are checked.
- Undesired side effects such as
 - Weak rules that may be falsely hidden
 - Rules that may be generated falsely
 are not created.
- Performance is good even in a large database

2.3 USER INTERFACE REQUIREMENTS

Modules are

- Mining of Strong Association Rules in Database.
- Evaluation of Modification Schemes for Rule Hiding.
- Index Construction and Template Generation
- Implementation of Properties of Sensitive Rule Hiding

1) **Mining of Strong Association Rules in Database:**

Association rule mining is a process of finding the set of strong association rules in a transaction database. An itemset is a set of elements recorded in a database recordset. The Support and Confidence of all the itemsets in the transaction database is calculated using Apriori Algorithm.

The Strong Rules are filtered out by using the predetermined Minimum Support Threshold and Minimum Confidence Threshold values. The Rules (set of itemsets) which have the Support and Confidence levels above the minimum Threshold levels are filtered out as Strong Rules by implementing the Apriori Algorithm. The Sensitive Rules among the Strong Association Rule are selected based on the Business interest and are separated aside to be targeted for Rule Hiding.

2) **Evaluation of Modification Schemes for Rule Hiding :**

The different modification schemes are evaluated for Efficient Rule hiding process. The schemes like Deletion, Insertion and swapping (both deletion and insertion together) are studied. The selection of the exact Modification scheme is done based on the factors

- 1) The modification scheme must affect minimum number of transaction itemsets to hide the rule
- 2) No Side Effects - Weak Rules Hiding and new spurious generation have to arise.

I

It can be seen that a modification schemes consists of three elements, including the modification scheme (either deletion or insertion), the item, and the transactions to be modified. The effects of a modification can be different if one of its elements is changed. It may increase or decrease both Support and Confidence.

A modification is said to be valid if it will not produce any side effect in the modified database. Since a number of valid modifications can be applied to hide a sensitive rule, the goal of this module is to select a proper subset of valid modifications to hide the sensitive rules. Specifically, the modification that affects the least number of transactions and helps to hide the most number of sensitive rules has priority over the others.

3) Index Construction and Template Generation:

We apply to the an algorithm for association rule mining with MST 20% and MCT 60% and obtain the frequent itemsets and the strong rules..We encode each item as a unique prime number such that all the transactions and rules are converted into products of prime numbers.

The frequent items are sorted by their counts and mapped to the prime numbers in reverse order. In this way, the product of prime numbers stored in the index and tables will not be too large. Not only the transactions but also the rules are represented in the form of prime numbers or products of prime numbers.

The strong rules are distributed into two tables. Since the number of sensitive rules is often much less than the number of non-sensitive rules, we do not keep the sensitive rules in the transaction-rule index. With this index, the transactions and non sensitive rules satisfying a logical form can be quickly identified. . The efficiency of template generation depends on both the numbers of sensitive rules and items in the individual sensitive rules. If there are common items among the sensitive rules, the costs in this stage can be reduced.

4) Implementation of Properties of Sensitive Rule Hiding Based on Constraints:

The Constraints to be considered while implementing of Properties of Sensitive Rule

- 1) The Rule which is Not to be Hidden denoted by N-T-H.
- 2) The Rule which is Not to be Generated denoted by N-T-G.

The constraints are carefully checked iteratively step by step and the properties of Sensitive Rule Hiding are selected from the listing which satisfies the above said Constraints.

In this module, we adopt Scheme 3 i.e. addition and deletion together to modify the database, and use the constraints N-T-H and N-T-G as the guides to avoid the side effects. To consider both the numbers of hidden sensitive rules and modified entries, we include the correlation among rules in our modification scheme.

DEVELOPMENT ENVIRONMENT

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 H/W ENVIRONMENT

Processor	:	Pentium IV
Speed	:	Above 500 MHz
RAM capacity	:	128 MB
Floppy disk drive	:	1.44 MB
Hard disk drive	:	20 GB
Key Board	:	Samsung 108 keys
Mouse	:	Logitech Optical Mouse
CD Writer	:	52x LG
Printer	:	DeskJet HP
Motherboard	:	Intel
Cabinet	:	ATX
Monitor	:	17" Samsung

3.2 S/W ENVIRONMENT

Operating System	:	Windows 2000 and above
Front end used	:	VB.net
Back end used	:	SQL Server 2000

3.3 SOFTWARE OVERVIEW



3.3.1 VB.NET

VB.NET is the new version of Visual Basic. Visual Studio.NET is an Integrated Development Environment that hosts VB.NET, C#, and C++.NET. Underlying all this is the .NET Framework and its core execution engine, the Common Language Runtime.

When we compile the code from any language that supports the .NET Framework, it compiles into something called MSIL, or Microsoft Intermediate Language. This MSIL file is binary, but it is not machine code instead, it is a format that is platform independent and can be placed on any machine running the .NET Framework. Within the .NET Framework is a compiler called the Just-In-Time, or JIT, compiler. It compiles the MSIL down to machine code specific to that hardware and operating system.

3.3.2 .Net Framework

The .NET Framework is a collection of services and classes. It exists as a layer between the applications we write and the underlying operating system. This is a powerful concept: The .NET Framework need not be a Windows-only solution. The .NET Framework could be moved to any operating system, meaning our .NET applications could be run on any operating system hosting the .NET Framework. This means that we could achieve true cross-platform capabilities simply by creating VB.NET applications, provided the .NET Framework was available for other platforms. Although this promise of cross-platform capability is a strong selling point to .NET, there has not yet been any official announcement about .NET being moved to other operating systems.

The Common Language Runtime

One of the major components of the .NET Framework is the Common Language Runtime, or CLR. The CLR provides a number of benefits to the developer, such as exception handling, security, debugging, and versioning and these benefits are available to

any language built for the CLR. This means that the CLR can host a variety of languages, and can offer a common set of tools across those languages. Microsoft has made VB, C++, and C# "premier" languages for the CLR, which means that these three languages fully support the CLR. In addition, other vendors have signed up to provide implementations of other languages, such as Perl, Python, and even COBOL.

Microsoft Intermediate Language (MSIL)

One of the more interesting aspects of .NET is that when we compile our code, we do not compile to native code. Before we VB developers panic and fear that we are returning to the days of interpreted code, realize that the compilation process translates our code into something called Microsoft intermediate language, which is also called MSIL or just IL. The compiler also creates the necessary metadata and compiles it into the component. This IL is CPU independent. After the IL and metadata are in a file, this compiled file is called the PE, which stands for either *portable executable* or *physical executable*, depending on whom we ask. Because the PE contains our IL and metadata, it is therefore self-describing, eliminating the need for a type library or interfaces specified with the Interface Definition Language (IDL).

The Just-In-Time Compiler

Our code does not stay IL for long, however. It is the PE file, containing the IL that can be distributed and placed with the CLR running on the .NET Framework on any operating system for which the .NET framework exists, because the IL is platform independent. When we run the IL, however, it is compiled to native code for that platform. Therefore, we are still running native code; we are not going back to the days of interpreted code at all. The compilation to native code occurs via another tool of the .NET Framework: the Just-In-Time (JIT) compiler. With the code compiled, it can run within the Framework and take advantage of low level features such as memory management and security.

The .NET Framework Class Library

The .NET Framework provides a number of types that are already created and ready for use in any language that targets the Common Language Runtime. These types include such items as the primitive data types, I/O functions, data access, and .NET Framework security. Perhaps one of the biggest changes in the way developers will work with VB.NET is the entire area of namespaces. .NET Framework provides a host of utility classes and members, organized within a hierarchy called a *namespace*. At the root of the hierarchy is the System namespace. A namespace groups classes and members into logical nodes. This way, you can have the same name for a method in more than one namespace. The Left() method could therefore exist in the System.Windows.Forms namespace and the Microsoft.VisualBasic namespace.

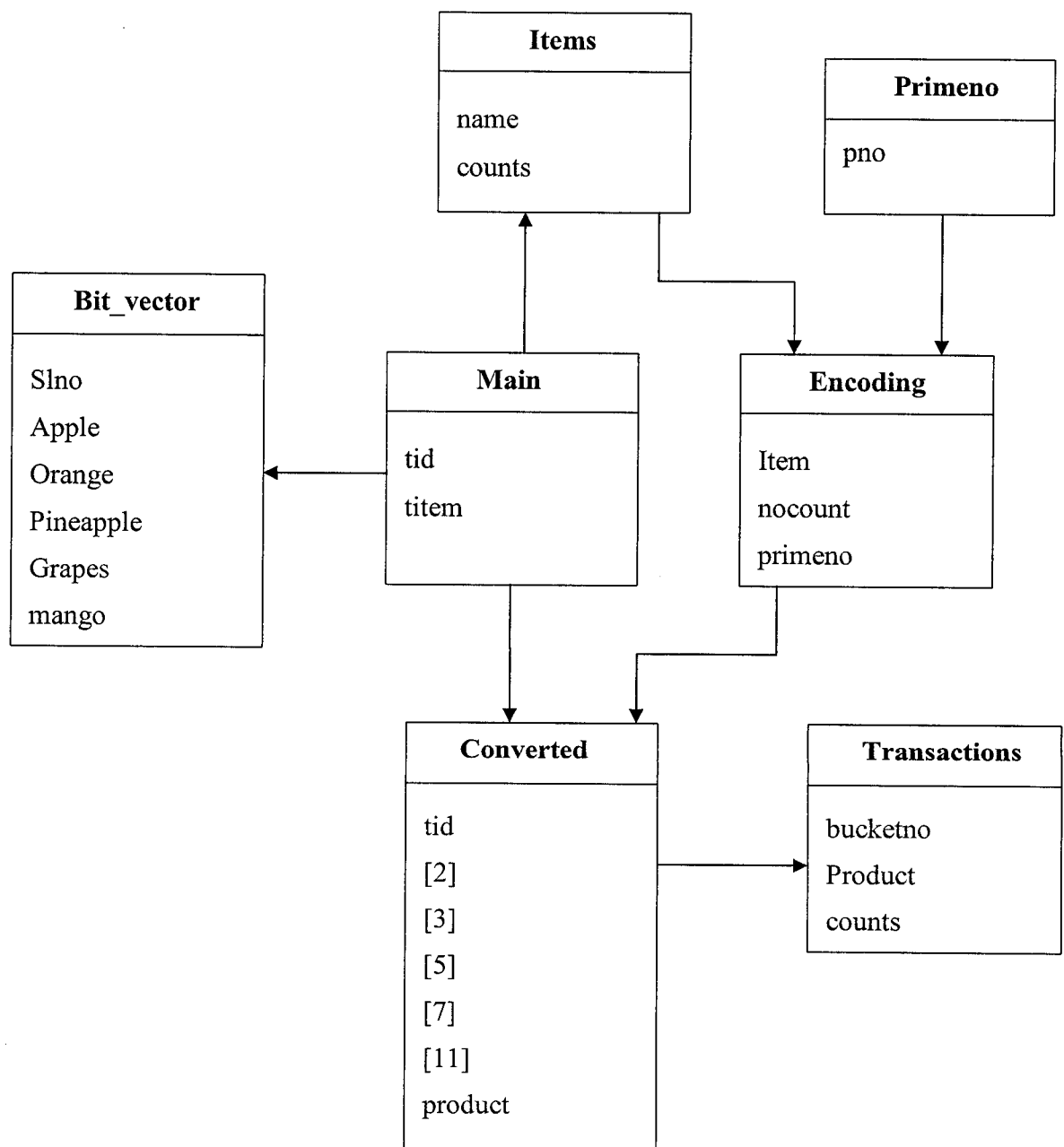
SYSTEM DESIGN

CHAPTER 4

SYSTEM DESIGN

4.1 DATA MODEL

4.1.1 Table Relationship

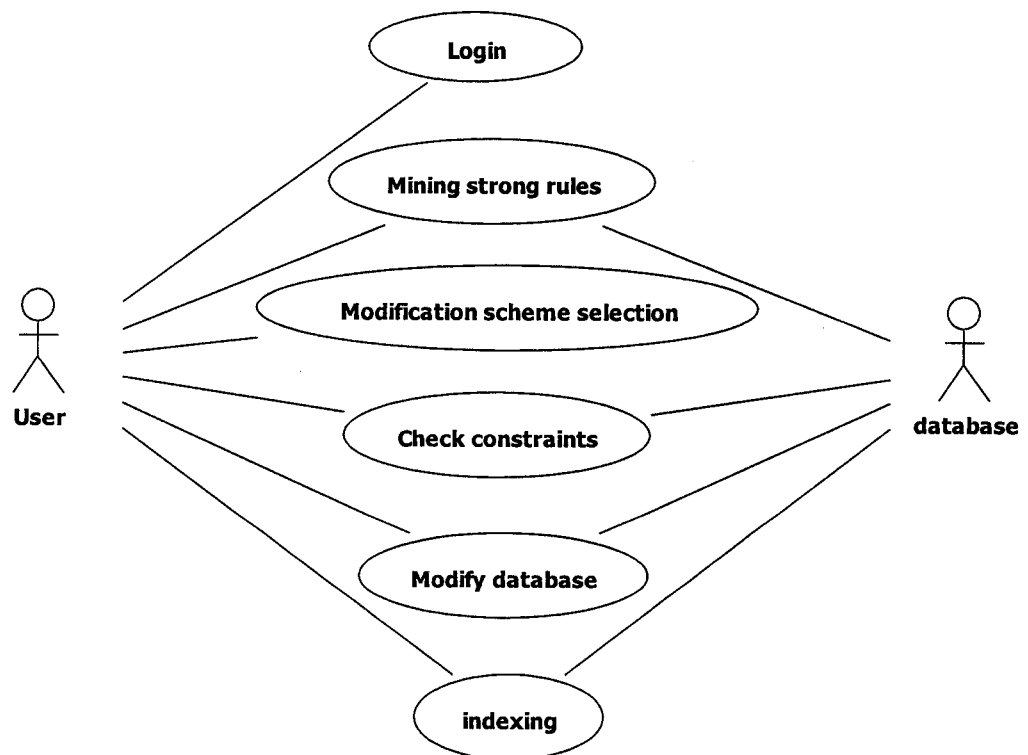


4.2 PROCESS MODEL

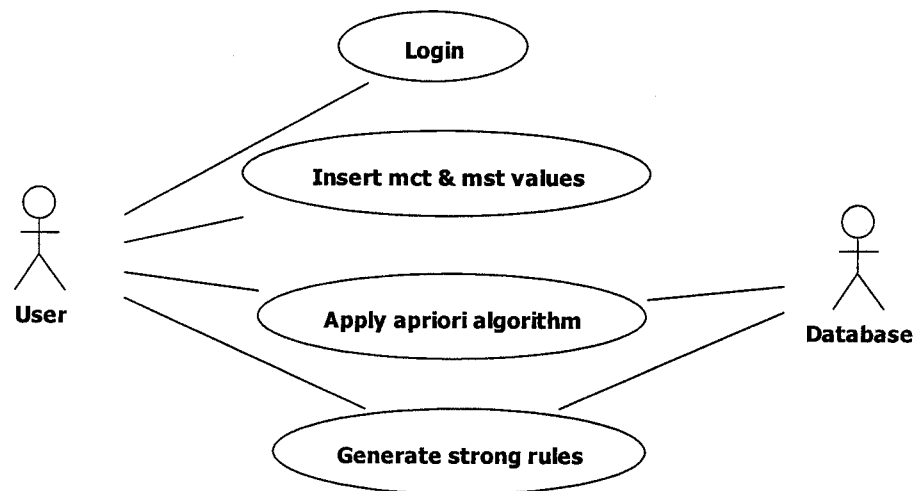
4.2.1 Use Case Diagram

A use case diagram is a type of behavioral diagram defined by the Unified Modeling Language. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals represented as use cases and any dependencies between those use cases.

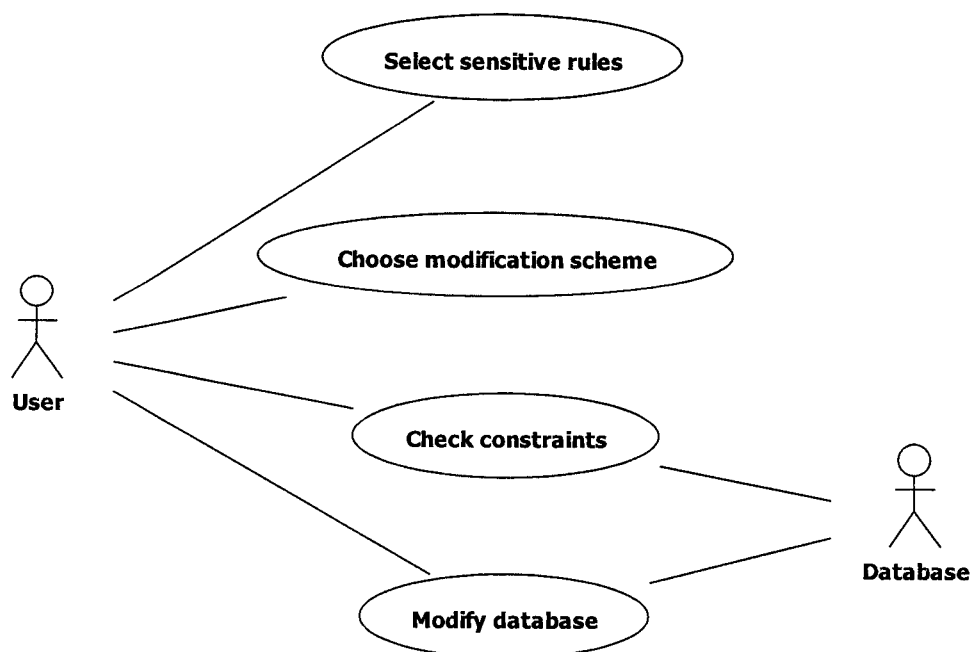
4.2.1.1 Main Use Case Diagram



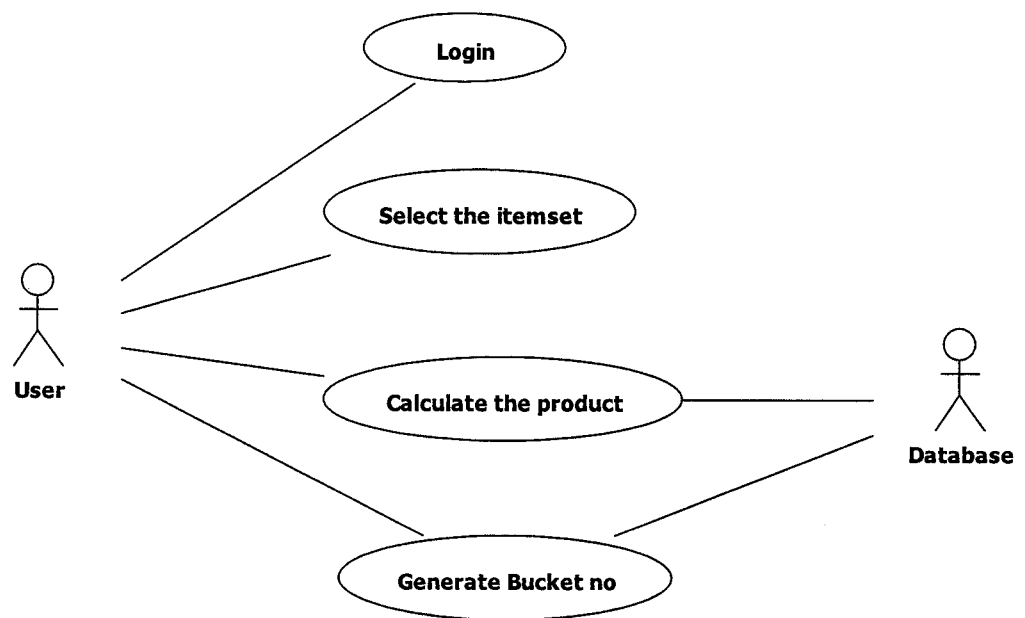
4.2.1.2 Mining Rules Uusecase diagram



4.2.1.3 Selecting Modification Scheme Usecase



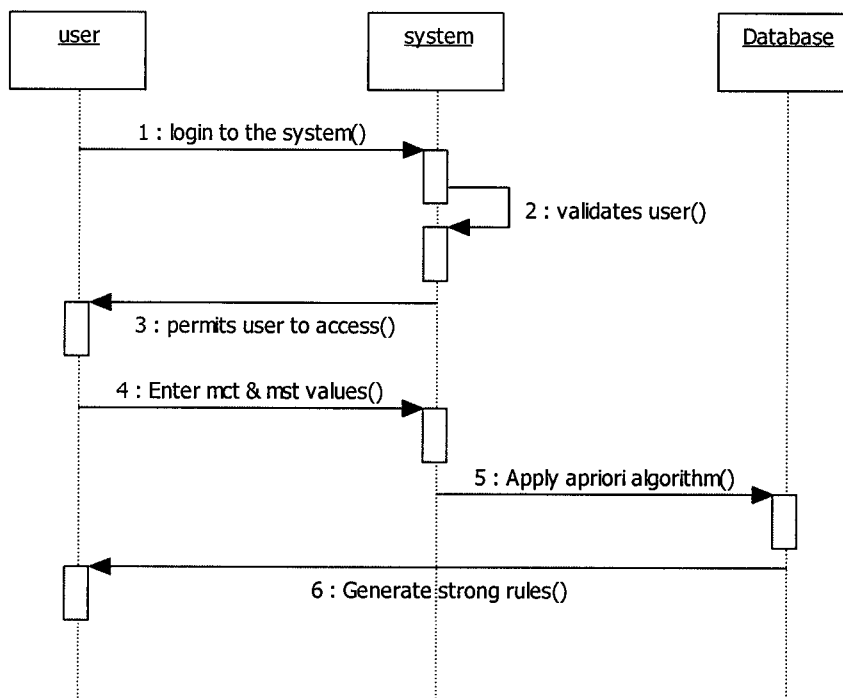
4.2.1.4 Indexing Usecase Diagram



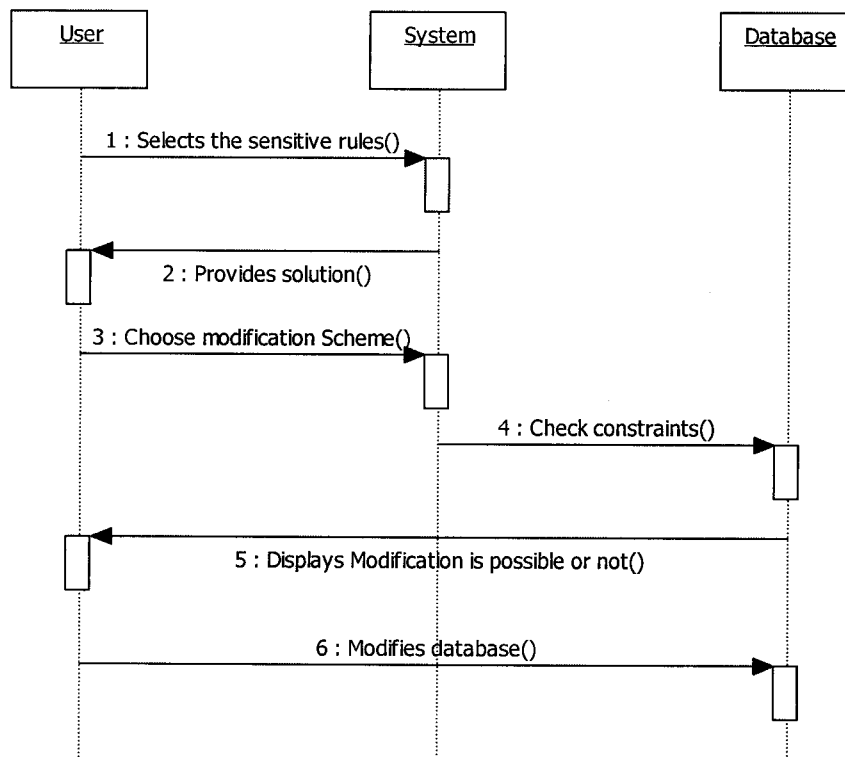
4.2.2 Sequence Diagram

A sequence diagram depicts the sequence of actions that occur in a system. It represents the dynamic behavior of the system. It is two dimensional in nature. On the horizontal axis it shows the life of the object while on the vertical axis, it shows the creation or invocation of the objects. A sequence diagram is made up of objects and messages.

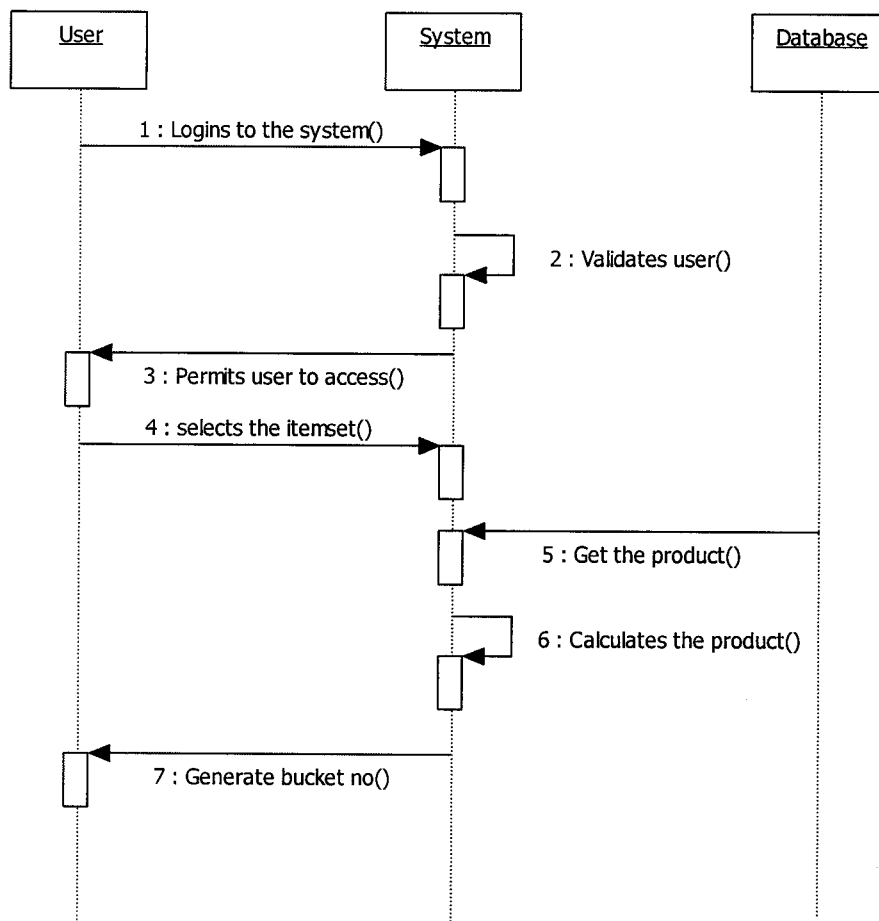
4.2.1.2 Mining Rules Sequence diagram



4.2.1.3 Selecting Modification Scheme Sequence Diagram

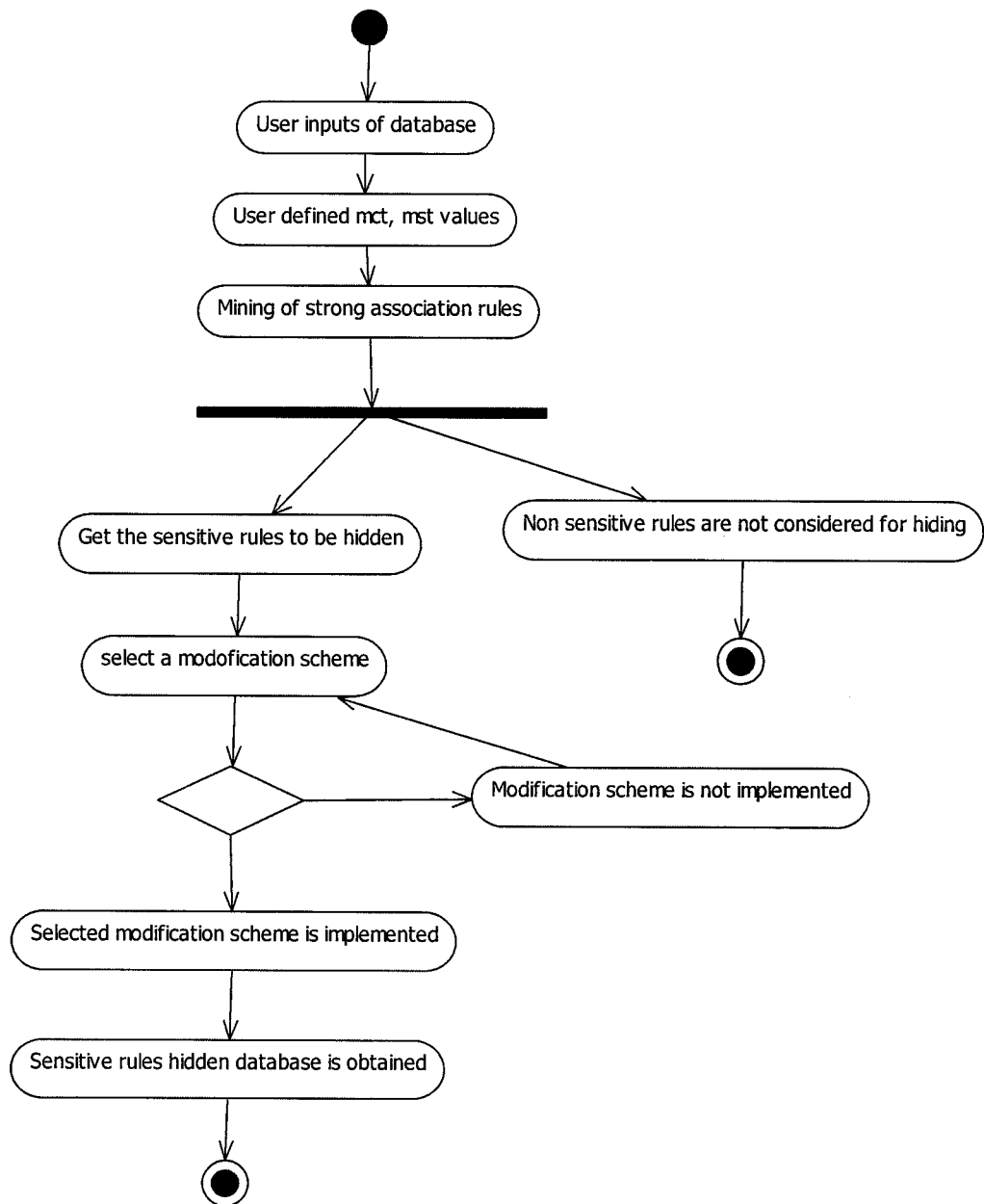


4.2.1.4 Indexing Sequence Diagram



4.2.3 Activity Diagram

Activity diagram provides a way to model the workflow of a business process. Activity diagram are similar to flow chart and it is basically a special case of a state machine.



4.3 DATABASE DESIGN

The system database contains the following table

NAME: main

PURPOSE: stores the transaction details

Table 4.3.1

FIELD NAME	DATA TYPE	DESCRIPTION
Tid	Int	Transaction
Titem	varchar	Transaction items

NAME: items

PURPOSE: stores the items and their counts

Table 4.3.2

FIELD NAME	DATA TYPE	DESCRIPTION
Name	Varchar	Name of the item
counts	Int	No. of counts of item

NAME: bit_vector

PURPOSE: converts the items into 1's & 0's

Table 4.3.3

FIELD NAME	DATA TYPE	DESCRIPTION
Slno	Int	Serial no.
Apple	Bit	Item
Orange	Bit	Item
Pineapple	Bit	Item
Grapes	Bit	Item
Mango	Bit	Item

NAME: primeno

PURPOSE: stores the prime number

Table 4.3.4

FIELD NAME	DATA TYPE	DESCRIPTION
Pno	Int	Prime number

NAME: encoding

PURPOSE: encodes the item with prime no.

Table 4.3.5

FIELD NAME	DATA TYPE	DESCRIPTION
Item	Varchar	Name of the item
noount	Int	No. of counts of item
Primeno	Int	Prime no. for the item

NAME: converted

PURPOSE: converts the transaction into products

Table 4.3.6

FIELD NAME	DATA TYPE	DESCRIPTION
Tid	Int	Transaction id
[2]	Int	Prime no.
[3]	Int	Prime no.
[5]	Int	Prime no.
[7]	Int	Prime no.
Product	Int	Product of prime no.

NAME: transactions

PURPOSE: stores the product counts of transaction

Table 4.3.7

FIELD NAME	DATA TYPE	DESCRIPTION
Bucketno	Int	Bucket number
Product	Int	Product of prime no.
counts	Int	No. of counts of product

TESTING

CHAPTER 5

TESTING

5.1 SYSTEM TESTING

It is the stage of implementation, which ensures that system works accurately and effectively before the live operation Commences. It is a confirmation that all are correct and opportunity to show the users that the system must be tested with text data and show that the system will operate successfully and produce expected results under expected conditions.

Before implementation, the proposed system must be tested with raw data to ensure that the modules of the system work correctly and satisfactorily. The system must be tested with valid data to achieve its objective.

The purpose of system testing is to identify and correct errors in the candidate system. As important as this phase is, it is one that is frequently compromised. Typically, the project the schedule or the user is eager to go directly to conversion. Actually, testing is done to achieve the system goal. Testing is vital to the parts of the system are correct; the goal will be successfully achieved.

Inadequate testing or non-testing leads to errors that may not appear until months later. This creates two problems:

- The time lag between the cause and appearance of the problem.
- The effect of system errors on files and records within the system. A small system error can conceivably exploded into much larger problem. Effectively early in the process translates directly into long term cost savings from a reduced number of errors.

5.2 UNIT TESTING

A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. Achieving an error free program is the responsibility of the programmer. Program testing checks for two types of errors: syntax and logical. Syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or error messages generated by the computer. Logic error deals with incorrect data fields, out-of range items, and invalid combinations. Since diagnostics do not determine logic errors the programmer must examine the output carefully.

When a program is tested, the actual output is compared with the expected output. When there is a discrepancy the sequence of instructions must be traced to determine the problem the process is facilitated by breaking the program down into self-contained portions, each of which can be checked at certain key points. The idea is to compare program values against desk-calculated values to isolate the problem.

Unit testing has been performed on all the form modules. The syntax and logical errors have been corrected then and there. All the syntax errors have been rectified during compilation. The output has been tested with the manual.

5.3 INTEGRATION TESTING

Programs are invariably related to one another and interact in the total system. Each program is tested to see whether it conforms to related programs in the systems. Each portion of the system is tested against the entire module with both the test data and the live data before the entire system is tested as a whole.

Integration testing is systematic techniques for conducting the program structure. While at the same time conducting tests to uncover errors associated with the interfacing.

The objectives are to take unit tested modules and to built a program that has been dedicated by design.

There are two types of Integration steps,

- Top down Integration
- Bottom up Integration

5.4 VALIDATION TESTING

The validation testing is performed for all the data in the system. The data are completely validated according to the companies requested and requirement.

Validation testing

In these testing, software is completely assembled as package, interfacing errors have been uncovered and correction testing begins after each one of the two possible conditions exists. They are,

The function or performance characteristic's is confirm specification and are accepted. A deviation from the specification is uncovered and efficiency list is created.

CONCLUSION

CHAPTER 6

PERFORMANCE AND LIMITATIONS

6.1 CONCLUSION

With the methods proposed in this system, we can modify the transactions in an order so that both the numbers of hidden sensitive rules and modified entries are considered. The results show that our approach is scalable in terms of database size.

Moreover, our approach to the avoidance of undesired side effects in rule hiding is effective in two well-designed experiments. In most cases, all the sensitive rules are hidden without false rules generated. In addition, it is observed that the common items and the overlapping degrees among sensitive rules have a great impact on the performance of rule hiding.

6.2 LIMITATION OF PROPOSED SYSTEM

- The procedure of hiding the sensitive rules will cause sometimes alteration of the actual essence of the project. These undesired side effects will alter the genuinity of the transaction database and will not benefit truly anyone who uses this database for their study
- The computational cost of the project will be high in case of large database which will hamper the progress of the data mining.
- The selection of the most optimized modification scheme will be tricky and time consuming in most cases.

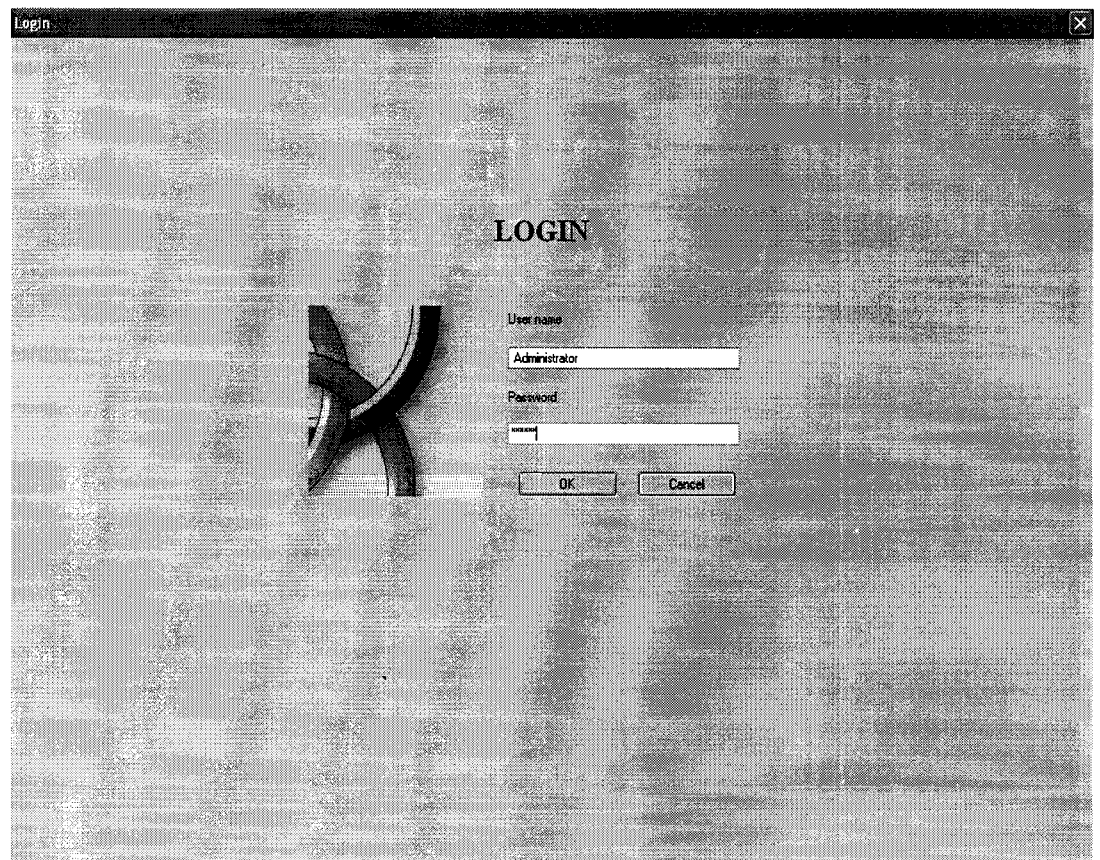
6.3 FUTURE ENHANCEMENTS

- The 100% elimination of the side effects for the hiding the sensitive rules have to be achieved.
- The selection of the most optimized modification scheme has to be more simple and straight forward.
- Efficient mechanisms are required to speed up the rule hiding process for large databases
- The number of itemsets in the database (n) has to be increased from the proposed system.
- The Template creation and indexing done for modification of the database for hiding the Sensitive Rules are made more efficiently and with less computational cost than the present proposed system.
- An ideal goal is to build a system that can aid the database administrator to find the sensitive rules for hiding.

APPENDICES

7.1 SAMPLE SCREENS

LOGIN:



GENERATING STRONG RULES:

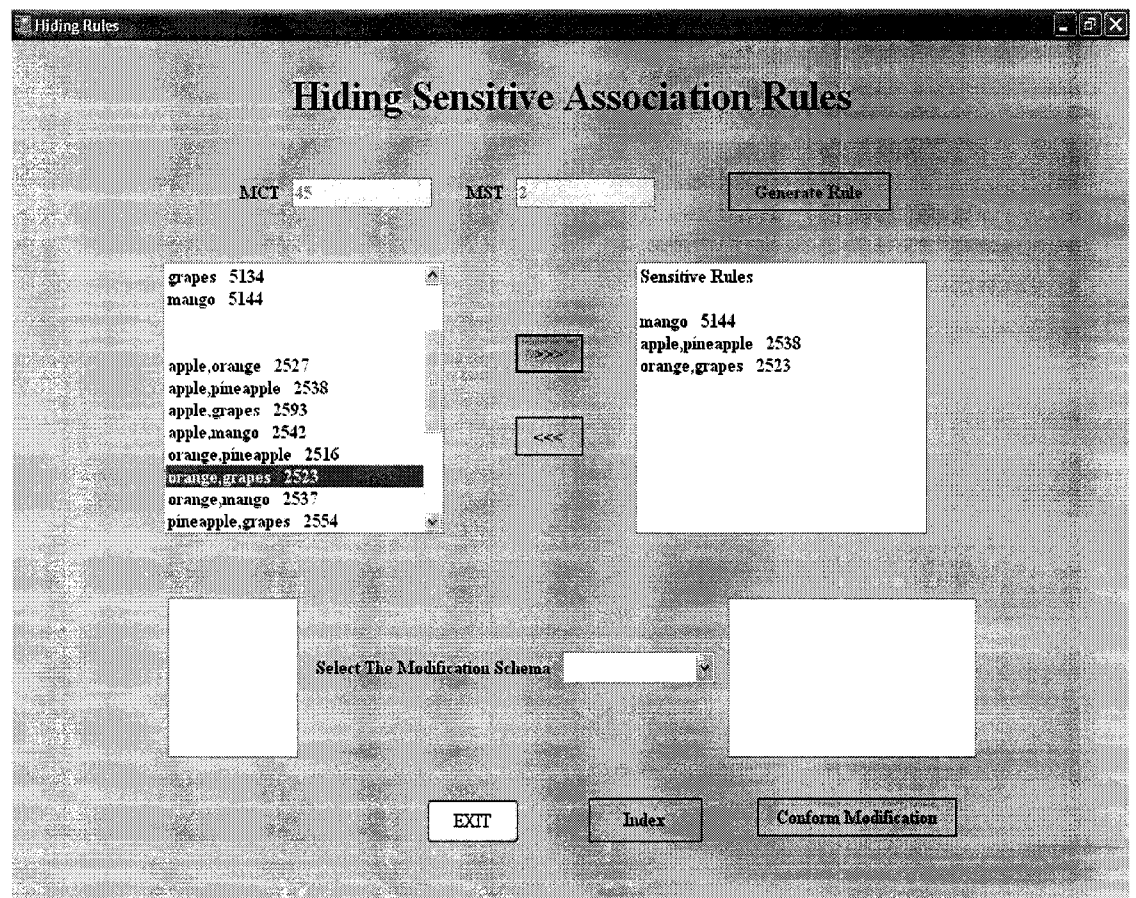
The screenshot shows a software window titled "Hiding Rules" with the main heading "Hiding Sensitive Association Rules". At the top, there are two input fields: "MCT" with the value "45" and "MST" with the value "2". To the right of these fields is a button labeled "Generate Rule".

Below the input fields, there is a list box on the left titled "Strong rules....." containing the following items:

- apple 5145
- orange 5085
- pineapple 5117
- grapes 5134
- maugo 5144
- apple,orange 2527
- apple,pineapple 2538
- apple,grapes 2593

Between the list box and a large empty rectangular area on the right are two buttons with double arrow symbols: ">>" and "<<".

At the bottom of the window, there is a label "Select The Modification Schema" followed by a dropdown menu. Below this are three buttons: "EXIT", "Index", and "Conform Modification".

SELECTING SENSITIVE RULES:

SOLUTIONS TO THE RULE:

Hiding Sensitive Association Rules

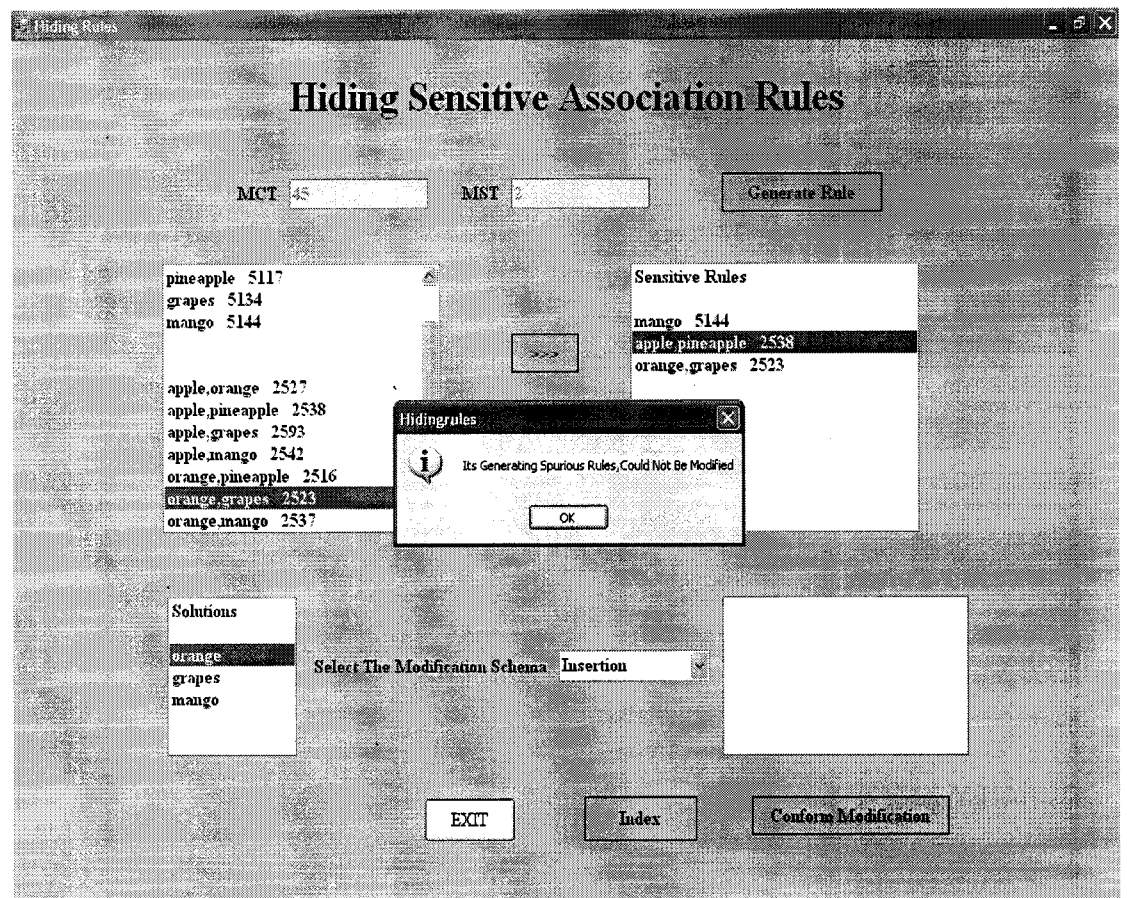
MCT: 45 MST: 2

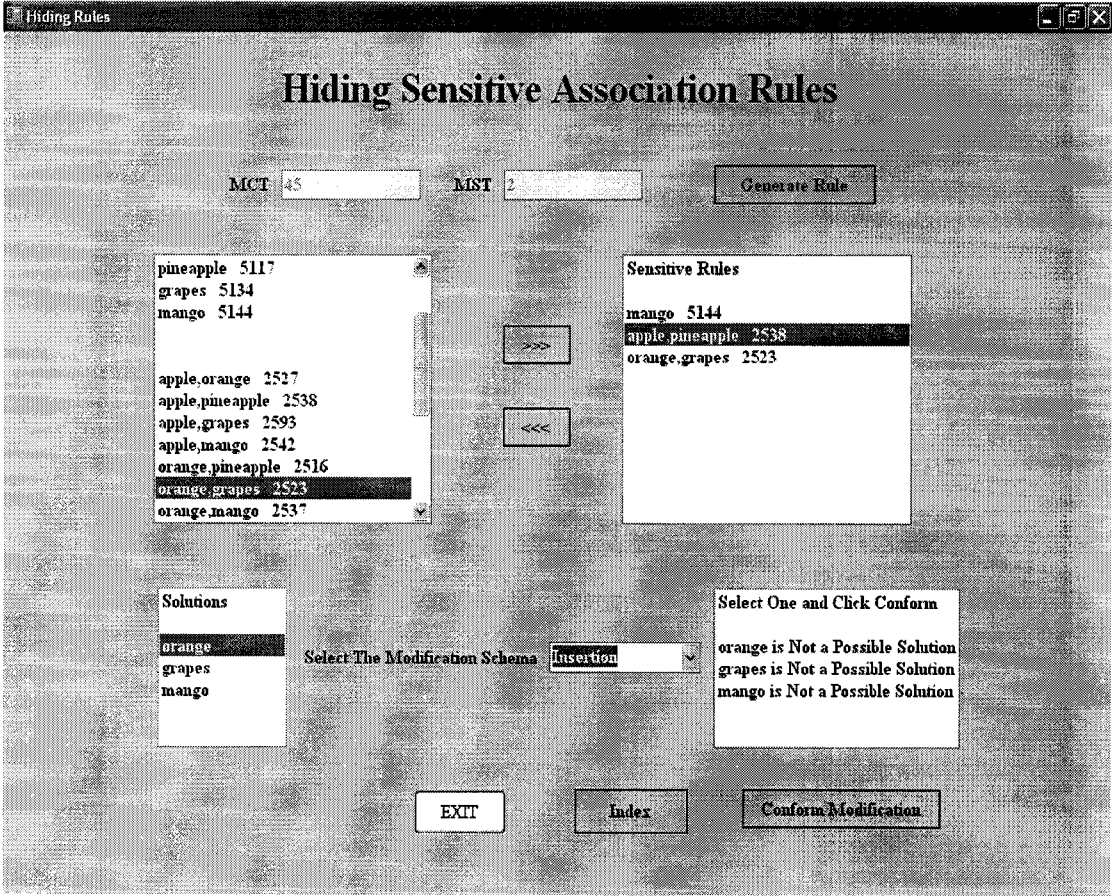
pineapple 5117
grapes 5134
mango 5144

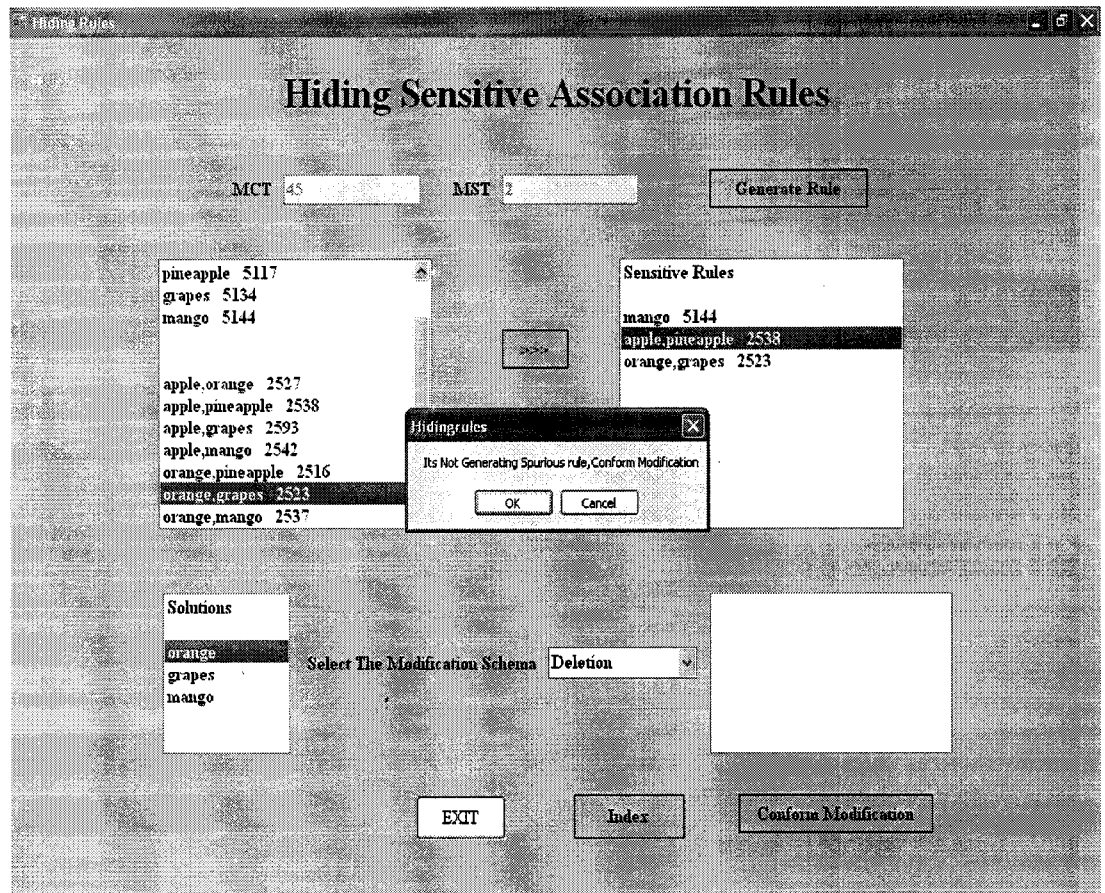
apple,orange 2527
apple,pineapple 2538
apple,grapes 2593
apple,mango 2542
orange,pineapple 2516
orange,grapes 2523
orange,mango 2537

>>>

MODIFICATION SCHEME SELECTION:







Hiding Sensitive Association Rules

MCT: 45 MST: 2 **Generate Rule**

pineapple 5117
grapes 5134
mango 5144

apple,orange 2527
apple,pineapple 2538
apple,grapes 2593
apple,mango 2542
orange,pineapple 2516
orange,grapes 2523
orange,mango 2537

>>>

<<<

Sensitive Rules

mango 5144
apple,pineapple 2538
orange,grapes 2523

Solutions

orange
grapes
mango

Select The Modification Schema: **Deletion**

Select One and Click Conform

orange 2538
grapes 2538
mango 2538

EXIT **Index** **Conform Modification**

MODIFYING THE DATABASE:

Hiding Sensitive Association Rules

MCT MST

Database

- pineapple 5117
- grapes 5134
- mango 5144
- apple,orange 2527
- apple,pineapple 2538
- apple,grapes 2593
- apple,mango 2542
- orange,pineapple 2516
- orange,grapes 2523**
- orange,mango 2537

Sensitive Rules

- mango 5144
- apple,pineapple 2538
- orange,grapes 2523**

>>>

Solutions

- orange**
- grapes
- mango

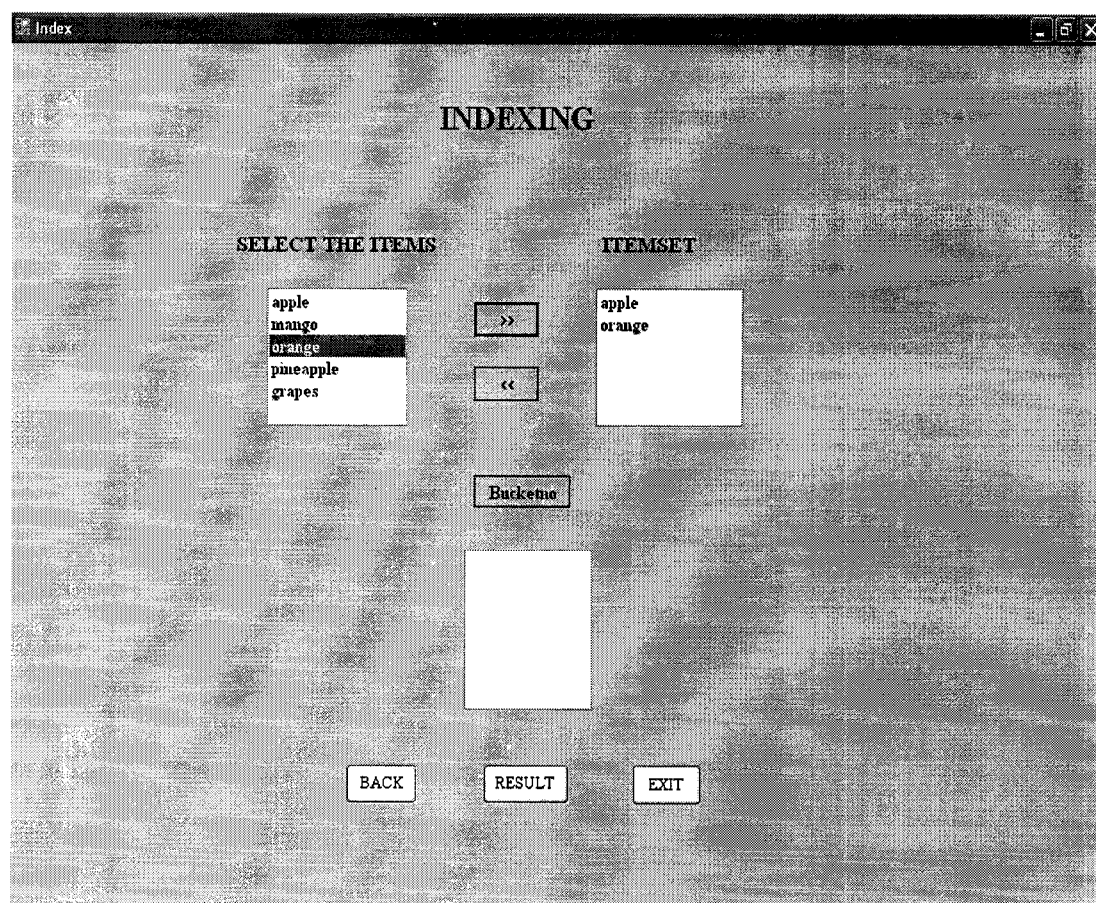
Select The Modification Schema:

Select One and Click Conform

- orange 2538
- grapes 2538**
- mango 2538

Hidingrules

Database has Been Modified Successfully

SELECTING THE ITEMSET FOR INDEXING:

GENERATING BUCKET NUMBER:

Index

INDEXING

SELECT THE ITEMS

- apple
- mango
- orange
- pineapple
- grapes

ITEMSET

- apple
- orange

>>

<<

Bucket no

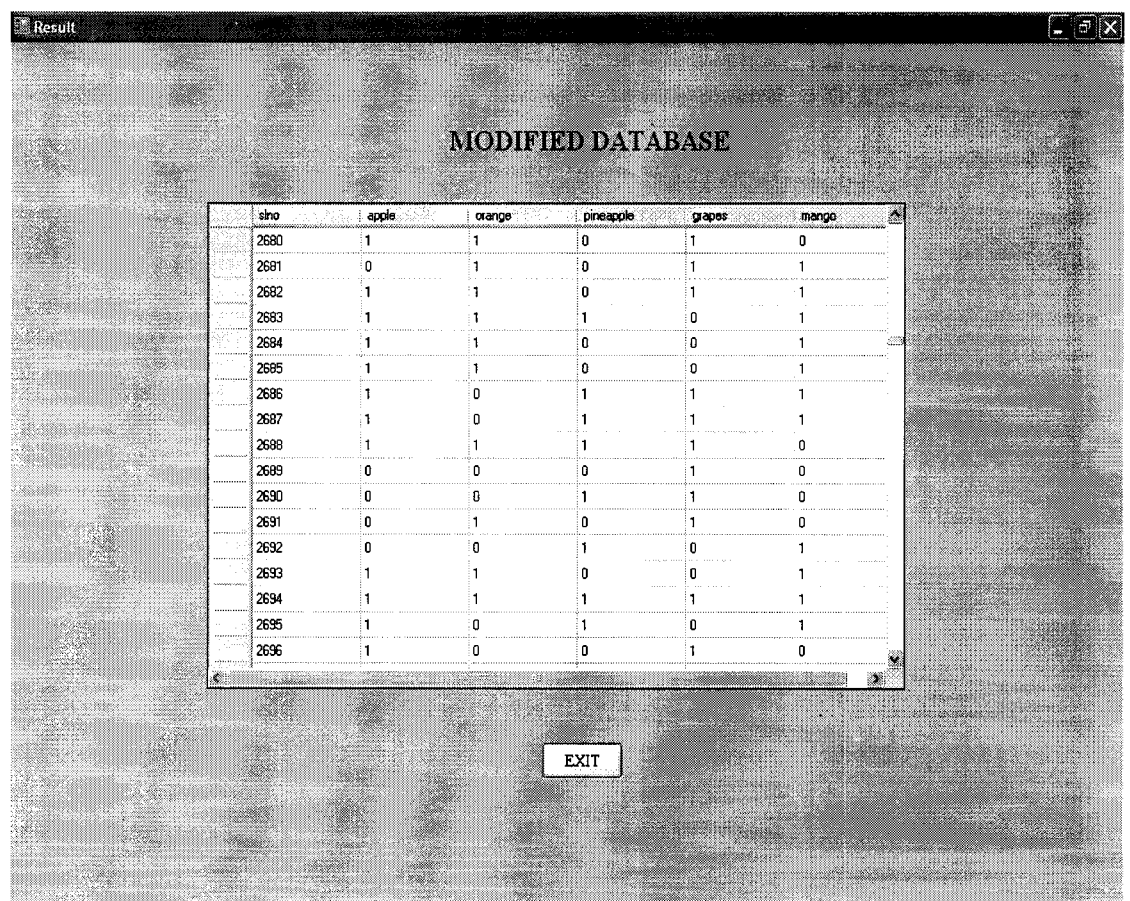
- 1
- 6
- 7
- 8
- 17
- 19
- 21

BACK

RESULT

EXIT

RESULT:



MODIFIED DATABASE

sno	apple	orange	pineapple	grapes	mango
2680	1	1	0	1	0
2681	0	1	0	1	1
2682	1	1	0	1	1
2683	1	1	1	0	1
2684	1	1	0	0	1
2685	1	1	0	0	1
2686	1	0	1	1	1
2687	1	0	1	1	1
2688	1	1	1	1	0
2689	0	0	0	1	0
2690	0	0	1	1	0
2691	0	1	0	1	0
2692	0	0	1	0	1
2693	1	1	0	0	1
2694	1	1	1	1	1
2695	1	0	1	0	1
2696	1	0	0	1	0

EXIT

REFERENCES

8. REFERENCES

- [1] D. Agrawal and C.C. Aggarwal(2001), "On the Design and Quantification of Privacy Preserving Data Mining Algorithms," Proc. ACM Symp. Principles of Database Systems, pp. 247-255.
- [2] R. Agrawal, T. Imielinski, and A. Swami(1993), "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM Conf. Management of Data, pp. 207-216.
- [3] R. Agrawal and R. Srikant(1994), "Fast Algorithms for Mining Association Rules," Proc. Conf. Very Large Data Bases, pp. 487-499.
- [4] Yi-Hung Wu, Chia-Ming Chiang, and Arbee L.P. Chen (2007), "Hiding Sensitive Association Rules with Limited Side Effects " IEEE Trans. Knowledge and Data Eng., vol. 19, no. 1, pp. 29-42.