



**Enhanced K\_way Method In "APRIORI" Algorithm for  
Mining the Association Rules Through Embedding SQL  
Commands**

**تحسين طريقه "K\_way" في خوارزميه "ابريوري" للتقيب عن القواعد**

**المرتبطه من خلال اوامر SQL**

**By**

**Basel Ali Dbwan**

**Supervisor**

**Dr. Hazim Farhan**

**Submitted in Partial Fulfillment of the requirements of the**

**Master Degree in Computer Information System**

**Faculty of Information Technology**

**Middle East University**

**July, 2013**

**Middle East University****Authorization Statement**

I Basel Ali Saleh Dbwan, authorize Middle East University to supply hardcopies and electronic copies of my thesis to libraries, establishments, or bodies and institutions concerned with research and scientific studies upon request, according to the university regulations.

**Name:** Basel Ali Saleh Dbwan

**Date:** 24/07/2013

**Signature:**



**Middle East University**  
**Examination Committee Decision**

This is to certify that the thesis entitled “Enhanced K\_way Method In  
 "APRIORI" Algorithm for Mining the Association Rules Through Embedding  
 SQL Commands”


was successfully defend and approved on July 24rd 2013.

**Examination Committee Member**

**Signature**

1- Prof. Reyadh S. Naoum  
 Professor  
 Dean of Faculty of Information Technology  
 Middle East University

Chairman



2- Dr. Jehad A. Al-Saadi  
 Associate Professor  
 Assistant Director for Academic Affairs  
 Arab Open University

Member



3- Dr. Hazim A. Farhan  
 Associate Professor  
 Department of Computer Science  
 Faculty of Information Technology  
 Alzaytoonah University of Jordan

Supervisor  
 and

Member



## Acknowledgments

“In the name of Allah the Most Gracious the Most Merciful”. My guidance can not come except from Allah, in Him I trust, to Him I repent, and to Him praise and thanks always go.

I offer my sincerest gratitude to my advisor Dr. Hazim Farhan for his valuable contributions, knowledge, encouragement and helpful advices. Also, I would like to express a very special thanks to Prof. Reyadh Naoum, for his vision which brought this work forward and for being there any time I knocked on his door. I wish both of them more and more success and giving.

I would like to express a very special thanks to Dr. Essa Abdullah Hezzam, for his valuable advices and helping.

I am highly indebted to my parents who taught me the right things, encouraged gave me the hope and unconditional love. I wish both of them happiness and good health. Thanks for my brothers, sisters, relatives and friends for supporting me.

Very special thanks belong to my wife and partner for life for being with me in happiness and sadness, giving me hope and strength and supporting me through this thesis. Also very special thanks go to my son for his patience and smiles. To all above persons, this thesis couldn't have been done without your support.

## **Dedication**

I dedicate this work to my father, my mother, my wife and partner for life, my son, my brothers and sisters; for their love, understanding and support, they were the light in my path. Without them nothing of this would have been possible. Thank you for everything. I love you!

## Table of Contents

Authorization Statement .....	I
Examination Committee Decision .....	II
Acknowledgements .....	III
Dedication .....	IV
List of Tables .....	VIII
List of Figures .....	IX
List of Abbreviations .....	XI
Abstract in English .....	XII
Abstract in Arabic .....	V

<b>Chapter One</b>	<b>Introduction.</b>	<b>1</b>
1.1	Introduction .....	2
1.2	Problem Statement .....	4
1.3	Objectives of the Study.....	4
1.4	Significance of the Study .....	5
1.5	Limitations of the Study .....	5
1.6	Thesis Organization.....	5
 <b>Chapter Two</b>	 <b>Data mining Overview</b>	 <b>7</b>
	<b>and Related Work.</b>	
2.1	Introduction .....	8
2.2	Data Mining Technology overview .....	8
2.3	Fundamental Components of Data Mining Technology .....	13
2.3.1	Applications .....	14
2.3.2	Operations .....	15
2.3.3	Classification and prediction .....	15
2.3.4	Association Rules .....	18
2.4	Association Rules Methods .....	18
2.5	Apriori Algorithm .....	21
2.6	Related Works .....	27

### **Chapter Three THE MODEL (ENHANCED APRIORI ALGORITHM) ARCHITECTURE, IMPLEMENTATION AND TESTING. 32**

3.1	Introduction .....	33
3.2	The Model Architecture .....	33
3.2.1	Model User Interface .....	34
3.2.1.1	The User Interface of the APRIORI algorithm....	34
3.2.1.1.1	Database conection .....	34
3.2.1.1.2	Run Apriori.....	35
3.2.1.1.3	Frequent Itemset .....	36
3.2.1.1.4	Generated Association Rule .....	37
3.2.2	Preprocessing .....	39
3.2.3	Enhanced k-way Method in Apriori Algorithm.....	39
3.2.4	Relational Engine.....	41
3.2.5	Postprocesses.....	42
3.3	Training and Testing The Proposed Model .....	42

### **Chapter Four Evaluation and Experimental Results. 45**

4.1	Introduction .....	46
4.2	Performance Evaluation .....	46
4.3	Experimental Results .....	46
4.4	Comparison with Other Studies Results .....	54
4.4.1	Running Apriori using SQL and using External text files....	54
4.4.2	Running the Apriori using indexes and without using indexes.	57
4.4.3	Running Apriori with and without greater than (>).....	59
	operator by different support values	

<b>Chapter Five</b>	<b>Conclusion and Recommendations.</b>	<b>60</b>
5.1	Introduction .....	61
5.2	Conclusion .....	61
5.3	Recommendations for Future Research .....	62
References.....		63



## List of Tables

Title	page
2.1 Notation for mining algorithm.	22
2.2 Sample transaction database.	25
3.1 Ttransaction table.	39
4.1 Three indexes on transaction table.	55

## List of Figures

Title	page
2.1 Rich data but poor information.	9
2.2 Searching for knowledge in your data.	10
2.3 Data mining as a step in process of knowledge discovery.	11
2.4 Data mining as a confluence of multiple disciplines.	12
2.5 Components of Data Mining.	13
2.6 Decision tree.	17
2.7 Classification process steps.	18
2.8 Market basket analysis.	19
2.9 Apriori algorithm.	23
2.10 Rule generation algorithm.	24
2.11 The process of finding frequent itemsets.	26
3.1Architecture of the model.	33
3.2 database connection.	34
3.3 SQL pseudo_code of the aPriori algorithm.	35
3.4 Run Apriori.	36
3.5 frequent Itemset	37
3.6 Generated Association rule	38
3.7 Original k-way method	40
3.8 SQL statement used by K-way method before enhanced	41
3.9 SQL statement used by K-way method after enhanced	41
3.10number of retrieval records and time before enhanced	43
3.11Number of retrieval records and time after enhanced	44
4.1 Execution time as a function of the support threshold	53

<b>Title</b>	<b>page</b>
4.2 Main memory usage of the C# version as a function of million randomly selected tuples from the Adr table with Support = 10	54
4.3 Comparisons of running Apriori with index and without index.	56
4.4 Comparisons of running Apriori with and without > operator.	57

## List of Abbreviations

Abbreviations	Description
SQL	Structured Query Language.
RDBMS	Relational Database Management System.
KDD	Knowledge Discovery in Databases.
$L_K$	Set of large (frequent) k-item set.
$C_K$	Set of candidate k-item set.
Minconf	Minimum-Confidence.
TID	Transaction-ID.
DMQL	Data Mining Query Language.
F	Frequent itemset.
C	Candidate itemset.
T	Transaction table.
VigiBaseTM	is a unique collection of international drug safety data.

## **Abstract**

No doubt, the notable and bursting growth in data and databases has produced an imperative necessity for new mechanism and devices that can rationally and spontaneously convert the handled data into helpful and valid information and knowledge. Data mining is such a style that evolves non axiomatic, tacit, formerly anonymous, and possibly beneficiary information from data in databases. In this thesis we implemented the most recognized algorithm “APRIORI” for mining association rules using SQL. Association rules that is linked to market basket data mining issue widely found in business application, which mostly used C# or Java and a text file that represent the transaction database. But at the mean time our application transact straightaway with transaction table in relational database(Warehouse) without any necessity to transmit the pertinent data to independent text file which facilitate the process of mining the database, also the most part in this Thesis, we achieved some enhancement to K-way method to calculate frequent itemsets. The thesis elucidated that using indexes may demean the performance of data mining process; moreover, the thesis implementation a (Graphical User Interface) for mining association rule using oracle 9i database tools.

**Keywords:** data mining; association rules; relational, database; Apriori ; SQL.

### الملخص

لا شك كان للنمو الهائل والملحوظ في البيانات وقواعد البيانات اثرا هاما في بروز حاجة ملحة لتقنيات وأدوات جديدة يمكن لها وبصورة تلقائية تحويل البيانات التي تم التعاطي معها الى معلومات ومعرفة مفيدة. ان عملية التنقيب عن البيانات هي بمثابة التقنية التي تعمل على استخراج البيانات غير البديهية والضمنية وتلك التي لم تكن معروفة في السابق ويمكن ان تكون مفيدة. تقوم هذه الرسالة على تطبيق الخوارزمية المعروفة وهي الاستدلال القبلي (APRIORI) للتنقيب عن القواعد المرتبطة من خلال استخدام لغة الاستعلامات الهيكلية Structure Query (Language). ان قواعد الارتباط على صلة بمسألة استخراج بيانات سلة السوق الموجودة على نطاق واسع في تطبيقات الأعمال التجارية. استخدمت معظم التطبيقات العديد من لغات البرمجة منها C # أو جافا والملف النصي التي تمثل قاعدة بيانات المعاملات. لكن التطبيق الذي قمنا بتنفيذه يتعاطى مباشرة مع جدول المعاملات في قاعدة البيانات العلائقية (المستودع) دون ان يكون هناك حاجة الى نقل البيانات ذات الصلة إلى ملف نصي منفصل والذي من شأنه ان يسهل من عملية التنقيب في قاعدة البيانات، وأيضا فان الجزء الأهم من هذه الرسالة الذي قمنا به وحققنا من خلاله بعض التحسين على K-way Method لحساب مجموعة العناصر (itemssets) المتكررة. لقد بينت الرسالة أن استخدام الفهارس قد يؤدي إلى تقليل أداء عملية التنقيب عن البيانات، كما قمنا أيضا ببناء واجهة المستخدم الرسومية (Graphical User Interface) للتنقيب عن القواعد المرتبطة باستخدام أوراكل i9 وأدوات قاعدة البيانات.

# **Chapter One**

## **Introduction**

## **Chapter one**

### **Introduction**

#### **1.1 Introduction.**

The quick growth in data and databases has created a pressing need for new tools and techniques that can quickly and efficiently process raw data into useable information. In the last years the development of information technology has motivated a parallel growing of facilities to store and manage database. The largest amount of stored data is more important for the demand of extracting the implicit information they contained to aid the decision-making in business, health care services, research...etc. Thus, to obtain useful knowledge from the data stored in large repositories (i.e.the "knowledge discovery"), which is recognized as a basic necessity in many area.

Since nineties of the last century, the research area named "data Mining " has become central topic in databases and Artificial Intelligence. The task of discovering association rules was introduced by Agrawal, Imielinske, and Swami in 1993(Agrawal, Imieliński and Swami 1993). In its original form, the task was defined for a special kind of data, often called basket data, where a tuple consists of a set of binary attributes called items. Each tuple corresponds to a customer transaction, where a given item has a value of true or false, depending on whether or not the corresponding customer bought the item in that transaction. This kind of data is usually collected through bar-code technology; the typical example is a supermarket scanner.

There are various algorithms have been proposed to discover the frequent item sets (Sarawagi, Thomas and Agrawal 1998), (Agrawal, Imieliński and Swami1993). The Apriori algorithm is one of the most popular algorithms in the mining of association rules in a centralized database, which will explained broadly later.



Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to scour large database in order to find novel and useful patterns that might otherwise remain unknown. They also provide capabilities to predict the outcome of a future observation (Tan P., Steinbach M. and Kumar V. 2006). Data mining is a method of extracting what is useable within a database and separating it out from what is unusable. Such methods are necessary because, as human being, we lack the capacity to sort and organize such large volumes of data.

One of the main and important topic of data mining is Association Rule Mining. Association rule mining finds interesting association or correlation relationships among a large set of data items. With massive amount of data continuously being collected and stored in databases, many industries are becoming interested in mining association rules from their databases. For example the discovery of interesting association relationships among huge amounts of business transaction records can help catalog design, cross-marketing, loss leader analysis, and other business decision making process (Al-hamami Alaa 2008).

An association rule is an expression  $X \Rightarrow Y$ , where  $X$  and  $Y$  are sets of items. The intuitive meaning of such a rule is that transactions of the database which contain  $X$  tend to contain  $Y$ . An example of such a rule might be that 70% of customers who purchase bread also purchase butter.

Mining association rules is a widely used in data mining technique. Association analysis is an unsupervised form of data mining that looks for links between records in a data set. Association analysis is referred to the most common application as 'market basket analysis'.

## 1.2 Problem Statement.

According to the definition of data mining which refers to extracting information from large amount of data. This information is hidden by nature and can not be extracted without special intelligent tools and domain of experts who can analyze this knowledge and introduce it to decision makers.

Most research in this area has built a mining association rule algorithm by different programming languages and represents it in a text file, This needs to relevant data to separate text file, which does not deal directly with database, which causes wasting time.

There are another important problem when mining association rule in transaction table in relational database that use SQL by association rule algorithm (Apriori) in K-way method to compute frequent item sets, its generate joining the item to it self, this redundant data is not logic and consumes more time and resources. So, we try to avoid redundant data to decrease retrieval time and storage space.

## 1.3 Objectives of the Study

The main objective of this Thesis is to improve the most common association rule algorithm (Apriori) in transaction table in relational database. As well as to achieve the following:

- Make enhancement in K-way method to compute frequent item sets that eliminate redundant data to save time.
- Implement the “APRIORI” algorithm to explore the associated rules directly with database, which increase performance of the process to mining in the database.
- Implement a User Interface that make mining association rule easy.

### **1.4 Significance of the Study.**

The significance of the study lies the most successful application of data mining is the “Market Basket” application. This study will be used to analyze transaction databases and look for patterns among existing customer transactions. These patterns are used to help make business decisions, such as, what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, and selecting the items required and associated together in a timely manner, As well as, to increase investment opportunities and imposes its presence in the competitive environments. Another major business use of data mining methods is the analysis and selection of stocks and other financial transactions.

### **1.5 Limitations of the Study.**

- If a large number of items in the database, it needs more time to recover the data because the comparison must pass each item.
- Adding new transaction to the database, needs to repeat all processes of data mining.
- The optimal performance of the system is a myth; because the rapid changes in the world of technology and business effected to the system, as well as, the operating environment.

### **1.6 Thesis Organization.**

This thesis consists of five chapters including this chapter. Chapter two reviews data mining technology, presents an overview of data mining operations, gives an overview of Association Rules method,as well as, an overview of Apriori Algorithm, and finally it summarizes the related work. Chapter three outlines research methodology used in this thesis. Also it presents the proposed model architecture and the software that has been implemented the enhancement system. Chapter four describes the dataset

used for experiments in this study, experiments environment and procedures and presents the evaluation measures and experimental results, finally a comparison with other studies results is made. Chapter five concludes the research and gives some future directions for future research.

## **Chapter Two**

### **Data mining Overview and Related Work**

## **Chapter Two**

### **Data mining Overview and Related Work**

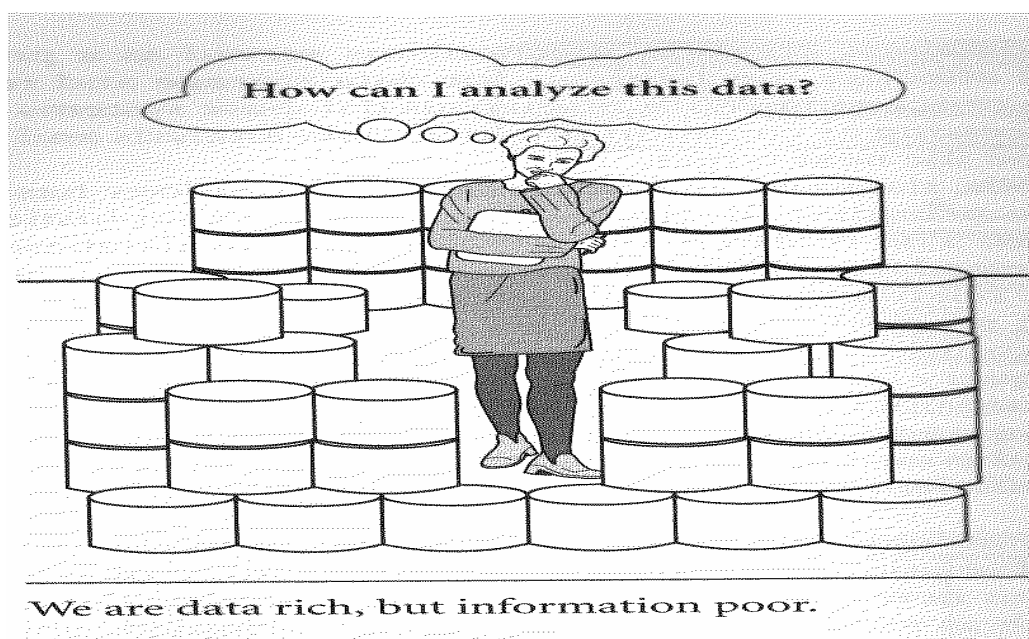
#### **2.1 Introduction**

This chapter consists of five sections. Section 2.2 data mining Technology overview , section 2.3 The Primary components of Data Mining Technology; section 2.4 discusses Association Rules method; section 2.5 overview of Apriori Algorithm, and section 2.6 summarize the works related to this thesis.

#### **2.2 Data Mining Technology overview.**

There is no doubt that the process of data mining has gained a great deal of attention of researchers whose views differed on the definition of the data mining process, but despite this difference, the definitions are generally the same. In this sense, we can say that the process of data mining suggest to extract information and knowledge through the large amount of data, which by their nature to be hidden and can not be extracted without intelligent tools and specialized experts who can analyze this information and submit it to the decision-makers. (Han J.and Kamber M. 2001). It also can be defined as extracting and discovery of interesting information, patterns or trends of a large database or data warehouse(Hansen, Hans R. and Neumann 2001). Data mining process has another meaning, where it can be defined as an operation of extracting important information, which is not already known from a large database. It has been notified that the last recent years have witnessed rapid growth in the capabilities that are concerned in generating and collecting data, such as, the noted develop in scientific data collection, the wide prevalence introduction of bar codes for almost all commercial products, and the automated of many businesses (e.g., credit card

purchases), it is worthy mentioning that those advanced capabilities have generated a large amount of data. The enhancement and great advancement in data storage technology, such as faster, higher capacity, and cheaper storage devices (e.g., magnetic disks or CD-ROMS); better database management systems; and data warehousing technology, have introduced us a great opportunity to transform this data into “mountains” of stored data.



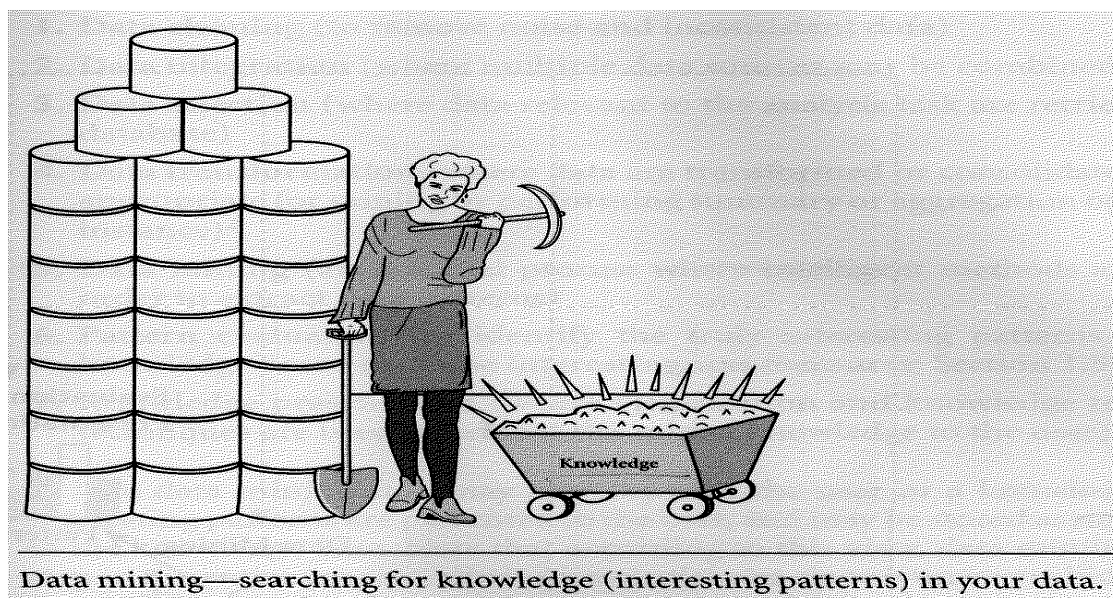
**Figure 2.1: Rich data but poor information (Han J.and Kamber M. 2001)**

The researchers and the scholars in the field of data mining have said that the large quantities of data that stored in large databases have far bypassed our ability for understanding and recognized it without powerful technique and appropriate tools.

The Figure 2.1 above shows how much a rich data is there, but at the same time the information was poor. As a result we ended up with data archives that are scarcely visited and inspected, from this point of view; the substantial decisions are made based

not on information-rich data stored in databases but primarily on a decision maker's perceptions.

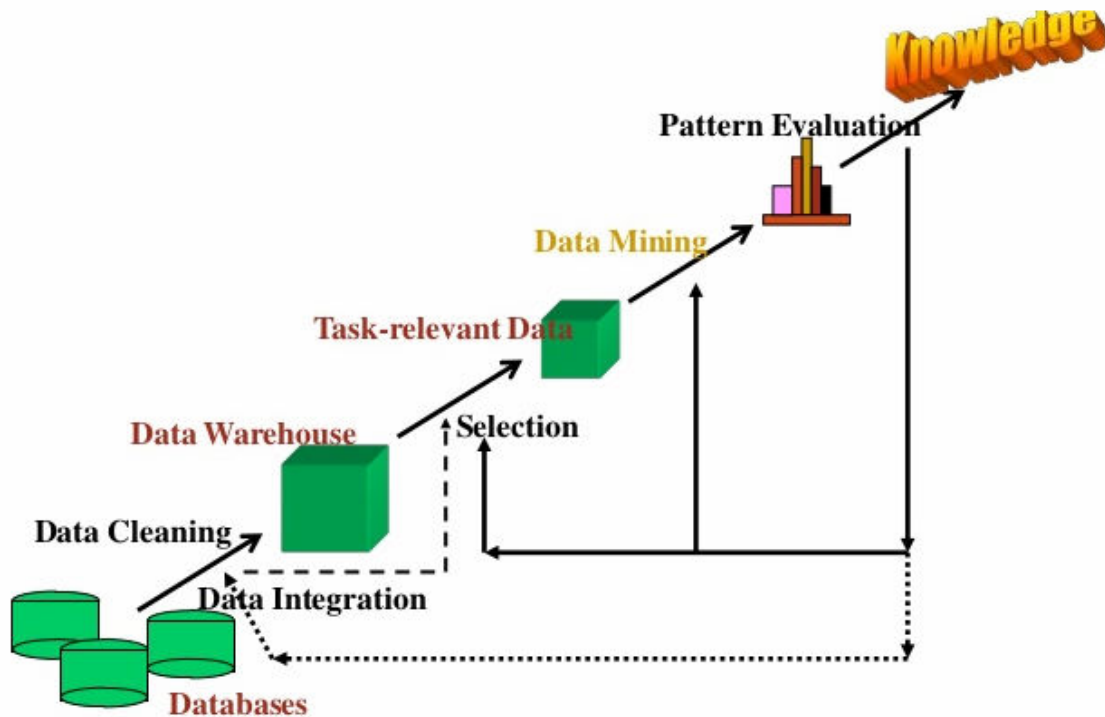
Thus, the enormous quantity of data steered to the futility of the old traditional methods to analyze data such as spreadsheets and ad hoc queries. A considerable data access and reporting tools (relational database management systems (RDBMSs), multidimensional analysis tools, ad hoc query and reporting software, and statistical analysis packages) allowed users query all these enormous data-stores. It is worthy to notify that these tools do not enable users to detect paradigms concealed in massive database or to accurately determine the factors they are seeking that will assist them produce faster, more precise decisions. It is not pragmatic to anticipate that human specialists closely analyze all this data. As being noted, a considerable need exists for contemporary and new generation of techniques and tools that can smartly, rationally and spontaneously convert the processed data into beneficial information and knowledge, as shown in Figure 2.2.



**Figure 2.2: Searching for knowledge in your data (Han J. and Kamber M. 2001)**



Some researchers and experts think data mining as knowledge discovery in databases (KDD), while others see it as a one step in the process of knowledge discovery.(Han J.and Kamber M. 2001).



**Figure 2.3: Data mining as a step in process of knowledge discovery (Han J.and Kamber M. 2001).**

As shown in figure 2.3, we conclude that knowledge discovery is a multi step processes that consists of 7 step which are

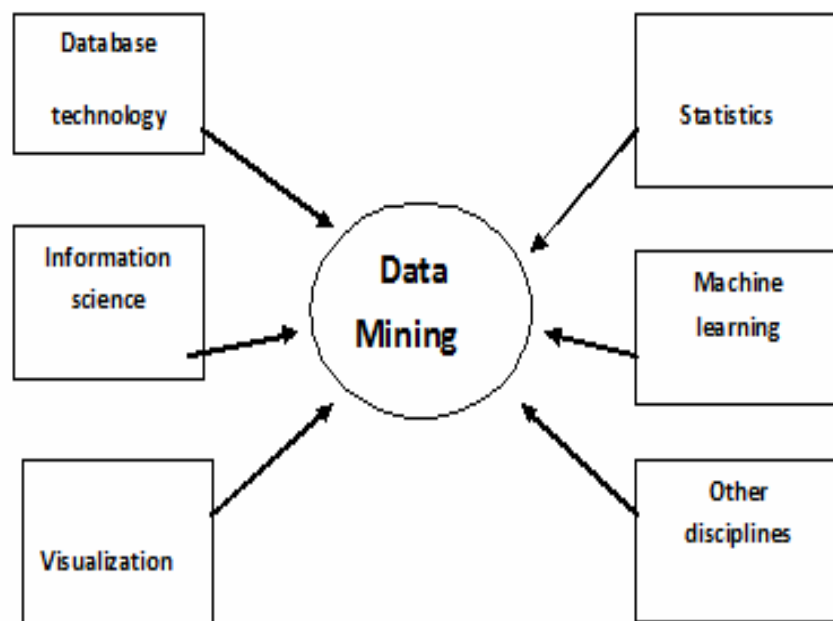
- **Data cleaning** removing noise from data.
- **Data integration** where multiple data sources may be combined.
- **Data selection** where relevant data is selected.
- **Data transformation** where aggregation and summaries are done in a way appropriate for mining.
- **Data mining** where intelligent tools and algorithms are used to mine the valuable knowledge from the data.

- **Pattern or rules evaluation** where truly important and interesting patterns are identified.

- **Knowledge presentation** where GUI is used to present the mined knowledge.

The overall and common idea of discovering “knowledge” in immense amounts of data is both appealing and critical, but technically it is significantly challenging, complicated and difficult. For example, the discovered information should not be apparent or obvious; where the information extracted should be easier than the data itself; implying that there should be a conclusive and high standard language for expressing such information; the information should be enjoyable and motivating.

Data mining can be described as an inter-disciplinary subject composed by the intersection of many diverse domains. It is really noticeable these days that researchers in knowledge base systems, artificial intelligence, machine learning, knowledge acquisition, and statistics have also shown great interest in data mining and paid more attention to this vital field, As shown below.



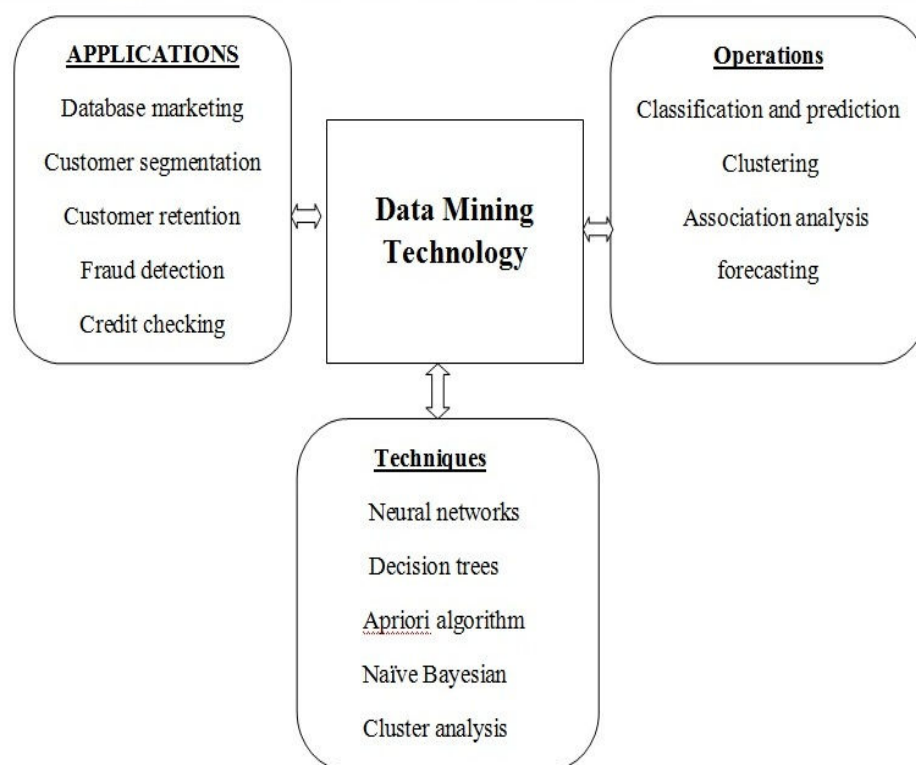
**Figure 2.4: Data mining as a confluence of multiple disciplines(Tan P., Steinbach M. and Kumar V. 2006).**

There is pivotal need to carry out devoted studies to find out new data mining modes or to promote incorporated techniques for dynamic and functional data mining. In this context, data mining itself has created distinct and an independent new field.

According to the field of business world, the most successful application of data mining is the “Market Basket” application. It is used to examine and analyze transaction databases and explore models among existing customer transactions. Those models are used to assist making business decisions, such as what to put on sale, how to layout coupons, how to put merchandise on shelves in order to maximize the profit, and so on. There is another principal use of business data mining methods, which mainly based on the analysis and picking up of stocks and other financial instruments.

### 2.3 Fundamental Components of Data Mining Technology

It is essentially important to mention that the prime key to comprehend and realize the data mining technology is the ability to differentiate between data mining applications, operations, techniques and algorithms, as shown in Figure 2.5.



**Figure 2.5: Components of Data Mining (Han J. and Kamber M. 2001).**

### 2.3.1 Applications.

According to the (Han J.and Kamber M. 2001), this point can be defined as an implementation of data mining technology that settles and solves a particular business or research problem. Here within examples for the application of data mining process:

**A pharmaceutical company** can examine and analyze its recent sales activity and intensity and their outcomes to enhance and upgrade targeting of high-value physicians and deciding which marketing activities will have the greatest influence in the next few months. The data requires containing competitor market activity as well as information about the local health care systems. The outcomes can be allocated to the sales force via a wide-area network that enables the agents to review the recommendations from the perspective of the prime features in the decision process. The outstanding, functional analysis of the data storehouse endorses best practices from throughout the organization to be applied in specific sales situations.

**A credit card company** can use its huge warehouse of customer dealing data to identify customers most likely to be concerned and attentive in a new credit product. Using a small test mailing, the features of customers with affinity for the product can be identified.

**A diversified transportation company** with a large direct sales force can apply data mining to determine and characterize the best possibilities for its services. It can be said that using data mining to analyze its own customer experience, this company can put up unparalleled allotment identifying the peculiarities of high-value prospects. Applying this segmentation to a general business database such as those provided by Dun & Bradstreet can expand a prioritized list of prospects by region.

Data mining can be applied by a large consumer package goods company to improve and develop its sales process to retailers, where the data from consumer panels, shipments, and competitor activity can be applied to comprehend and realize the reasons for brand and store switching. Through this analysis, the manufacturer can pick out promotional strategies that best reach their target customer segments.

### **2.3.2 Operations**

It can be identified as an application that utilizes data mining technology by implementing one or more data mining operations (sometime referred to as data mining ‘tasks’), where each operation reverberates a diverse form of characterized styles or orientations in a complex data set(Han J.and Kamber M. 2001).

### **2.3.3. Classification and prediction**

Classification is the procedure that mostly assisted by commercial data mining tools. It is a process that enables organizations to find out patterns in large or complex data sets in order to solve specific business problems (Alex A. Freitas 2000).

Classification is the operation of sub-dividing a data set with consideration to a number of particular results.

**Example 1**, we perhaps want to distinguish our customers into ‘high’ and ‘low’ categories with respect to credit risk. The category or ‘class’ into which each customer is placed is the ‘outcome’ of our classification.

**Example 2** A financial services organization wishes to recognize those customers likely to be concerned in a new investment opportunity. It has sold a similar product before and has historical data showing which of its customers responded to the previous offer. The goal behind this application is to understand which factors identify likely responders to the offer, so that the marketing and sales effort can be targeted more efficiently.

There is a field in the customer record that is set to true or false, depending on whether a customer did or did not respond to the offer. This field is called the ‘target field’ or ‘dependent variable’ for the classification. The aim is to analyze the way other features of the customer (such as level of income, type of job, age, sex, marital status, number of years as a customer, and other types of investments or products purchased) effect the category to which they belong (as indicated by the target field). This information will usually be stockpiled in other fields in the customer record. The various fields included in the analysis are called ‘independent’ or ‘predictor’ fields or variables(Patricia E. N. Lutu 2002).

The most popular techniques for classification is decision trees, as shown in Figure 2.6.

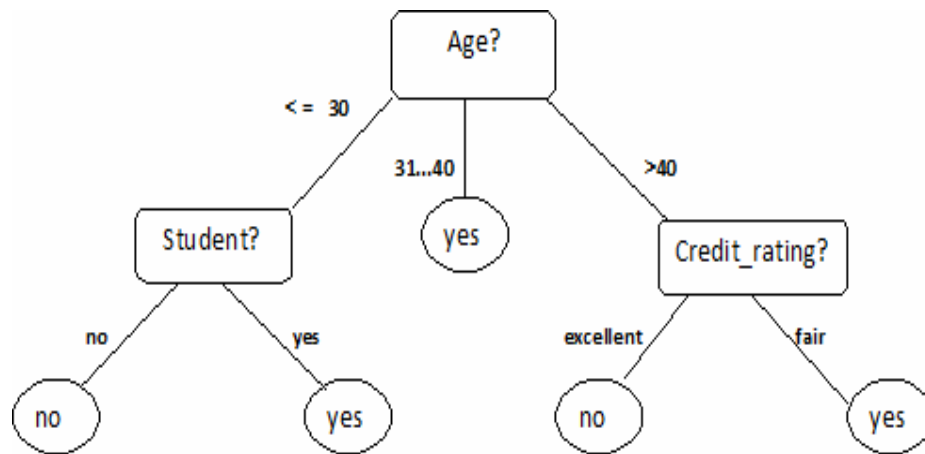
If a decision tree is used, it will provide a set of branching conditions that successively split the customers into groups defined by the values in the independent variables. The aim is to be able to produce a set of rules or a model of some sort that can identify a high percentage of responders(Han J.and Kamber M. 2001). A decision tree may formulate a condition such as:

Customers who are students with age  $\geq 30$  will respond to our offer.

Customers with age 31...41 will respond to our offers.

Customers with age  $> 40$  and who have “fair” credit rating will respond to our offer.

Thus, the situation chooses a much high percentage of responders than if you took a random selection of customers.



**Figure 2.6: Decision tree (Patricia E. N. Lutu 2002)**

A classification is a two step process, see Figure 2.7. In the first step (learning step), the training data is analyzed by classification algorithm. In the second step (classification step), test data are used to assess the accuracy and reliability of the rules. A classification model is said to be ‘trained’ on historical data, for which the outcome is known for each record. It is then applied to a new, unclassified data set in order to predict the outcome for each record. (Patricia E. N. Lutu 2002), (Han J. and Kamber M. 2001).

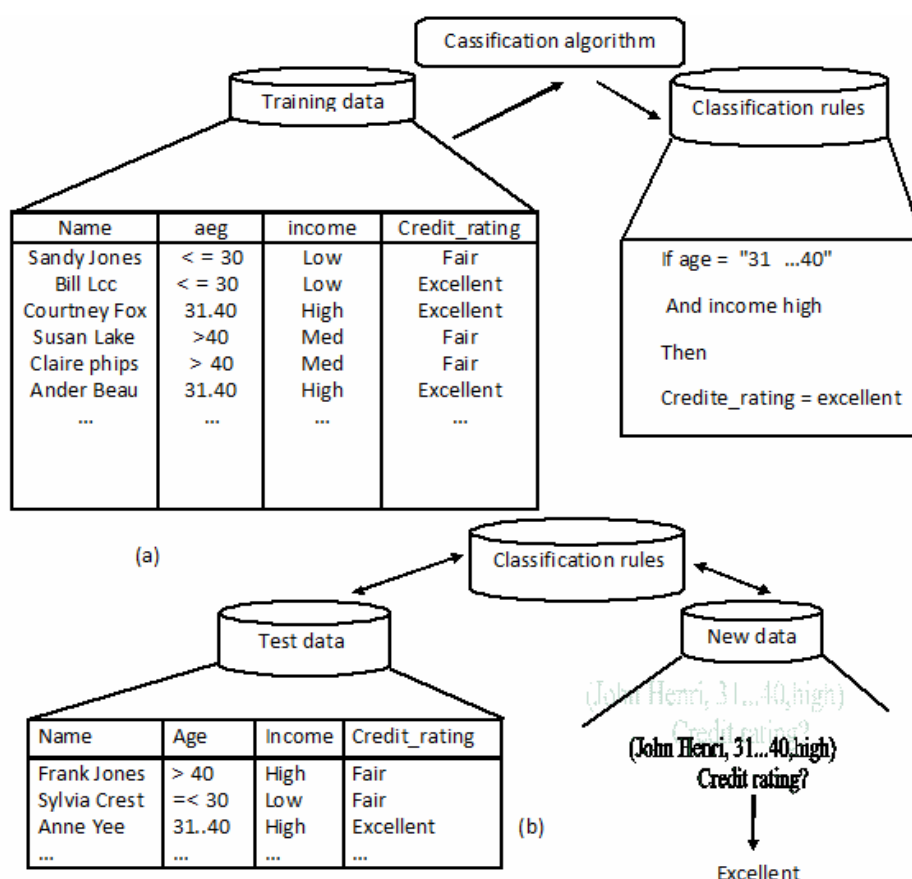


Figure 2.7: classification process steps (Han J.and Kamber M. 2001).

### 2.3.4. Association Rules

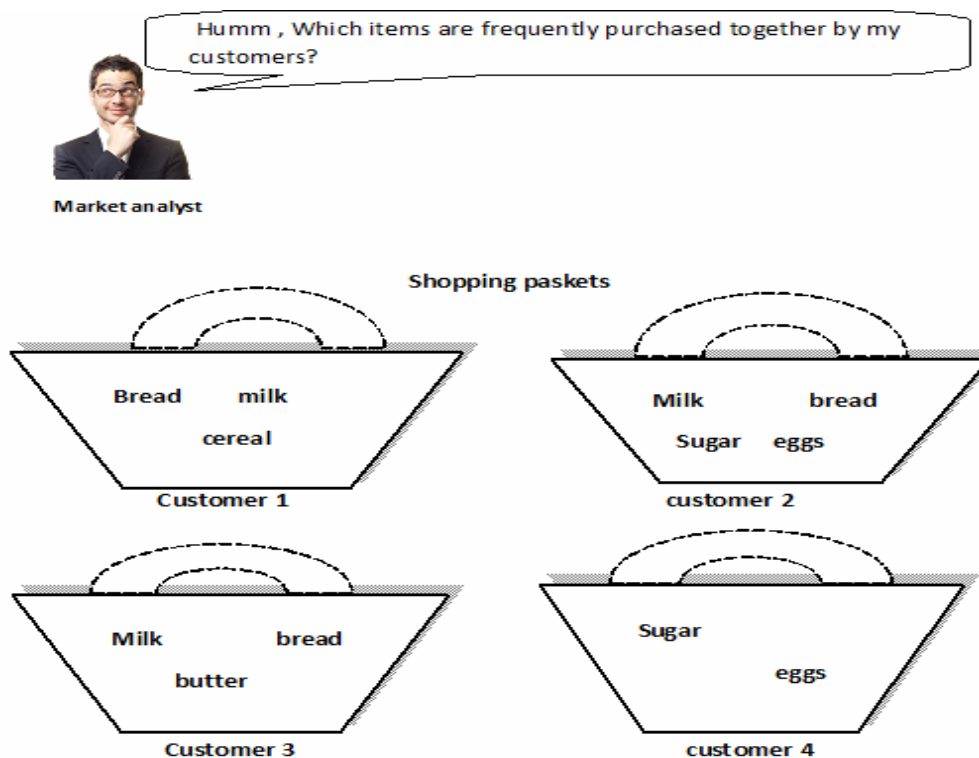
The association rule of data mining is a fundamental topic in mining of data (Ceglar A. and Roddick J. 2006). Association rule mining finds interesting association or correlation relationships among a large set of data items. Its prime idea relies in looking for causal relationships between sets of items, commonly called itemsets, where the presence of some items proposes that others follow them. The key issue that can be addressed to the contribution related to association rules is their focus on co-occurrences between items (Steinbach M., Kumar V. 2007).

## 2.4 Association Rules Methods

Association analysis is an unsupervised form of data mining that looks for links between records in a data set. It is worthy mentioning that association analysis is sometimes referred to as 'market basket analysis', its most common application, as



shown in Figure 2.8. The aim is to discover, for example, which items are commonly purchased at the same time to help retailers organize customer incentive schemes and store layouts more efficiently.



**Figure 2.8: Market basket analysis.**

**Association rules can be formally defined as follows**

Let  $\mathbf{I} = \{i_1, i_2, \dots, i_m\}$  be a set of literal, called **items**. Let  $\mathbf{D} = \{t_1, t_2, \dots, t_n\}$  be a set of transactions, where each transaction,  $t$ , is a set of items such that  $t \subseteq \mathbf{I}$ . Note that the quantities of items in a transaction are not considered. Each transaction is associated with an identifier, called TID. Given an itemset  $X \subseteq \mathbf{I}$ , a transaction  $t$  **contains**  $X$  if, and only if,  $X \subseteq t$ . The itemset  $X$  has **support**,  $s$ , in the transaction set  $\mathbf{D}$  if  $s\%$  of transactions in  $\mathbf{D}$  contain  $X$ ; we denote  $s = \text{support}(X)$ . **An association rule** is an implication of the form  $X \Rightarrow Y$ , where  $X, Y \subset \mathbf{I}$ , and  $X \cap Y = \Phi$ . Each rule has two

measures of value, support, and confidence. The **support** of the rule  $X \Rightarrow Y$  is support  $(X \cup Y)$ . The **confidence, c**, of the rule  $X \Rightarrow Y$  in the transaction set  $D$  means  $c\%$  of transactions in  $D$  that contain  $X$  also contain  $Y$ , which can be written as the ratio  $\text{support}(X \cup Y) / \text{support}(X)$ .

The task of mining association rules in transaction or relational databases is to derive a set of strong association rules in the form of " $A_1 \wedge \dots \wedge A_m \Rightarrow B_1 \wedge \dots \wedge B_n$ ," where  $A_i$  (for  $i \in \{1, \dots, m\}$ ) and  $B_j$  (for  $j \in \{1, \dots, n\}$ ) are sets of attribute values, from the relevant data sets in a database (Agrawal, Imieliński and Swami 1993). For example, a particular supermarket may find that of the 1000 customers shopping on a Thursday night, 200 bought milk, and of those 200 who bought milk, 50 bought tea. Thus, the association rule would be "if buy milk, then buy tea" with a support of  $200/1000 = 20\%$  and a confidence of  $50/200 = 25\%$ .

Another example of such an association rule is the statement that "70% of transactions that purchase bread also purchase butter." Support and confidence specified by the user are the major parameters determining the quality of the discovered association rules. The normal application of association rules is to raise the sales of some item. In this very simple example, this rule suggests a way of increasing the sales of butter in a supermarket, by placing butter in a shelf close to the bread's shelf. In real-world very large basket data, some interesting, unexpected association rules can be discovered and used in a similar way.

Consider the following sugar and rice example:

500,000 transactions

20,000 transactions contain rice (4%)

30,000 transactions contain sugar (6%)

10,000 transactions contain both rice and sugar (2%)

Support measures how often items occur together, as a percentage of the total transactions. In this example, sugar and rice occur together 2% of the time ( $10,000/500,000$ ).

Confidence measures how much a particular item is dependent on another. Because 20,000 transactions contain rice and 10,000 also contain sugar, when people buy rice, they also buy sugar 50% of the time. The confidence for the rule:

“When people buy rice they also buy sugar 50% of the time”. The inverse rule, which would be stated as:

“When people buy sugar they also buy rice 1/3 of the time” has a confidence of 33.33% (computed as  $10,000/30,000$ ).

Note that these two rules have the same support (2% as computed above). Support is not dependent on the direction (or implication) of the rule; it is only dependent on the set of items in the rule (Agrawal R. & Strikant R. 1).

In the absence of any knowledge about what else was bought, we can also make the following assertions from the available data:

People buy rice 4% of the time.

People buy sugar 6% of the time.

These numbers - 4% and 6% - are called the expected confidence of buying rice or sugar, regardless of what else is purchased.

## **2.5 Apriori Algorithm**

The researchers says that the Apriori algorithm as one of the most popular algorithms in the mining of association rules in a centralized database. The main idea of Apriori algorithm is described in the following paragraphs (Sarawak, Thomas and Agrawal 1998). Table 1 shows the notations for mining algorithm.

**Table 2.1 Notation for mining algorithm.**

k-item set	An item set having k items.
$L_k$	Set of large (frequent) k-item set.
$C_k$	Set of candidate k-item set.

1. The frequent item sets are computed through iterations. In each iteration, the database is scanned one time, and all frequent item sets of the same size are computed. The frequent item sets are computed in the ascending order of their sizes.

In the first iteration, the size-1 frequent item sets are computed by scanning the database once. Subsequently, in the  $k$ th iteration ( $K > 1$ ), a set of candidate sets,  $C_k$ , is created by applying the candidate set generating function Apriori-gen on  $L_{k-1}$ , where  $L_{k-1}$  is the set of all frequent  $(k-1)$ -itemsets found in iteration  $k-1$ . Apriori-gen generates only those  $k$ -item sets whose every  $(k-1)$ -item set subset is in  $L_{k-1}$ . The support counts of the candidate itemsets in  $C_k$  are then computed by scanning the database once, and the size- $k$  frequent item sets are extracted from the candidates.

Apriori candidate generation. The Apriori-gen function takes as an argument,  $L_{k-1}$ , the set of all frequent  $(k-1)$ -itemsets. It returns a superset of the set of all frequent  $k$ -itemsets. First, in the join phase,  $L_{k-1}$  is joined with itself, the join condition being that the lexicographically ordered first  $k-2$  items are the same, and that the attributes of the last two items are different. Second, in the subset pruning phase, all itemsets from the join result which have some  $(k-1)$ -subset that is not in  $L_{k-1}$  are deleted.

The Apriori algorithm is shown in Figure 2.9.

```

    L1 = {frequent 1-itemsets};

    For (k = 2; Lk-1 ≠ Φ; k++) do begin

        Ck = apriori-gen (Lk-1);           // New
candidates

        For all transactions t ∈ D do begin

            Ct = subset (Ck, t);           //
Candidates contained in t

            For all candidates c ∈ Ct do

                c.count ++;

            end

            Lk = {c ∈ Ck | c.count ≥ minsup}

        End

    Answer = ∪k Lk;

```

**Figure 2.9: Apriori algorithm (Sarawak, Thomas and Agrawal 1998).**

2. Generating rules. For every frequent item set **I**, we output all rules **a** ⇒ (**I**-**a**), where **a** is a subset of **I**, such that the ratio support (**I**) / support (**a**) is at least minconf.

From a frequent itemset **I**, the algorithm first generates all rules with one item in the consequent. The algorithm then use the consequents of these rules to generate all possible consequents with two items that can appear in a rule generated from **I**, etc. The rule of generation algorithm is shown in Figure 2.10.

```

For all frequent k-itemsets  $l_k$ ,  $k \geq 2$ , do begin
     $H_1 = \{\text{consequents of rules from } l_k \text{ with one item in}$ 
     $\text{the consequent}\};$ 
    Call ap-genrules ( $l_k, H_1$ );
End

Procedure ap-genrules ( $l_k$ : frequent k-itemset,  $H_m$ : set of
m-item
consequents)
    If ( $k > m+1$ ) then begin
         $H_{m+1} = \text{apriori-gen}(H_m);$ 
        For all  $h_{m+1} \in H_{m+1}$ , do begin
             $\text{Conf} = \text{support}(l_k) / \text{support}(l_k - h_{m+1});$ 
            If ( $\text{conf} \geq \text{minconf}$ ) then
                Output the rule  $(l_k - h_{m+1}) \Rightarrow h_{m+1}$ 
                With confidence = conf and
                support = support ( $l_k$ );
            Else
                Delete  $h_{m+1}$  from  $H_{m+1}$ ;
        End
        Call ap-genrules ( $l_k, H_{m+1}$ );
    End
End

```

**Figure 2.10: Rule generation algorithm (Sarawak, Thomas and Agrawal 1998).**

Example of Applying Apriori Algorithm Consider the database in Table 2.2.

**Table 2.2 Sample transaction database.**

TID	Items
100	ACD
200	BCE
300	ABC E
400	BE

For example, Let minimum-support = 50% and minimum-confidence = 60%. Since there are four records in the table, the number of transactions above the minsup is 2 ( $4 \times 50\% = 2$ ).

Figure 2.11 shows the process of finding frequent itemsets.

Database D		Candidate 1-itemset		Frequent 1-itemset	
TID	Items	Itemset	Support_Count	Itemset	Support_Count
100	A C D	{A}	2	{A}	2
200	B C E	{B}	3	{B}	3
300	A B C E	{C}	3	{C}	3
400	B E	{D}	1	{E}	3
		{E}	3		

Candidate 2-itemset		Candidate 2-itemset		Frequent 2-itemset	
Itemset	Itemset	Support_Count	Itemset	Support_Count	
{A, B}	{A, B}	1	{A, C}	2	
{A, C}	{A, C}	2	{B, C}	2	
{A, E}	{A, E}	1	{B, E}	3	
{B, C}	{B, C}	2	{C, E}	2	
{B, E}	{B, E}	3			

{C, E}		{C, E}	2	
Candidate 3-itemset		Candidate 3-itemset		Frequent 3-itemset
Itemset	Itemset	Support_Count	Itemset	Support_Count
{B, C, E}	{B, C, E}	2	{B, C, E}	2

**Figure 2.11: The process of finding frequent itemsets.**

- (1) Frequent 1-itemset generation. Scan the database and count the support for every item, the frequent 1-itemset is {A, B, C, E}.
- (2) Frequent k-itemset generation. Mining algorithm includes candidate generation and pruning two phases. Candidate 2-itemset: {{A,B}, {A,C}, {A,E}, {B,C}, {B,E}, {C,E}}.

Count each candidate 2-itemset support, then prune the 2-itemset whose support is lower than minsup. we have:

Frequent 2-itemset: {{A,C}, {B, C}, {B, E}, {C, E}}.

Candidate 3-itemset: {{B, C, E}}.

Frequent 3-itemset: {{B, C, E}}.

Since frequent 4-itemset is empty, frequent k-itemset generation terminates.

- (3) Derive the association rules. Now, we have frequent 3-itemset {{B, C, E}} where  $s = 50\%$ .

Remember the predetermined minconf = 60%. We get:

B and C  $\Rightarrow$  E, with support = 50% and confidence = 100%.

B and E  $\Rightarrow$  C, with support = 50% and confidence = 66.7%.

C and E  $\Rightarrow$  B, with support = 50% and confidence = 100%.



$B \Rightarrow C$  and  $E$ , with support = 50% and confidence = 66.7%.

$C \Rightarrow B$  and  $E$ , with support = 50% and confidence = 66.7%.

$E \Rightarrow B$  and  $C$ , with support = 50% and confidence = 66.7%.

## 2.6 Related Works.

Many researchers have been concentrating their research on the field of mining association rule by many different methods through using different techniques. Following is a brief some of these related works.

(Agrawal, Imielinske, and Swami, 1993) proposed an approach called (Mining Association Rule between sets of items in Large Databases) by introduced the problem of mining association rules between sets of items in large database of customer transaction. Each transaction consists of items purchased by a customer in a visit. Then interested in finding those rules that have, minimum support and minimum confidence. Then proposed an efficient algorithm to solve this problem which has the following features, first, it uses a carefully tuned estimation procedure to determine what itemsets should be measured in a pass. Second, it uses pruning techniques to avoid measuring certain itemsets, while guaranteeing completeness. Third, It incorporates buffer management to handle the fact that all the itemsets that need to be measured in pass may not fit in memory, even after pruning.

(Agrawal and Strikant, 1994) "Fast Algorithms for Mining Association Rules". Their work show how the best features of the two proposed algorithms (APRIORI, APRIORITID) can be combined into a hybrid algorithm, called AprioriHybrid. They presented two new algorithms, Apriori and AprioriTid, for discovering all Significant association rules between items in a large database of transactions. They compared these algorithms to the previously known Algorithms, Their work presented experimental results, showing that the proposed algorithms always outperform AIS and

SETM. The performance gap increased with the problem size, and ranged from a factor of three for small problems to more than an order of magnitude for large problems. But this performance not enough for large database transaction.

(Han et al. 1996) DMQL: A data mining query language for relational databases, they designed and developed a preliminary version of a data mining query language, DMQL, for effective data mining in relational databases and they focus on extends SQL with a series of operators for generation of characteristic rules, discriminant rules and classification rules only.

(Sarawagi, Thomas and Agrawal, 1998) integrating Association rule mining with relational database systems, they consider a spectrum of architectural alternatives for coupling mining with database systems. These alternatives include: loose-coupling through a SQL cursor interface; encapsulation of a mining algorithm in a stored procedure; caching the data to a file system on-the-fly and mining; tight-coupling using primarily user-defined functions; and SQL implementations for processing in the DBMS. but the performance of time still the big problem for mining in huge data.

(Jamil, H.M. 2001)Ad hoc association rule mining as SQL3 queries. They focus the cost of generating association rule for a given item in object relational databases has been tested to show that the cost of computing association rules does not depend on support and confidence threshold.

(Cyrille, Céline, and Jean-François 2004) Optimizing subset queries: a step towards SQL-based inductive databases for itemsets. they propose a new way to handle sets from relational database. it based on a data structure that partially encodes the inclusion relationship between sets. it is an extension of the hash group bitmap key.

(Girish K., Mmandar S. and Manoj M. 2005) proposed an approach called (Association Rule Mining Using Heavy Itemsets) that introduce a concept called a

heavy itemset. An itemset  $A$  is heavy (for given support and confidence values) if all possible association rules made up of items only in  $A$  are present. And prove a simple necessary and sufficient condition for an itemset to be heavy. Then using a formula for the number of possible rules for a given heavy itemset, and show that a heavy itemset compactly represents an exponential number of association rules. The efficient greedy algorithm used to generate a collection of disjoint heavy itemsets in a given transaction database. Then using a modified apriori algorithm that uses a given collection of heavy itemsets and detects more heavy itemsets, not necessarily disjoint with the given ones, and of course the remaining association rules.

(Mohammed J. Zaki, Ching-Jui Hsiao 2005) this study proposed CHARM(Closed Association Rule Mining), an algorithm for mining closed frequent itemsets, and CHARM-L, an algorithm for generation the closed itemset lattice. These include the following features: First, they simultaneously explore both the itemset space and transaction space over a novel IT-tree(itemsettidset tree) search space. Second, they use a highly efficient hybrid search method that skips many levels of the IT-tree to quickly identify the frequent closed itemsets, instead of having to enumerate many possible subsets. Third, CHARM uses a fast hash-based approach and CHARM-L uses an intersection-based approach to eliminate nonclosed itemsets during subsumption checking. Forth, CHARM-L explicitly outputs the frequent itemset lattice, which is useful for rule generation and visualization.

(Gang, Zu-Kuan and Yu-Lu2009) An algorithm of improved association rules mining. They propose an algorithm of association rules mining based on sequence number. The algorithm would use the method of binary Boolean calculation to generate candidate frequent itemsets of binary form, and gain support of candidate frequent itemsets by computing Sequence Number Degree (SND), which is gained through

computing these Sequence Number (SN) of all these items contained by candidate frequent itemsets.

(Alashqur2010) RDB-MINER: A SQL-Based Algorithm for Mining True Relational Databases, he introduced an algorithm called *RDB-MINER* that can be used to directly mine relational databases without having to resort to any conversions prior to starting the mining process, but in large Relational Databases the joint operation need a lot of time to execution because there are a lot of unused data redundant for the same item.

(Mirela, Stefanand Iolanda 2011) Mining Association Rules Inside a Relational Database – A Case Study. Their work consider a way to discover association rules from data stored into a relational database. They make also a comparative study of performances obtained by applying the following methods: stored procedures in database or candidate and frequent itemsets generated in SQL using a k-way join and a subquery-based algorithm. But in their research the joint operation in k\_way Method consumes time and need large space for execution.

(Rao, V.V.& R, 2011) “Efficient association rule mining using indexingsupport” Their work present the Btree index, a general and compact structure which provides tight integration of item set extraction in a relational DBMS. The Data bases may be Transactional Data bases or Relational Data bases. Since no constraint is enforced during the index creation phase.

(Alashqur 2012) Using a Lattice Intension Structure to Facilitate User-Guided Association Rule Mining, he introduced a new approach for user-guided association rule mining. In his approach, a user is presented with a lattice intension structure that encodes the different possible attribute combinations. By selecting a node from the structure, the user identifies the attribute combination of interest. This selection directs

the system to discover any association rules among data values that belong to the attributes represented by the selected node.

(Sanober and Madhuri 2012) Association rule mining based on Trade List, they proposed a new mining algorithm is defined based on frequent item set. Apriori Algorithm scans the database every time when it finds the frequent item set so it is very time consuming and at each step it generates candidate item set. So for large databases it takes lots of space to store candidate item set. In undirected item set graph, it is improvement on apriori but it takes time and space for tree generation. The defined algorithm scans the database at the start only once and then from that scanned data base it generates the Trade List.

From all previous studies still the performance issue is the biggest problem because of the large volume of data, so what distinguishes this proposal is to focus on

Maintaining the speed during the time of return data and remove redundant data and non important to maintain the storage capacity.

## **Chapter Three**

# **THE ENHANCED MODEL ARCHITECTURE IMPLEMENTATION AND TESTING**

## Chapter Three

# THE ENHANCED MODEL ARCHITECTURE IMPLEMENTATION AND TESTING

### 3.1 Introduction.

In this chapter, we present the architecture of our model, As well as the implementation of the enhanced “APRIORI” algorithm for mining association rules using oracle (database and form designer) in some details. Also, discussion the enhanced key-way method in APRIORI algorithm, finally, discussion the testing phase of our enhanced method.

### 3.2 The Model Architecture

The model architecture shown in Figure 3.1.

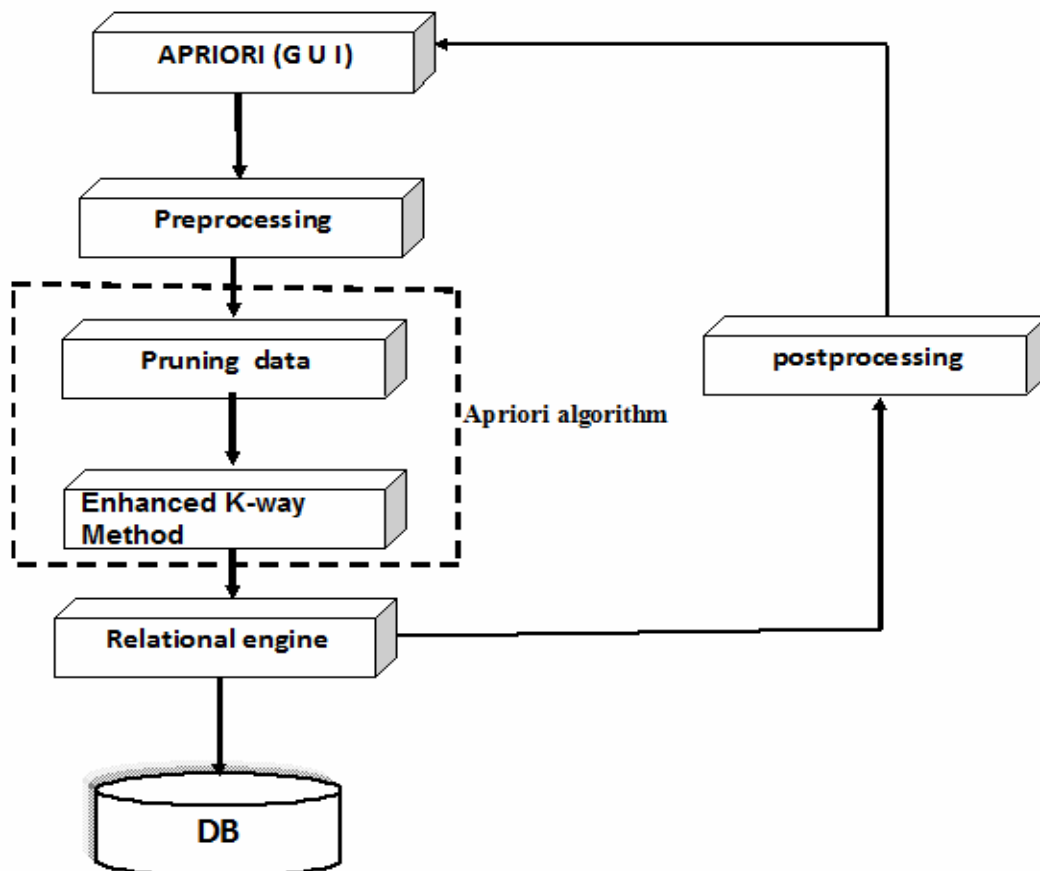


Figure 3.1: Architecture of the model.

### 3.2.1 Model User Interface

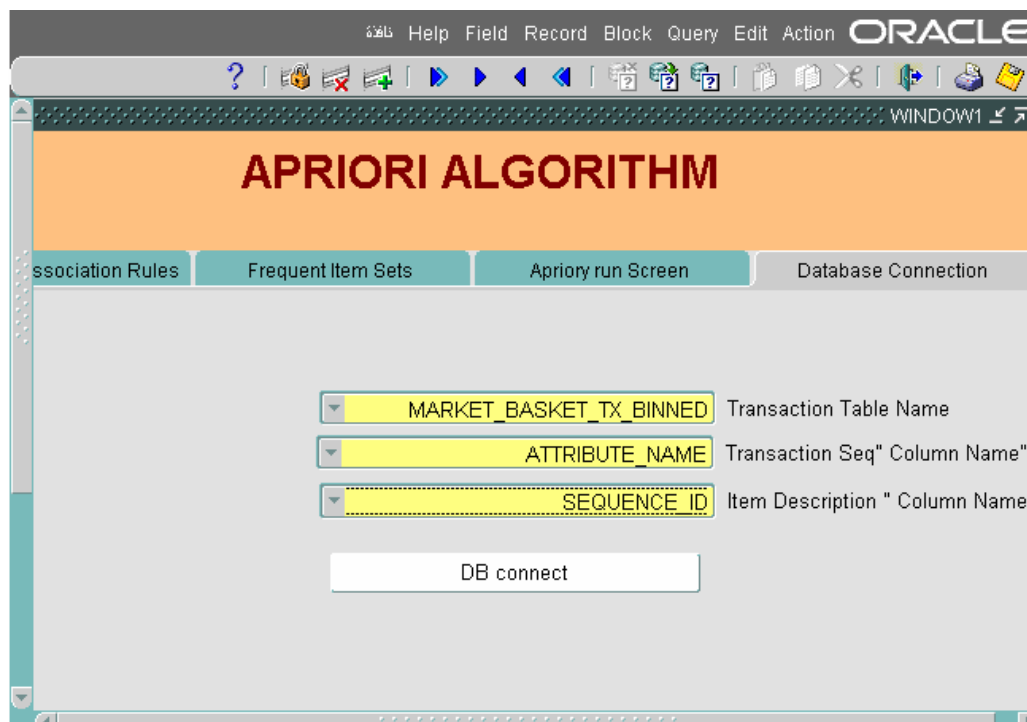
Apriori is implemented using SQL to integrate the algorithm directly with Database. We use Oracle developer9i as APRIORI Graphical User Interface (GUI) for executing and querying the mining results directly from Relational Database. The following is a brief description about APRIORI user interface.

#### 3.2.1.1 The User Interface of the APRIORI algorithm

There are many Form/Screens which represent the different processes and functions in apriori algorithm, The first form is the main form, that consists of four parts: Database connection, Run Apriori, Frequent itemsets and Association Rules. where each part consists of some components.

##### 3.2.1.1.1 Database connection

In database connection we assign the transaction table and the column name in it to be as a target transaction database, when clicking the DB Button the view is created, that contains all information in target table, as shown in figure 3.2.



**Figure 3.2:database connection.**



### 3.2.1.1.2 Run Apriori

After connected with the database we run Apriori form, in Apriori form we can specify the minimum support and the minimum confidence in text field, then we use generate association rules button, this button call apriori Procedures and run the core Apriori algorithm. Apriori algorithm pseudo\_code in sql, as shown below.

```

k = 1
C1 = generate counts from R1

repeat

    k = k + 1

    INSERT INTO R'k
    SELECT p.Id, p.Item1, ..., p.Itemk-1, q.Item
    FROM Rk-1 AS p, TransactionTable as q
    WHERE q.Id = p.Id AND
           q.Item > p.Itemk-1

    INSERT INTO Ck
    SELECT p.Item1, ..., p.Itemk, COUNT(*)
    FROM R'k AS p
    GROUP BY p.Item1, ..., p.Itemk
    HAVING COUNT(*) >= support

    INSERT INTO Rk
    SELECT p.Id, p.Item1, ..., p.Itemk
    FROM R'k AS p, Ck AS q
    WHERE p.item1 = q.item1 AND
           .
           .
           p.itemk = q.itemk

until Rk = {}

```

Figure 3.3: SQL pseudo\_code of the apriori algorithm (Olofsson N. 2010).

After finishing, the program will produce start time and end time for the period of execution and the number of transactions. after that we show the frequent itemsets

when click the show frequent itemsets button and show the association rules when click the show association rules button, as shown in figure 3.4.

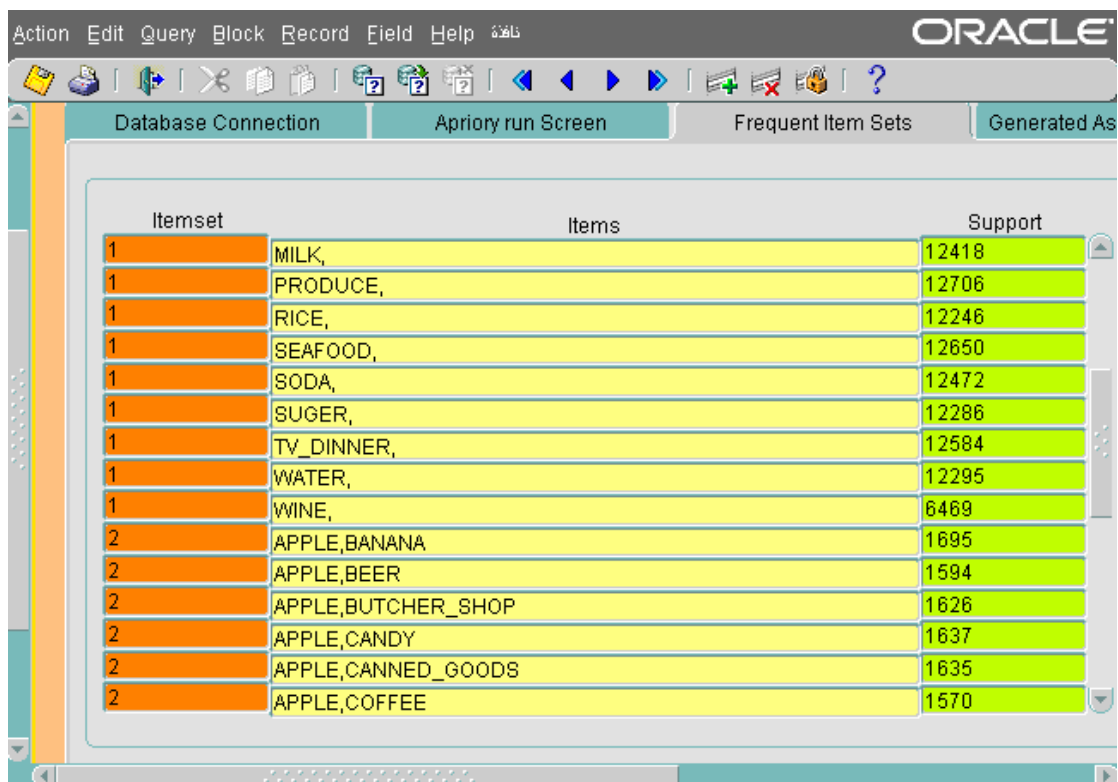
The screenshot shows the Oracle Apriori run screen. The interface includes a menu bar (Action, Edit, Query, Block, Record, Field, Help) and a toolbar with various icons. Below the toolbar, there are four tabs: Database Connection, Apriori run Screen, Frequent Item Sets, and Generated Association. The Apriori run Screen tab is active. It contains the following fields and buttons:

- Enter Minus Support(relative): 40 %
- Enter Minus Support(absolute): 100
- Transaction Count: 1
- Enter Minimum confidece: .2
- Buttons: Generate Association Rules, Show frequent itemsets, Show Association Rules
- Start time: 26-07-2013 03:39:55
- End Time: (empty field)
- Elapsed Time(seconds): (empty field)

**Figure 3.4: Run Apriori.**

### 3.2.1.1.3 Frequent Itemset

When we click show frequent itemset button, the mining result procedure execution. in frequent Itemset form, we see three components, the first one is the itemset, this component show the number of items in items field. The second component is items, this component show the name of items that frequent and greater or equal the minimum support. the third component is support, this component count the number of frequent items in transaction table, as shown in figure 3.5.



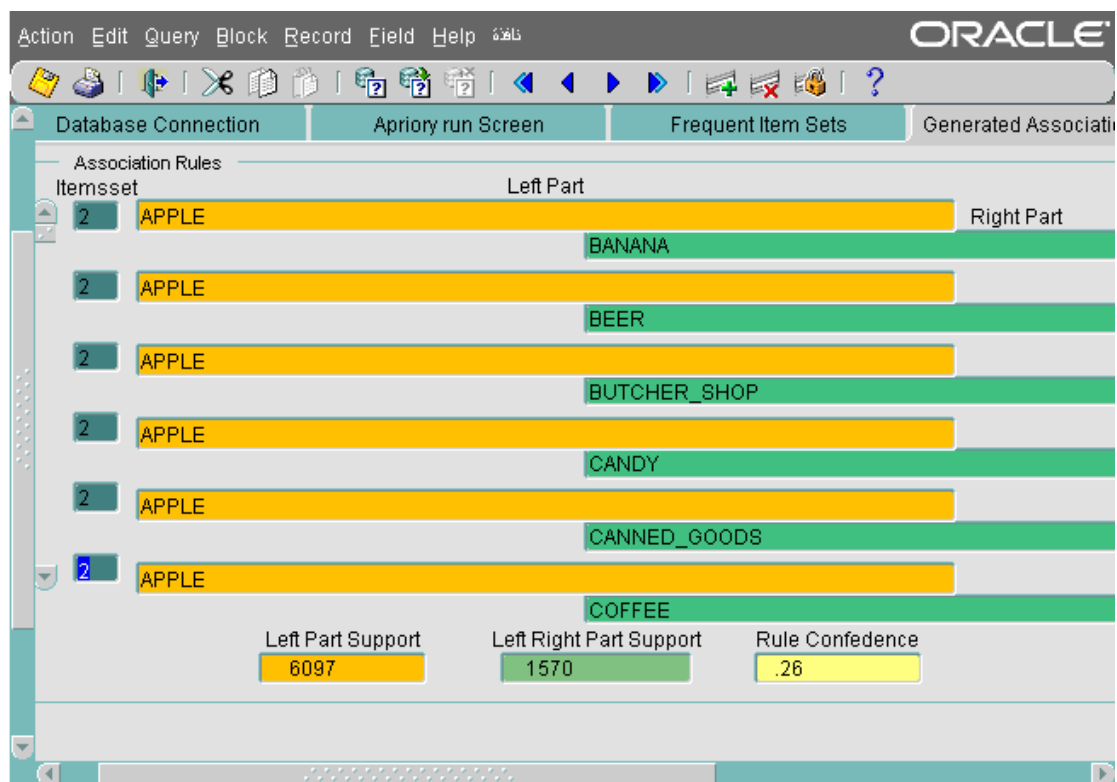
Itemset	Items	Support
1	MILK,	12418
1	PRODUCE,	12706
1	RICE,	12246
1	SEAFOOD,	12650
1	SODA,	12472
1	SUGER,	12286
1	TV_DINNER,	12584
1	WATER,	12295
1	WINE,	6469
2	APPLE,BANANA	1695
2	APPLE,BEER	1594
2	APPLE,BUTCHER_SHOP	1626
2	APPLE,CANDY	1637
2	APPLE,CANNED_GOODS	1635
2	APPLE,COFFEE	1570

**Figure 3.5: frequent Itemset.**

#### 3.2.1.1.4 Generated Association Rule

When we click show association rules button, the generated rules procedure execution, in association rule form, we see all the generated rules that has a confidence value greater than or equal the minimum confidence which user entered before, this form consist of some components, itemset component display number of item in each association rules, left part component display name of items in left side of association rules, right part component display name of items in Right side of association rules, left part support component display number of frequent item in left side of association rule, left right part support component display number of frequent association items in left and right side of association rules, and rule confidence component display the ratio of confidence in association rules, the ratio of rule confidence should greater than or equal minimum confidence which user entered in before. **An association rule** is an

implication of the form  $X \Rightarrow Y$ , As shown in figure 3.6  $BEER \Rightarrow CANNED\_GOODS$ ,  $BEER, CANNED\_GOODS \subset \text{Items in transaction table}$ , and  $BEER \cap CANNED\_GOODS = \Phi$ . Each rule has two measures of value, support, and confidence. The **support** of the rule  $BEER \Rightarrow CANNED\_GOODS$  is  $\text{support}(BEER \cup CANNED\_GOODS)$ . The **confidence, c**, of the rule  $BEER \Rightarrow CANNED\_GOODS$  in the transaction table means  $c\%$  of transactions in transaction table that  $BEER$  also contain  $CANNED\_GOODS$ , which can be written as the ratio  $\text{support}(BEER \cup CANNED\_GOODS) / \text{support}(BEER)$ .



**Figure 3.6: Generated Association rule.**

### 3.2.2 Preprocessing

Preprocessing aims to translate the mining problem in the corresponding SQL instruction set.

### 3.2.3 Enhanced k-way Method in Apriori Algorithm

Because determine the frequent itemsets is the most significant step in association rules mining, there are variety techniques and algorithm have been developed to get the frequent itemsets, but apriori algorithm still the most efficient and well known. Apriori algorithm is implemented using SQL to integrate the algorithm directly with Oracle Database by using many of Procedures and table.

In this thesis, we moved the algorithm forward by enhancing the SQL statement that is used to get frequent item sets in k-way method ; in such a way that avoid joining the item to itself as this step is not logic and consumes more time and resources, so we used an improved version of SQL statement using “>” operator to avoid joining the table to itself. DataSet used is a transaction table named “t”, the structure of table shown below.

**Table 3.1: transaction table**

Column name	Type
Sequence_id	Number
Attribute_name	Char(50)

the k-way method before and after enhanced to Generating frequent itemset (F5) using candidate itemset (C5), as shown below.

```
Select i1, i2, i3, i4, i5, count(*) support1
from c5, t t1, t t2, t t3, t t4, t t5
Where
t1.attribute_name= c5.i1 and
t2.attribute_name= c5.i2 and
t3.attribute_name= c5.i3 and
t4.attribute_name= c5.i4 and
t5.attribute_name= c5.i5 and
t1.sequence_id = t2.sequence_id and
t2.sequence_id = t3.sequence_id and
t3.sequence_id = t4.sequence_id and
t4.sequence_id = t5.sequence_id
group by i1, i2, i3, i4, i5
having count(*) > min_support
```

**Figure 3.7: Original k-way method (Sarawagi, Thomas and Agrawal, 1998)**

```

select count(*)
from t t1, t t2, t t3, t t4, t t5
where
t1.sequence_id=t2.sequence_id and
t2.sequence_id=t3.sequence_id and
t3.sequence_id=t4.sequence_id and
t4.sequence_id=t5.sequence_id;

```

**Figure 3.8: SQL statement used by K-way method before enhanced.**

```

select count(*)
from t t1, t t2, t t3, t t4, t t5
where (t1.sequence_id=t2.sequence_id
and t2.sequence_id=t3.sequence_id
and t3.sequence_id=t4.sequence_id
and t4.sequence_id=t5.sequence_id)
and (t1.attribute_name>t2.attribute_name
and t2.attribute_name>t3.attribute_name
and t3.attribute_name>t4.attribute_name
and t4.attribute_name>t5.attribute_name);

```

**Figure 3.9: SQL statement used by K-way method after enhanced.**

### 3.2.4 Relational Engine

Relational Engine process data definition language and executing the series of logical operations. Our consider the Oracle SQL dialect because it contains object relational capabilities and it allows user-defined function and functions table.

### 3.2.5 Postprocesses

The processing results are converted and presented to the user in an intelligible form through the GUI.

### 3.3 Training and Testing the Proposed Model

The Data Set used is a transaction table named “t” that has 40028 transactions, 210520 records, 20 different items, 5.25 as an average number of items in transaction. Structure of table shown below.

**Table 3.2: transaction table**

Column name	Type
Sequence_id	Number
Attribute_name	Char(50)

Generating frequent itemset (F5) has been used in this test using candidate itemset (C5) by original SQL statement, As shown Above in figure 3.7 with some enhanced. In this statement, there is a need to join the transactional table (t) to itself 5 times. Counting the number of records that would result because of such join, As shown in figure 3.10 ends up with (880985330) records retrieved within 7 minutes. This number in fact is not the correct one that we have to use in counting for frequent items. It is large number because of missing important point that there is no need to join the item in t1 to itself in t2, t3, t4, t5; so our Improved in SQL statement avoid joining the tab to itself, As shown in figure 3.11 where the correct number of records is (1839307 records) retrieved in 35 seconds only.



```
select count(*)  
  
from tt t1, tt t2, tt t3, tt t4, tt t5  
  
where  
  
t1.sequence_id=t2.sequence_id and  
  
t2.sequence_id=t3.sequence_id and  
  
t3.sequence_id=t4.sequence_id and  
  
t4.sequence_id=t5.sequence_id;  
  
COUNT (*)  
  
-----  
  
880985330  
  
----- Execution Time: 7 minutes
```

**Figure 3.10: number of retrieval records and time before enhanced.**

```
select count(*)  
  
from tt t1, tt t2, tt t3, tt t4, tt t5  
  
where (t1.sequence_id=t2.sequence_id  
and t2.sequence_id=t3.sequence_id  
and t3.sequence_id=t4.sequence_id  
and t4.sequence_id=t5.sequence_id)  
  
and (t1.attribute_name>t2.attribute_name  
and t2.attribute_name>t3.attribute_name  
and t3.attribute_name>t4.attribute_name  
and t4.attribute_name>t5.attribute_name);  
  
COUNT(*)  
  
-----  
  
1839307  
  
-----  
  
Execution Time: 35 seconds
```

**Figure 3.11: number of retrieval records and time after enhanced.**

## **Chapter Four**

### **Evaluation and Experimental Results**

## **Chapter Four**

### **Evaluation and Experimental Results**

#### **4.1 Introduction.**

In this chapter we will discuss evaluation performance of our work and the experimental results on our work. The experiment was conducted on a dataset that was previously used in other related works as standard case study. The performance measures that are used for evaluation the quality and accuracy are described in section 4.2. The experimental results are shown in section 4.3, and finally we will compare our results with other studies, which will be shown in section 4.4.

#### **4.2 Performance Evaluation**

In order to evaluate the performance and accuracy of Apriori algorithm after adding our improvements, we must evaluate it using some measures. The measures are (time and size) i.e., the period of time to retrieve the data, and the size of data to be retrieved from the database.

#### **4.3 Experimental Results**

In this section we present out experimental results of applying the enhancement that is done on "Apriori" algorithm by using SQL statement. The original statement (Han J.and Kamber M. 2001) uses the SQL statement as shown in the next paragraph to generate fifth frequent (F5). In our experiment we use a dataset (transaction table) that has 40028 transactions, 210520 records, 20 different items, and 5.25 is the average number of items in transaction. The table that is used, has two fields which are: sequence\_id [number], and attribute\_name [char(50)].

```

SELECT I1, I2,i3,I4,I5, COUNT(*) SUPPORT1
FROM C5,T T1,T T2,T T3,T T4,T T5 WHERE
T1.ATTRIBUTE_NAME= C5.I1 AND
T2.ATTRIBUTE_NAME= C5.I2 AND
T3.ATTRIBUTE_NAME= C5.I3 and
T4.ATTRIBUTE_NAME= C5.I4 AND
T5.ATTRIBUTE_NAME= C5.I5 AND
T1.SEQUENCE_ID= T2.SEQUENCE_ID and
T2.SEQUENCE_ID= T3.SEQUENCE_ID AND
T3.SEQUENCE_ID= T4.SEQUENCE_ID AND
T4. SEQUENCE_ID = T5.SEQUENCE_ID
GROUP BY I1,I2,i3,I4
HAVING COUNT(*) >:MIN_SUPPORT;

```

If we count the number of records that will be result from joining the transaction table to itself 5 times depend on the original above statement,we will see 88,098,5330 records in 7 minutes (using the SQL statement #1 below for count). This number in fact is not a correct number, that we have use in counting for frequent items. This number is huge number, because of missing important point that there is no need to join the item in table1 to itself in table2,table3,table4 and table5.

#### **Statement#1**

```

SELECT COUNT(*)
FROM T t1,T t2,T t3,T t4,T t5
WHERE
t1.SEQUENCE_ID=t2.SEQUENCE_ID and

```

t2.SEQUENCE\_ID=t3.SEQUENCE\_ID and  
 t3.SEQUENCE\_ID=t4.SEQUENCE\_ID and  
 t4.SEQUENCE\_ID=t5.SEQUENCE\_ID;

COUNT(\*)

-----

880985330

-----

Execution Time: 7 minutes

Therefore we use an enhanced version of SQL statement as shown in statement#2 below in which we get the correct number of records (1,839,307 records) within 35 seconds only.

**Statement#2**

SELECT COUNT(\*)

FROM T t1, T t2, T t3, T t4, T t5

WHERE

(t1.SEQUENCE\_ID=t2. SEQUENCE\_ID \_id

and t2. SEQUENCE\_ID =t3. SEQUENCE\_ID

and t3. SEQUENCE\_ID =t4. SEQUENCE\_ID

and t4. SEQUENCE\_ID =t5. SEQUENCE\_ID)

and (t1.ATTRIBUTE\_NAME > t2. ATTRIBUTE\_NAME

and t2. ATTRIBUTE\_NAME > t3. ATTRIBUTE\_NAME

and t3. ATTRIBUTE\_NAME > t4. ATTRIBUTE\_NAME

and t4. ATTRIBUTE\_NAME > t5. ATTRIBUTE\_NAME);

```
COUNT(*)
```

```
-----
```

```
1839307
```

```
-----
```

```
Execution Time: 35 seconds
```

Experiment results in other example, In this experiment we use a dataset (transaction table) that has 40028 transactions, 110520 records, 20 different items, and 2.76 is the average number of items in transaction. The table that is used, has two fields which are: sequence\_id [number], and attribute\_name [char(50)]. If we count the number of records that will be result from joining the transaction table to itself 5 times depend on the original statement, we will see 306,725,764 records in 2 minutes , As shown below.

```
SELECT COUNT(*)
```

```
FROM T t1,T t2,T t3,T t4,T t5
```

```
WHERE
```

```
t1.SEQUENCE_ID=t2.SEQUENCE_ID and
```

```
t2.SEQUENCE_ID=t3.SEQUENCE_ID and
```

```
t3.SEQUENCE_ID=t4.SEQUENCE_ID and
```

```
t4.SEQUENCE_ID=t5.SEQUENCE_ID;
```

```
COUNT(*)
```

```
-----
```

```
306725764
```

```
-----
```

Execution Time: 2 minutes

Therefore we use an enhanced SQL statement , we get the number of records (137,297 records) within 15 seconds only, As shown below.

```

SELECT COUNT(*)
FROM T t1, T t2, T t3, T t4, T t5
WHERE
(t1.SEQUENCE_ID=t2. SEQUENCE_ID _id
and t2. SEQUENCE_ID =t3. SEQUENCE_ID
and t3. SEQUENCE_ID =t4. SEQUENCE_ID
and t4. SEQUENCE_ID =t5. SEQUENCE_ID)
and (t1.ATTRIBUTE_NAME > t2. ATTRIBUTE_NAME
and t2. ATTRIBUTE_NAME > t3. ATTRIBUTE_NAME
and t3. ATTRIBUTE_NAME > t4. ATTRIBUTE_NAME
and t4. ATTRIBUTE_NAME > t5. ATTRIBUTE_NAME);
COUNT(*)
-----
137297
-----
Execution Time: 15 seconds

```



To explain this idea we illustrate this simple example:

Table t is a transaction table that has one transaction with 3 items.

```
SQL> SELECT * FROM T;
```

```
SEQUENCE_ID ATTRIBUTE_NAME
```

```
-----
```

```
1 item3
```

```
1 item2
```

```
1 item1
```

Using our enhancement method the statement must be rewritten to show that only one record is needed for the data mining process.

```
SELECT
```

```
t1.ATTRIBUTE_NAM ITEM1,
```

```
t2. ATTRIBUTE_NAM ITEM2,
```

```
t3. ATTRIBUTE_NAM ITEM3,
```

```

COUNT(*)

FROM T t1,T t2,T t3 WHERE

t1.SEQUENCE_ID = t2. SEQUENCE_ID

and t2. SEQUENCE_ID = t3. SEQUENCE_ID

and t1.ATTRIBUTE_NAME > t2. ATTRIBUTE_NAME

and t2. ATTRIBUTE_NAME > t3. ATTRIBUTE_NAME

GROUP BYt1. ATTRIBUTE_NAME,

t2.ATTRIBUTE_NAME, t3. ATTRIBUTE_NAME;

```

ITEM	1ITEM2	ITEM3	COUNT(*)
-----	-----	-----	-----
Item1	item2	item3	1

The below k-way (statement), we get 27 records which actually are not needed because there are many unwanted redundant records.

```

SELECT

t1.ATTRIBUTE_NAM ITEM1,

t2. ATTRIBUTE_NAM ITEM2,

t3. ATTRIBUTE_NAM ITEM3,

COUNT(*)

FROM T t1,T t2,T t3 WHERE

t1.SEQUENCE_ID = t2. SEQUENCE_ID

and t2. SEQUENCE_ID = t3. SEQUENCE_ID

GROUP BYt1. ATTRIBUTE_NAME,

t2.ATTRIBUTE_NAME, t3. ATTRIBUTE_NAME;

```

ITEM1	ITEM2	ITEM3	COUNT(*)
-----	-----	-----	-----
Item 1	item1	item1	1
Item 1	item1	item2	1
Item 1	item1	item3	1
Item 1	item2	item1	1
Item 1	item2	item2	1
Item 1	item2	item3	1
Item 1	item3	item1	1
Item 1	item3	item2	1
Item 1	item3	item3	1
Item 2	item1	item1	1
Item 2	item1	item2	1
Item 2	item1	item3	1
Item 2	item2	item1	1
Item 2	item2	item2	1
Item 2	item2	item3	1
Item 2	item3	item1	1
Item 2	item3	item2	1
Item 2	item3	item3	1
Item 3	item1	item1	1
Item 3	item1	item2	1
Item 3	item1	item3	1
Item 3	item2	item1	1

Item 3	item2	item2	1
Item 3	item2	item3	1
Item 3	item3	item1	1
Item 3	item3	item2	1
Item 3	item3	item3	1

27 rows selected.

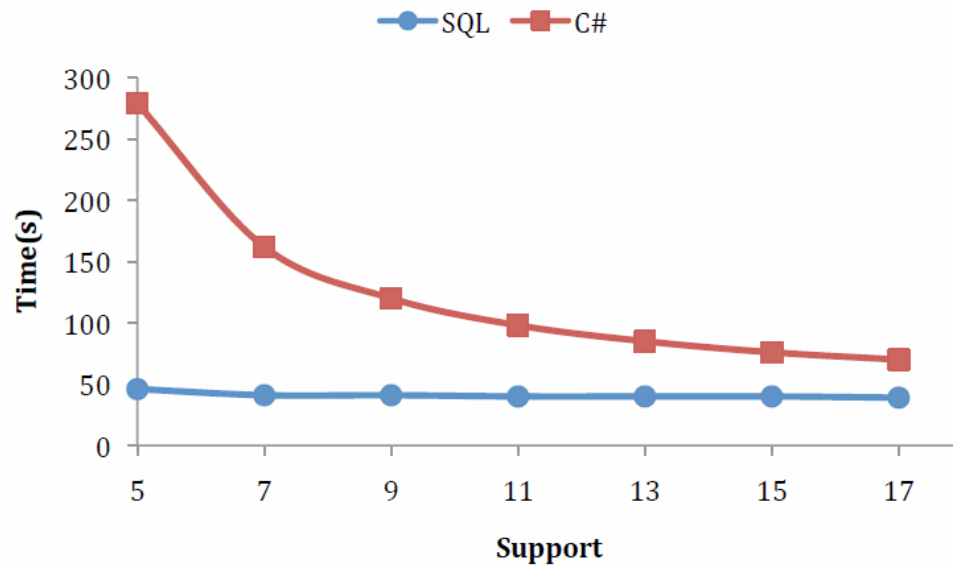
## 4.4 Comparison with Other Studies Results

In this section we will compare the performance of running Apriori algorithm with other related work with our enhanced method, As shown in the following scenarios:

- 1-Running the Apriori using SQL and using External text file.
- 2-Running the Apriori using indexes and without using indexes.
- 3-Running Apriori with and without greater than (>) operator by different support values.

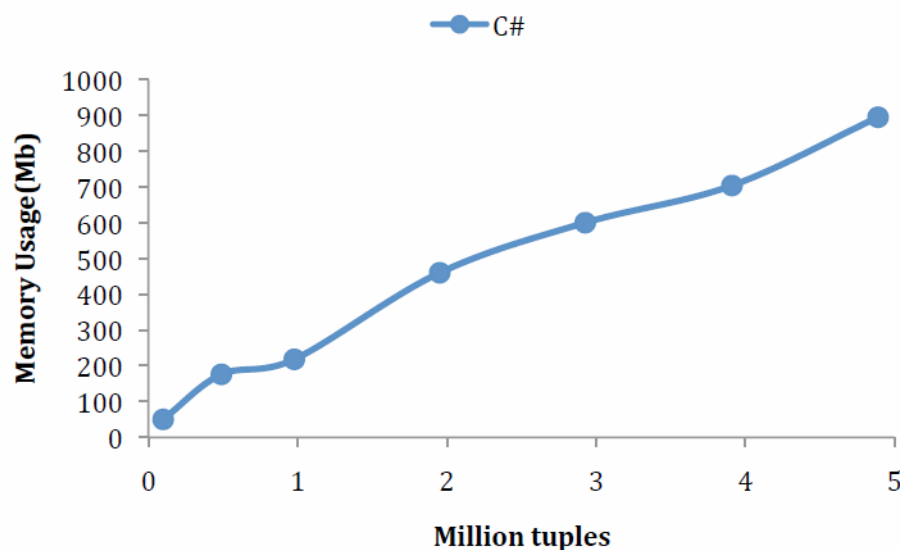
### 4.4.1 -Running Apriori using SQL and using External Text Files.

Comparing the methods efficiency of finding association rules in large amount of data (Olofsson N. 2010), The assignment was to implement the Apriori algorithm for effective item set mining in VigiBaseTM in two different ways. First via an application program written in C# and secondly directly in the database management system, The purpose is to compare the methods efficiency of finding association rules in large amounts of data with respect to execution times and memory consumption and also to list the large item sets, The results are unanimity and shows that the SQL implementation is better than C# implementation, As shown in the flowing figures.



**Figure 4.1: Execution time as a function of the support threshold. (Olofsson N. 2010).**

In figure 4.1 we have plotted the execution time as a function of the support threshold, The tests has been done on 2 million randomly selected tuples from the Adr table, When the support threshold is low, the C# version runs significantly slower whilst when the support threshold is greater, the C# version has a tendency to approach the SQL implementations execution times, Note that the SQL Implementation almost is unaffected by the change in support.



**Figure 4.2: Main memory usage of the C# version as a function of million randomly selected tuples from the Adr table with Support = 10 (Olofsson N. 2010).**

In figure 4.2 the memory usage is illustrated as a function on million tuples. The system for which this test has been done is a Apple Macbook running Windows Xp with the use of Parallels with 1 GB main memory and a CPU at 2 GHZ dual core, As seen in the figure the amount of main memory is as expected increasing with incresing tuple. It is hard checking the actual memory usage of the SQL version since MS SQL allocates all available main memory regardless of the operations beeing done, This is an intended behavior of the SQL buffer pool.<sup>2</sup> When running the C# implementation on 6 million tuples we get an out of memory

Exception,that is, the main memory has runned out whilst the SQL implementation manages this amount of tuples, This is an indication of that the C# version is more memory demanding.

However, the researcher have started to focus on issues related to integrating mining with database. There have been language proposals to extend SQL to support Mining operators. For instance, the Data Mining Query Language(DMQL) extend SQL with a collection of operators for mining characteristics rules, association rules. The M-SQL language extend SQL with special unified operator Mine to generate and query a whole set propositional rules(Imielinski T., Virmani A.& Abdulghani A. 1996) . We can summarize the benefits of using the SQL as follows:

- 1-Make use of the database indexing and query processing capabilities thereby leveraging on more than a decade of efforts spent in making these system robust (Sarawagi S., Thomas S. & Agrawal R. 1998).
- 2- Benefit from SQL parallelization to speed up computation in Symmetric multiprocessing (SMP) systems.
- 3-DBMS support for checkpointing and space management can be a valuable for long-running mining algorithms (Sarawagi S., Thomas S. & Agrawal R. 1998).

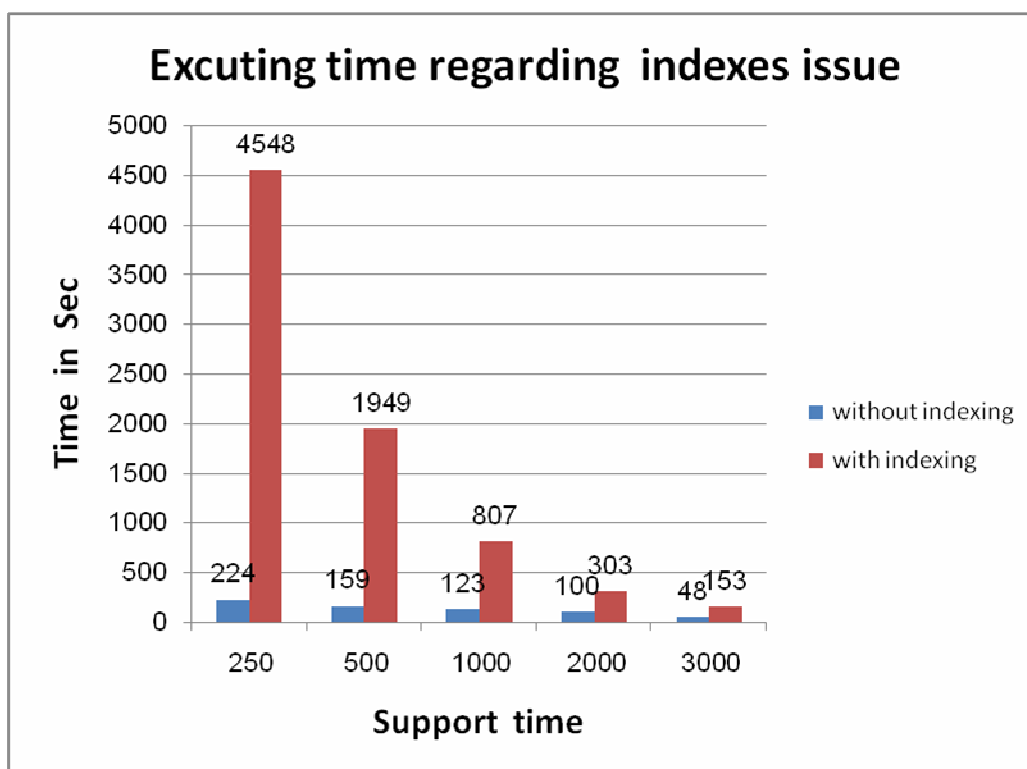
#### **4.4.2 Running the Apriori using Indexes and without using Indexes.**

A In this section we can use the Indexes to see if it give us a performance benefits. In our Implementation we create three indexes on transaction table, As shown in table 4.1.

**Table 4.1: three indexes on transaction table**

<b>Index namae</b>	<b>Indexed columns</b>	<b>Index type</b>
Seq_idx1	Sequence_id	B-Tree
Seq_idx2	Sequence_id	B-Tree
Att_Name	Attribute_name	Bitmap index

We use Oracle DBMS\_STAT.GATHER\_INDEX\_STAT procedure to collect some statistics to help the optimizer to select the best path(plan) for a given query. For more information on this subject refer to Oracle Tuning manuals (Oracle university 2001). We tried to running the Apriori in two cases, one without using indexes, and another with using indexes. During running Apriori, the optimizer choose NOT to use the indexes in executing the query and the results show better performance than using indexes. As we see in figure 4.3 With-indexes approach in the best cases will be 5 times slower than without-index approach and in the worst cases will be 22 times slower than without-indexes approach.



**figure 4.3: comparisons of running Apriori with index and without index.**



#### 4.4.3 Running Apriori with and without greater than (>) operator by different support values.

Here, we run Apriori in two cases, one without using (>) operator, and one with using (>) operator by different support count. We see from figure 4.4, that our enhancement on k-way approach by using (>) operator give significant performance enhancement in time and it is 4 times better than without using (>) operator.

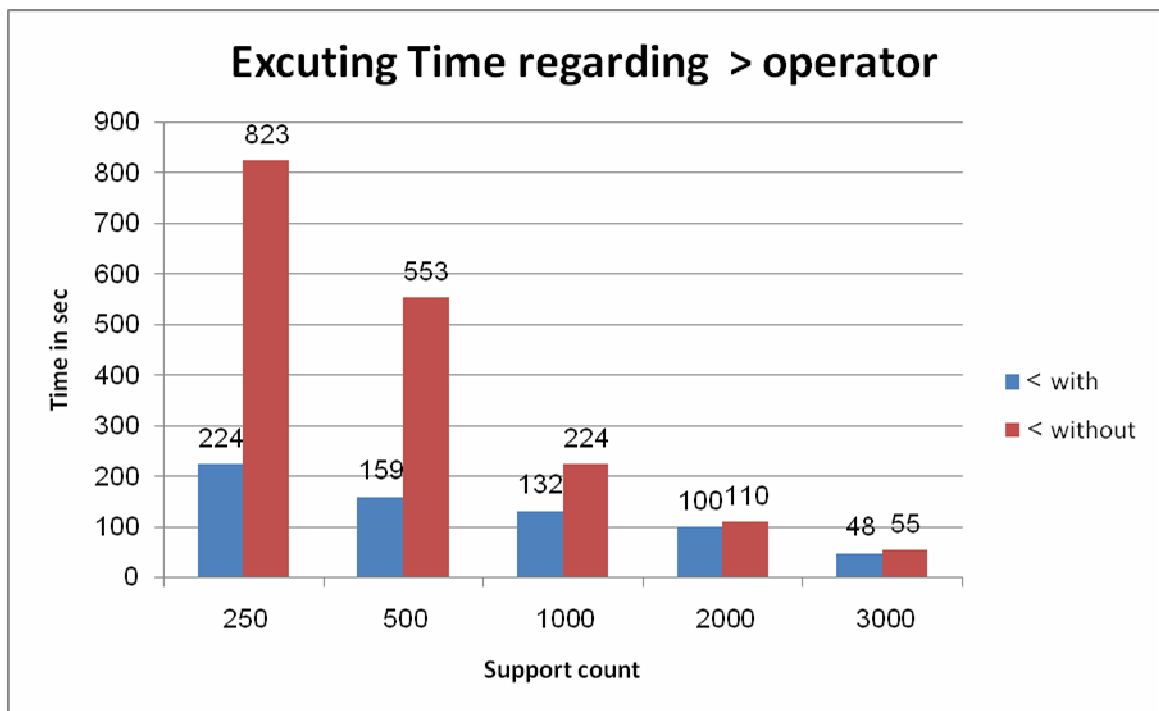


figure 4.4: comparisons of running Apriori with and without > operator.

## **Chapter Five**

### **Conclusion and Recommendations**

## **Chapter Five**

### **Conclusion and Recommendations**

#### **5.1 Introduction**

This chapter concludes the research and suggest some future directions for future work in order to make more improvement in Mining the Association Rules.

#### **5.2 Conclusion**

Where the generation of frequent itemsets is the most important and vital part of association rule mining, we introduce a promoted method for frequent itemset generation through rewriting the SQL statment of frequent itemset generation in a way that can eschew joining the item to itself, where this step is not logic and consumes more time and resources. The process that we have adopted in this thesis demonstrated a considerable performance promotion in terms of query execution time, which is less than conventional K-way method. As well as, we implement a GUI for mining association rule utilizing oracle 9i database tools, our implementation deals straightaway with transaction table in relational database (Warehouse) without any necessity to transmit relevant data to independent text file which facilitates the process of mining in the database. Also thesis elucidated that using indexes may decrease the performance of data mining process.

### **5.3 Recommendations for Future Research:**

In order to improve the performance of mining the Association Rules, We recommends the follows:

- More investigations are needed in order to find the optimal way to increase the performance in data mining such decrease retrieval time and size of data .
- focus is needed to be shed on the field of data mining in pruning redundant data and rules that make mining more efficient.
- Attention to privacy and security when some association rules of data mining are sensitive, so you need to show it in a more privacy.

## References

## References

- Alashqur A. (2010), 'RDB-MINER: A SQL-Based Algorithm for Mining True Relational Databases', *Journal of Software*, vol. 5, NO. 9, September 2010.
- Alashqur A. (2012), "Using a Lattice Intension Structure to Facilitate User-Guided Association Rule Mining," *Computer and Information Science*, Vol. 5 No. 2; March 2012.
- Agrawal R., Shim K.(1996), 'Developing tightly coupled data mining applications on a relational database system', *In Proc of the 2<sup>nd</sup> int'l Conference on knowledge discovery in Databases and Data mining*, Portland, Oregon, August 1996.
- Agrawal R., Imieliński T. & Swami A.(1993), 'Mining association rules between sets of items in large databases', *Preceedings of the ACM SIGMOD Conference on Management of Data*, New York, USA, Vol. 22, no.2, June 1.
- Agrawal R. & Strikant R. (1994). 'Fast Algorithms for Mining Association Rules', *In Proc. of the Very Large Database (VLDB) Conference*, San Jose.
- Alex A. Freitas (2000), 'Understanding the crucial differences between classification and discovery of association rules: a position paper ', *understanding the crucial differences between class and asso*, ACM SIGKDD Explorations Newsletter, Volume 2 Issue 1, June 2000.
- Al-hamami A. (2008), "Data mining:-Concept, Techniques and Applications", Ithraa Publishing and Distribution, Amman, Jordan, 2008.
- Ceglar, A., Roddick, J.F(2006).: Association mining ACM Computing Surveys, volume 38(2) (2006).
- Cyrille M., Céline R. & Jean-François B. (2004), 'Optimizing subset queries: a step towards SQL-based inductive databases for itemsets', *in processing of ACM symposium of applied Computing SAC 2004*, pp. 535-539.
- Girish K., Mmandar S., Manoj M. (2005), 'Association Rules Mining Using Heavy Itemsets', Jayant Haritsa, T.M. Vijayaraman, pp.148-155.

Gang F., Zu-Kuan W. & Yu-Lu L (2009), 'An algorithm of improved association rules mining', *In proceeding of International Conference on Machine Learning and Cybernetics*, pp. 133 – 137.

Han J., Fu Y., Koperski K. i, Wang W. & Zaiane O., (1996), 'DMQL: A data mining query language for relational databases', *In Proc. of the 1996 SIGMOD workshop on research issues on data mining and knowledge discovery*, Montreal, Canada, May 1996.

Han J. & Kamber M.(2001), *Data Mining: Concepts and Techniques*, First edition, Morgan Kaufmann publisher, USA.

Hansen, Hans Robert, Neumann, Gustaf: *Wirtschaftsinformatik I*. Lucius & Lucius, 2001.

Imielinski T., Virmani A., & Abdulghani A.(1996), 'Discover Board Application programming Interface and Query Language for Database Mining', *In proc. Of the 2nd int'l conference on knowledge discovery and data mining, Portland, Oregon, august 1996*.

Jamil, H.M (2001),. 'Ad hoc association rule mining as SQL3 queries', *Proceedings IEEE International Conference on Data Mining*, pp. 609 – 612.

Mirela D., Stefan P. & Iolanda T. (2011), *Mining Association Rules Inside a Relational Database – A Case Study. IARIA*, pp14-20.

Olofsson N.(2010) 'Implementation of the Apriori algorithm for effective item set mining in VigiBaseTM', PhD thesis, UPPSALA UNIVERSITY, Sweden.

Oracle university, Oracle 9i: Databases performance tuning. Volume1, 2001.

Patricia E. N. Lutu(2002), 'An integrated approach for scaling up classification and prediction algorithms for data mining', *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, South African Institute for Computer Scientists and Information Technologists, Republic of South Africa, pp. 110-117.

Rao, V.V.& R (2011), 'Efficient association rule mining using indexing support', *Proceedings of the International Conference on Recent Trends in Information Technology (ICRTIT)*, 3-5 June 2011, Chennai, Tamil Nadu. pp. 683 – 688.

Sarawagi S., Thomas S. & Agrawal R.(1998), 'Integrating Association rule mining with relational database systems', *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, Vol. 27, January.

Sanober S. & Madhuri R (2012), "ASSOCIATION RULE MINING BASED ON TRADE LIST," *International Journal of Data Mining & Knowledge Management Process (IJDMP)*, Vol.1, No.6.

Steinbach M., Kumar V.(2007), 'Generalizing the notion of confidence ', *Knowledge and Information Systems*, volume 12(3) (2007), pp. 279-299.

Tan P., Steinbach M. & Kumar V., "Introduction to Data Mining", Pearson Education, Inc., 2006.

Zaki M. J., Hsiao C. J. (2005), 'Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure', *IEEE Trans. On Knowledge and Data Engineering*, vol.17, no. 14, pp.462-477.