

QATAR UNIVERSITY
COLLEGE OF ENGINEERING

CONCEPTUAL BASED HIDDEN DATA ANALYTICS AND
REDUCTION METHOD FOR SYSTEM INTERFACE
ENHANCEMENT THROUGH HANDHELD DEVICES

BY

SYRINNE ADNAN BABI

A Thesis Submitted to the Faculty of

College of Engineering

In Partial Fulfillment

Of the Requirements

For the Degree of

Master of Science in Computing

June 2016

© 2016 Syrinne Adnan Babi. All Rights Reserved

Committee

The members of the Committee approve the thesis of Syrinne Adnan Babi defended on 24/5/2016.

Prof. Ali Jaoua
Thesis Supervisor

Prof. Heng Huang
Committee Member

Prof. Abdelaziz Bouras
Committee Member

Prof. Mohammed Saleh Mustafa
Committee Member

Approved:

Dr. Khalifa Nasser Al-Khalifa, Dean, College of Engineering

ABSTRACT

With the increasing demand placed on online systems by users, many organizations and companies are seeking to enhance their online interfaces to facilitate the search process on their hidden databases. Usually, users issue queries to a hidden database by using the search template provided by the system. In this thesis, a new approach based mainly on hidden database reduction preserving functional dependencies is developed for enhancing the online system interface through a small screen device. The developed approach is applied to online market systems like eBay. Offline hidden data analysis is used to discover attributes and their domains and different functional dependencies. In this thesis, a comparative study between several methods for mining functional dependencies shows the advantage of conceptual methods for data reduction. In addition, by using online consecutive reductions on search results, we adopted a method of displaying results in order of decreasing relevance. The validation of the proposed designed and developed methods prove their generality and suitability for system interfacing through continuous data reductions.

TABLE OF CONTENTS

List of Tables	vi
List of Figures	vii
List of Abbreviation	viii
Acknowledgments.....	ix
Chapter 1: Introduction	1
1.1 Introduction.....	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Document Overview	4
Chapter 2: Background and Related Work	5
2.1 Functional Dependencies	5
2.2 Mining Functional Dependency Algorithms.....	7
2.3 Formal Concept Analysis.....	13
2.4 Concept Lattice	15
2.6 Lukasiewicz Implication	19
2.7 Deep Web or Hidden Database.....	22
2.8 Online System Interfacing	25
2.9 Interfacing Small Screen Devices	26
Chapter 3: Data Reduction by Functional Dependencies Preservation: A New Conceptual Approach	29
3.1 Transform DB into an Approximate Formal Context	30
3.2 Apply Incremental Reduction to FC	33
3.3 Extracting Functional Dependencies	37
3.4 Map a Formal Context into a Database.....	39
3.5 Complexity Analysis.....	41
3.6 Validation of the Conceptual Method.....	42
Chapter 4: Methodology and Implementation	48
4.1 Offline, Static Hidden-Data Analysis	48
4.2 Real Time Data Analytic	57
4.3 Implementation	59

Chapter 5: Experimental Results, Validation and Evaluation.....	61
5.1 Testing the Developed App.....	61
5.2 Compare Developed App with eBay App.....	65
Chapter 6: Conclusion and Future Work	68
6.1 Conclusion	68
6.2 Future Works.....	69
References.....	70

List of Tables

Table 1: Relation r	6
Table 2: Comparison of the seven algorithms	12
Table 3: Formal Context	14
Table 4: Dataset R [20]	17
Table 5: Result of Π (Cylinders, Size, Mileage)[20]	18
Table 6: Comparison Between the two Data Reduction Methods	19
Table 7: Fuzzy Binary Relation R [22]	21
Table 8: Reduced Database for Different Values of δ [22].	21
Table 9: Formal Context (FC)	34
Table 10: Calculating $h_\delta(B)$	35
Table 11: Computing $f(A)$	36
Table 12: Experiment with Different Sample Sizes and Approximation Levels on the Conceptual Method [33]	44
Table 13: Evaluating the Dataset Using Three Evaluation Metrics [33].	46
Table 14: API Call Parts	50
Table 15: Average Results of Evaluating the Developed App.	66
Table 16: Average Results of Evaluating the EBay Application	67

List of Figures

Figure 1 : Concept Lattice.....	16
Figure 2 : Overview of the Entity-Oriented Crawl System [24].....	24
Figure 3: The Design Process of a Mobile Device Interface [3]	27
Figure 4: Conceptual Method	29
Figure 5: Converting a Database into a Formal Context (FC).....	31
Figure 6: Converting FC into Reduced FC	36
Figure 7: FC in ConExp.....	38
Figure 8: Extracting Attribute Implications using ConExp	38
Figure 9: Algorithm to Map Formal Context into a Database Instance.....	39
Figure 10: The Input Datasets of the Conceptual Method.....	40
Figure 11: Compact Version of DB after Running the Conceptual Method	41
Figure 12: Offline Static Data Analysis.....	49
Figure 13: API Call.....	50
Figure 14: XML Response for an API Call	51
Figure 15: API Data in the Form of an Excel Worksheet.....	52
Figure 16: Data after Preprocessing the API Request.	54
Figure 17: Applying the Conceptual Method on the Dataset	56
Figure 18: Real-Time Data Analysis	58
Figure 19: Application Workflow.....	60
Figure 20: Testing the Developed App.....	64

List of Abbreviation

Abbreviation	Explanation
API	Application Programing Interface
App	Application
FCA	Formal Concept Analysis
FC	Formal Context
FD	Functional Dependency
DB	Database
RMSE	Root Mean Square Error

Acknowledgments

First, I would like to express my sincere gratitude to my advisor Prof. Ali Jaoua for his continuous support, patience, motivation, and immense knowledge during my master thesis. His guidance helped me all the time of the research and writing this thesis.

Besides my advisor, I would like to thank my colleague Eng. Fahad Islam, Eng. Aboubakr Aqle and Eng. Eman Rezk for their insightful comments, encouragement and helps whenever I need it.

In addition, I would like to thank my family for their unconditional love and support during my master thesis.

This contribution was made possible by NPRP-07-794-1-145 grant from the Qatar National Research Fund (a member of Qatar foundation). The statement made herein are solely the responsibility of the author

Chapter 1: Introduction

1.1 Introduction

Recently, with the increased use of online systems for different scientific and/or marketing purposes, many organizations are working toward enhancing their web interfaces to facilitate user searches and provide better results. Usually, these search interfaces are correlated to hidden databases or deep web content. One way to enhance these online interfaces is based primarily on hidden data analytics. In [1], a third-party system named “MOBIES (MOBile Interface Enhancement System)” was implemented to improve the online mobile interface of a hidden database using data analytics. This helps identify the domain of the attributes of the hidden database. Another enhancement method was proposed in [2] based on “attribute temporality” to allow for tailoring of the interface. An automatic interface for mobile devices was generated in [3], which originates from constraint-based and division methods, such as depth-first and breadth-first principals.

In this thesis, a technique for enhancing the web interface of hidden databases is developed. Our technique depends on the Formal Concept Analysis theory and data reduction methods.

Formal Concept Analysis (FCA) was first introduced as an applied mathematical field derived from concepts and concept hierarchy paradigms[4]. It has been widely used in different fields, like medicine, biology, computer science, electrical, chemical and civil engineering, sociology and linguistics [4]. This mathematical tool has been used efficiently to identify functional dependencies (FDs) [5] from a relational database after transforming the database into a binary relation between the object’s set and the attribute’s set and building a formal context (FC). It was mathematically

proven that the attribute implications extracted from the FC are equivalent to functional dependencies in the relational database [6]. However, this transformation leads to a quadratic representation of the dataset compared to the original relation; thus, data reduction methods are used to minimize the number of records obtained in the FC [6].

In this thesis, two types of data analytics are performed on the hidden data. The first type is an offline static hidden-data analysis performed on the hidden database. The second type is an online real-time data analysis for the enhancement of the interface of an online system. The aim of an offline static data analysis is to discover the domain of the hidden database in terms of attributes and their cardinalities. In addition, functional dependencies are extracted at this stage for the knowledge discovery (KD) process of deep web data. Real-time data analysis, mainly using conceptual data reduction, is applied to enhance the interface of the mobile application by providing different consecutive summaries of the current remaining results for a better representation of query results.

1.2 Problem Statement

In this thesis, a new approach based on offline static data analysis and real-time data analysis is developed to enhance the web interface of an e-commerce website such as eBay so it can be used on a small screen device.

The success of the developed techniques can be measured by answering the basic research questions of this thesis:

- To what extent is an offline static data analysis useful for discovering the domain of a hidden database?

- Is the implemented real-time data analysis using approximate formal context with conceptual data reduction and the preservation of functional dependencies improving the mobile application interface?
- Are the results obtained from a search query after using the developed techniques satisfying the users?

1.3 Objectives

The aim of this thesis is to propose a general methodology based on data analysis of a hidden database that might be applied to improve the user interface and enhance the user experience. The general method could be applied to small screens for a commercial website such as eBay. This goal can be met through the following specific objectives:

- Propose offline static data analysis to discover the domain of hidden databases and mining FDs.
- Explore real-time hidden-data analysis to improve the interface of mobile applications.
- Explain to what extent applying formal concept analysis on reduced search data based on the preservation of functional dependencies is rewarding.
- Study the quality and efficiency of online real-time systems based on the new interfacing method.
- Present test results in detail for at least one real-life application on small screens on android devices.
- Provide recommendations for future research work.

1.4 Document Overview

This thesis is structured as follows: First, we start in this current chapter with an introduction. In chapter 2, I present background and related work about functional dependencies, formal concept analysis, concept lattice, the data reduction method, the Lukasiewicz implication, hidden databases, online system interfacing and interfacing small screen devices. In chapter 3, I discuss the new conceptual approach that is proposed, which is data reduction by the preservation of functional dependencies, and I justify its utilization, as reduced databases have a good prediction accuracy. After that, one may find a presentation of the methodology and its implementation in chapter 4, which describes the two stages of offline static data analysis and real-time data analysis. In chapter 5, I will talk about the experimental result, the evaluation and the validation of the proposed interface. Finally, in chapter 6, I conclude my work and give recommendations for future development.

Chapter 2: Background and Related Work

This chapter discusses the background and related work in regards to functional dependencies, algorithms for mining functional dependencies, formal concept analysis, concept lattices, the data reduction method, the Lukasiewicz implication, a deep web or hidden database, online system interfacing and interfacing with a small screen design .

2.1 Functional Dependencies

In a relational database, different types of dependencies exist. Among these different types of dependencies, functional dependencies are the most important and have been widely studied in research [5][6][7][8][9][10]. Characterizing functional dependencies from a relational database is an important database topic and has many applications in different fields, such as database management, query optimization, database normalization and reverse engineering. In a given relation, functional [6]dependency is usually used to express the relationship between the attributes. A functional dependency can be expressed as “ $X \rightarrow Y$ ” [10], which means that the value of attribute X functionally determines the value of attribute Y and that any two tuples that share attribute X 's value also share attribute Y 's value [10].

Definition 1: “Let R be a database schema; a functional dependency over R is an expression $X \rightarrow A$, where $X \subseteq R$, and $A \in R$. The functional dependency $X \rightarrow A$ holds in an instance of relation r if and only if $\forall t_i, t_j \in r, t_i[X] = t_j[X] \rightarrow t_i[A] = t_j[A]$ ” [10]. The below table expresses the functional dependency between the attributes in the relation r .

Table 1: Relation r

A	B	C	D
2	5	4	1
3	5	4	2
2	6	4	1
3	5	7	2

From the above relation r in Table 1, we can conclude that $A \rightarrow D$ and $D \rightarrow A$ are valid in the relation r ; however, $A \rightarrow C$ does not hold since $t_2[A] = t_4[A]$ and $t_2[C] \neq t_4[C]$.

Definition 2: “The set of all dependencies that include F as well as all dependencies that can be inferred from F is called the closure of F , and it is denoted by F^+ ” [36].

The following will explain the well-known set of “Armstrong inference rules” used to determine new dependencies from a given set of dependencies [36]:

- **“IR1 (Reflective Rule):** If $X \supseteq Y$, then $X \rightarrow Y$, which means that a set of attributes always determines itself or any of its subsets” [36].
- **“IR2 (Augmentation Rule):** $\{X \rightarrow Y\} \models XZ \rightarrow YZ$, which means that adding the same set of attributes to both the left- and right-hand sides of dependency yields in another valid dependency” [36].

- **“IR3 (Transitive Rule):** $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$, which means that functional dependencies are transitive” [36].
- **“IR4 (Decomposition or Projective Rule):** $\{X \rightarrow YZ\} \models X \rightarrow Y$, which means that we can remove attributes from the right hand side of dependency” [36].
- **“IR5 (Union or Additive Rule) :** $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$, which means that we can combine a set of dependencies into a single one” [36].
- **“IR6 (Pseudo-Transitive Rule):** $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$, which allows us to replace a set of attribute Y on the left hand side of a dependency with another set X that functionally determines Y” [36].

2.2 Mining Functional Dependency Algorithms

In the literature, there have been many research studies done in the field of discovering the functional dependencies of a database. Many algorithms have been developed for the goal of extracting functional dependencies from a given set (or sets) of data. Among these algorithms, there were seven that were the most cited and important algorithms [11]. According to [11], these algorithms were categorized into three main categories, which are lattice traversal algorithms, difference- and agree-set algorithms and dependency induction algorithms. In the lattice traversal algorithms, a powerset lattice of attribute combinations was built and traversed either by using a bottom-up traversal strategy or by using a depth-first random walk. It produces FD candidates and validates it using a stripped partition approach. TANE, FUN, FD_MINE and DFD algorithms are all lattice traversal algorithms. Two algorithms belong to the category of difference- and agree-set algorithms, which are DEP-MINER and FASTFDs. In this category, difference and agree sets are generated to characterize minimal functional dependency by looking for sets of attributes that have

the same values in some tuples. After obtaining the agree sets, valid FDs are extracted from them. The dependency induction algorithm category, such as FDEP, begins by assuming that each attribute is functionally determining other attributes and then by using observation on the data set the FDs were either validated or removed [11].

The TANE algorithm introduced by Huhtala et al. [12] was used for extracting FDs and approximate FDs. The search space in this algorithm is represented as a Hasse diagram of its attributes. Its functionality to detect FDs is based on three main principles to detect. First, it uses partition refinement to determine whether an FD is holding or not. Second, it uses apriori-gen functions to verify that only minimal functional dependencies are discovered. Finally, to reduce the search space of the lattice, pruning rules are used [12]. In the TANE algorithm, the input lattice is divided into levels of attributes, where each level number represents the attribute combination size. Detecting FDs starts from level 1 of the lattice and continues upward, level by level. Each attribute combination in every level is tested for functional dependency. Then, the supersets of the detected FDs are pruned using pruning rules. At the last step of the algorithm, the apriori-gen function tests the attribute combinations that were left unchecked from the previous level [12].

The FUN algorithm, which is proposed by Novelli and Cichetti [13], is a level-wise algorithm that traverses the attribute input lattice relation level after level bottom-up and uses the partition refinement techniques to characterize functional dependencies. It uses the concept of free sets and non-free sets to validate FD candidates that result in non-minimal FDs [13]. By free sets, we mean, “the sets of attributes that do not include attributes that are functionally dependent on another subset of attributes” [13]. Other attributes that are not in the free sets are members of non-free sets. In this algorithm, the attribute sets are validated incrementally, level by level, based on their length, after considering the knowledge acquired from the previous level. For testing to identify if a

candidate is a free set or not, a comparison is done between its cardinality and the cardinality of its entire maximal subset. If it is found that it is a free set, then it might be a possible candidate for FD. Otherwise, its entire superset of attributes are discarded because they are not candidates for FDs. The candidate generation mechanism follows the Apriori algorithm [14].

Similar to the TANE and FUN algorithms, in the DFD algorithm [15], a powerset lattice is built, which represents all attribute combinations. In order to characterize FDs, the powerset lattice has to be traversed in depth-first random walk to test valid FDs. This power set lattice is considered to be multiple lattices that are traversed one after another—using a decidable path [15].

The FD_MINE algorithm proposed by Yao et al. [8] is another algorithm belonging to the lattice traversal category that traverses the attribute lattice level-wise bottom up by using stripped partitions and partition intersection techniques to mine the FDs. In addition to the latter techniques, it uses the equivalence classes concept of attribute sets to minimize the number of sets and FDs to be validated [8]. If two attribute sets are functionally dependent on each other, then they are considered equivalent. Each level in the attribute lattice is visited and validated to detect the FDs and the equivalent FDs. Whenever equivalent attributes are discovered, the algorithm prunes only one from the lattice because they are functionally dependent [8].

The DEP_MINER algorithm was proposed by Lopes et al. [10]. Its basic idea works on computing the minimal FDs from the agreed sets of attributes and their inverse difference agreed sets. Agreed sets can be defined as sets of attributes that have the same value in some tuples, and their inverses are the difference agreed sets. Executing the algorithm can be divided into five stages [10]. In the first stage, the algorithm computes the stripped partition for each attribute in a given relation to be used in the second stage for calculating the agreed sets of attributes. Partitions can be defined as

sets of equivalence classes that contain tuples that have the same values in given attributes [10]. A stripped partition can be defined as a partition that groups equivalence classes having a size greater than one [10]. In the third stage, the agreed set is converted to maximal sets of attributes, which are sets of attributes with no supersets that have the same value in any two given tuples. Next, in the fourth stage, complement sets are calculated from the agreed sets. In the last stage, the minimal FDs are extracted using a level-wise search on the complement sets [10].

The FASTFDs algorithm proposed by Wyss et al. [16] uses the concept of agree sets and difference agree sets of attributes to detect minimal FDs. This algorithm is considered an improvement upon the DEP_MINER algorithm, and it was proven more efficient than DEP_MINER. It depends on a depth-first, heuristic-driven search (DFHS) to traverse the search tree of attributes. Similar to the DEP-MINER algorithm, it starts by calculating the agree sets of attributes to extract the FDs, then it calculates the difference sets as “ $D_r := \{ R \setminus X \mid X \in \text{ag}(r) \}$ ” (where R is a database schema and $\text{ag}(r)$ denotes agree sets) on the agree sets to derive the minimal functional dependency [16]. After that, it computes the difference set of r module A as “ $D_r^A := \{ D - \{A\} \mid D \in D_r \wedge A \in D \}$,” which was proven to be more efficient than the complement set calculated in DEP_MINER [16]. Then, at the last stage, it finds the minimal cover over “ D_r^A ” by traversing the search tree using the depth-first driven heuristic.

The FDEP algorithm was proposed by Flash and Savnik [17]. Unlike the previously mentioned algorithms, FDEP uses a special approach to extract the FDs. This approach is based on the comparison of every two tuples in any given relation to find the minimal FDs. There are three mechanisms for implementing the traversal of the FDEP search tree; they are top-down, bidirectional and bottom-up. Among the three versions of the FDEP algorithm, the bottom-up approach was proved to be the best in performance [17]. It consists of two phases, which are

calculating the negative cover construction and the negative cover inversion. The negative cover set refers to the set of all non-FDs for a given relation and contains all non-FDs that do not hold in a given relation, while the negative cover inversion refers to the set of all minimal FDs. Thus, the algorithm identifies the set of all FDs after transforming the negative cover into the positive cover of FDs (i.e. it represents the set of all minimal FDs).

Many experiments were conducted for assessing the performance of the seven algorithms and shows their strength and weakness points[11]. It was observed that lattice traversal algorithm performs best with dataset that contains few tuples. However, its performance decreases with dataset that contains many attributes. Moreover, difference and agreed set algorithm as well as dependency induction algorithm perform well with dataset that contains many attributes but their performance decreases with dataset that contains many tuples[11]. The below table 2 summarizes the performance of the seven algorithms.

Table 2: Comparison of the seven algorithms

Algorithms	Row Scalability	Runtime	Column Scalability	Runtime
		(Row Sc)		(Column Sc)
TANE	Best with many rows dataset	Linear	Best with low columns dataset	Exponential
FUN	Best with many rows dataset	Linear	Best with low columns dataset	Exponential
DFD	Best with many rows dataset	Linear	Best with low columns dataset	Exponential
FD_MINE	Best with many rows dataset	Not Applicable	Best with low columns dataset	Exponential
DEP_MINER	Best with few rows dataset	Quadratic	Middle performance	Exponential
FASTFD	Best with few rows dataset	Quadratic	Middle performance	Exponential
FDEP	Best with few rows dataset	Quadratic	Best with many columns dataset	Exponential

One application of extracting functional dependency was used in [38] for completing missing data. If the functional dependency was already defined for a given set , then the dependency was used for completing missing data .Otherwise, the functional dependency was extracted first by transforming the dataset into Formal Concept Analysis then applying data reduction method to reduce the size of the Formal Context and Finally extracting the Functional dependency using “ ConExp “ tool [38].

2.3 Formal Concept Analysis

Formal Concept Analysis (FCA) is a mathematical tool used in the data analytics field. It was originally created based on concepts and concept hierarchy [4]. The term “concept” can be defined as “the basic unit of thought formed in dynamic processes within social and cultural environments” [4]. A concept consists of two parts: The first part is the “extension” and represents all objects that belong to the concept, and the second part is the “intension” of the concept, which represents all attributes shared by the extension [4]. In FCA, there exists an ordered relationship between the concepts of a context, called the “subconcept-superconcept relation “. In this relation, the “extension” of the “subconcept” is included in the “extension” of the “superconcept” and the “intension” of the “subconcept” includes the “intension” of the “superconcept” [4].

Definition 3: “A formal context is defined as a set of structure $\mathbb{K} := (G, M, I)$, for which G and M are sets, while I is a binary relation between G and M , i.e. $I \subseteq G \times M$ ” [4].

The elements of G represent the objects, and the elements of M represent the attributes.

To obtain a formal concept in a given context, two derivation operators are used for $X \subseteq G$ and $Y \subseteq M$, as follows [4]:

“ $X \mapsto X^I := \{m \in M \mid g I m \text{ for all } g \in X\}$ ” (1)

“ $Y \mapsto Y^I := \{g \in G \mid g I m \text{ for all } m \in Y\}$ ” (2).

“A formal concept of a formal context $\mathbb{K} := (G, M, I)$ is represented as the pair (A, B) , where $A \subseteq G$, $B \subseteq M$, $A = B^I$ and $B = A^I$; A and B are called the extent and the intent of the formal concept, respectively” [4].

The above two derivation operators are used as a general method in the literature to obtain formal concepts “ (X^{II}, X^I) and (Y^I, Y^{II}) ” [4][3]. “The subconcept-superconcept order relation is mathematically defined as follows “ [4]:

“ $(A_1, B_1) \leq (A_2, B_2) : \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_1 \supseteq B_2)$ ” (3).

Example: Consider the following formal context shown below in Table 2.

Table 3: Formal Context

	d1	d2	d3
c1	×	×	
c2			×
c3	×	×	

From the above table and using the two derivative operators defined previously, we can obtain the following concept $(\{c1, c3\}, \{d1, d2\})$, which can be interpreted as saying that both object $c1$ and $c3$ share attributes $d1$ and $d2$.

2.4 Concept Lattice

A concept lattice, which is known also as Galois lattice, has many applications in several different fields, like information retrieval, knowledge representation and bioinformatics [19]. The concept lattice is represented as a Hasse diagram of concepts. The input of a concept lattice is the set of all ordered formal concepts of the context (G, M, I) .

Definition 4: “A concept lattice can be briefly defined as follows. Given a binary relation between an object set and an attribute set, a concept is a pair of object set A and attribute set B denoted by (A, B) and the concept lattice is the partially ordered set of all concepts” [19].

Basic Theorem on Concept Lattice: “Let $\mathbb{K} = (G, M, I)$ be a formal context. Then, (\mathbb{K}) is a complete lattice, called the concept lattice of (G, M, I) , for which infimum and supremum can be described as follows” [4]:

$$“\bigwedge_{t \in T} (A_t, B_t) = (\bigcap_{t \in T} A_t, (\bigcup_{t \in T} B_t)^\parallel)” \quad (1)$$

$$“\bigvee_{t \in T} (A_t, B_t) = ((\bigcup_{t \in T} A_t)^\parallel, \bigcap_{t \in T} B_t)” \quad (2).$$

The following figure (Figure 1) represents the concept lattice of the formal context in Table 2.

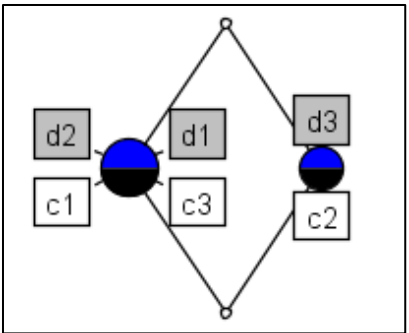


Figure 1 : Concept Lattice

2.5 Data Reduction Methods

Since many databases contain massive number of records and attributes, it is essential to reduce the size of the database either by eliminating unnecessary attributes, removing rows, reducing both database dimensions while preserving some relevant information or keeping a good prediction accuracy. There have been many database reduction methods in the literature. In [20], a technique was developed to exact or approximate the reduction of database attributes. This technique was implemented successfully using Structured Query Language (SQL) in a database. The approximate reduction technique is beneficial for dirty data, while the exact reduction method is used when no dirty data exist.

In the exact reduction of [20], all the information is preserved by eliminating only superfluous attributes, which are attributes or sets of attributes that are not required for the correct classification of database objects.

Consider the example below in Table 4:

Table 4: Dataset R [20]

Cyl	Size	Power	Weight	Mileage
4	Sub	Low	Low	High
4	Sub	Medium	High	High
6	Comp	Medium	Medium	*
4	Comp	Medium	High	Low
6	Comp	High	High	Low

In the above table, the first four columns represent the condition attributes, and the last column represents the decision attribute, which is mainly an attribute that is used to classify or categorize database objects. The reduction process is done using two database operations, which are project and card. The project operation (Π) is used to eliminate some columns and remove duplicated rows, as illustrated in Table 5. After applying the project operation, two columns are deleted, and one row is removed.

Table 5: Result of π (Cylinders, Size, Mileage)[20]

Cylinders	Size	Mileage
4	Sub	High
6	Comp	*
4	Comp	Low
6	Comp	Low

The card operation is performed after applying the projection process to return the number of rows.

From the above table, $\text{card}(\pi(\text{Cylinders, Size, Mileage})) = 4$ [20].

In [20], the approximate reduction was performed by computing the information preservation ratio (IPR) and the information loss ratio (ILR) for a subset of attributes that is included in a set such as $Q' \subseteq Q$.

$$\text{IPR} = \text{Card}(\pi(Q - Q')) / \text{Card}(\pi(Q - Q' + \{c\}))$$

$$\text{ILR} = 1 - \text{IPR}$$

When IPR is near to 1, it implies that discarding attributes in the set Q' yields the loss of a very small portion of information. ILR shows the amount of decrease in the dependency degree when eliminating attributes in Q' . It was noticed that eliminating one attribute reduces the IPR by 13%, while eliminating two attributes decreases the IPR value by 23% [20].

In [21], an automated scalable approach was proposed by Tuya et al. for reducing large database rows using sets of SQL queries and a coverage criterion. This reduction approach was implemented to reduce the size of test databases. The general idea of this method is to create a subset of meaningful data from the production database to become the test database. To accomplish the task of selecting a subset of meaningful data, a test criterion, which is called SQL full predicate coverage (SQLFpc), is used. SQLFpc consists of modified condition/decision coverage (MCDC) for SQL and creates sets of coverage rules written in SQL expressions. Then, these rules are executed to determine whether the specific requirements for a given query are satisfied [21]. After that, the data that meet the coverage rule are collected, reduced and inserted into a new empty database that becomes the reduced test database. Several optimization techniques were performed for this method to reduce the reduction time by parallelizing different tasks and reducing the rows retrieved from the initial database [21]. Table 6 compare the above two disused data reduction methods.

Table 6: Comparison Between the two Data Reduction Methods

Data Reduction Methods	Implementation	Application	Reduction Dimensions
[20]	SQL	Exact and Approximate Reduction	Database attributes
[21]	SQL	Exact reduction	Database rows

2.6 Lukasiewicz Implication

Data reduction methods are used to reduce the size of the database without losing knowledge. By knowledge, we mean the association rules of the database. In the literature, there have been many

data reduction methods implemented; however, most of these methods are not efficient, inaccurate and do not fit for fuzzy data [22]. The Lukasiewicz implication, which is based on a fuzzy Galois connection at different precision levels, is one of the efficient reduction methods used for reducing the size of objects or attributes of fuzzy or crisp formal context [22] .

“ Definition of a Lukasiewicz-Based Fuzzy Galois Connection: Let R be a fuzzy binary relation defined on U for two sets A and B such that $A \subseteq O$, B is a fuzzy set defined on P and $\delta \in [0,1]$ ” [22].

The operators f and h_δ are Lukasiewicz-based Galois connections and are defined as follows [22]:

“ $f(A) = \{d/\alpha \mid \alpha = \min \{\mu_R(g, d) \mid g \in A\}, d \in P\}$ ” (1) and

“ $h_\delta(B) = \{g \mid d \in P \Rightarrow (\mu_B(d) \rightarrow_L \mu_R(g, d)) \geq \delta\}$ ” (2).

Where “ \rightarrow_L ” represents the “Lukasiewicz implication” for $a, b \in [0,1]$, “ $a \rightarrow_L b = \min(1, 1-a+b)$ ”, $\mu_R(g, d)$ is the weight of attribute d in object g in the fuzzy relation R , and $\mu_B(d)$ represents the weight of attribute d in fuzzy set B . $f(A)$ computes the fuzzy set of the common properties of the objects A . $h_\delta(B)$ represents the object sets that satisfy all properties in B at a given level. The two operators f and h_δ are the fuzzy Galois connection between the subsets A and B [22].

The Lukasiewicz reduction algorithm consists of the following steps [22]:

1. For each object in the (FC), a set of verifying objects is calculated by using the Lukasiewicz implication $h_\delta(B)$.
2. The functional dependencies are maintained by computing $f(A)$ on the set of the verifying object and comparing it with the candidate object.

Example: Consider the fuzzy binary relation R in Table 7.

Table 7: Fuzzy Binary Relation R [22].

	A	B	C
O1	0.5	0.7	1.0
O2	0.2	0.3	0.4
O3	0.1	0.2	0.4
O4	0.4	0.3	1.0
O5	0.1	0.2	0.7
O6	0.2	1.0	0.4

After applying the Lukasiewicz fuzzy reduction at a different precision level, we obtain the following reduced relation, as shown below in Table 8.

Table 8: Reduced Database for Different Values of δ [22].

	$\delta = 0.9$	$\delta = 0.7$	$\delta = 0.5$
Remaining Objects	{O1,O4, O5, O6}	{O1, O5, O6}	{O5, O6}

In the next chapter, the Lukasiewicz reduction is used to find another algorithm for reducing database instance, preserving functional dependency.

2.7 Deep Web or Hidden Database

Deep web content represents a huge amount of structured data on the web. By the term “deep web” or “hidden database,” we mean the data content that is hidden behind HTML forms [23]. To retrieve data or information from the deep web, which is represented by the expression “crawling deep web,” the user enters a valid query on the form provided by the organization or institute, and then the deep web returns the result that matches the query.

Crawling deep web content is used in many tasks such as data integration and web indexing [24]. Two methods of accessing deep web content have been studied and implemented in the literature [23]. The idea of the first method is to build for each specific domain a search engine such as books, mobiles, etc. To implement this method, a mediator form was developed for each domain, and then each mediator form is mapped to its data source individually. By mediator form, we mean the form that is provided by any mediator system [25], which is a system that provides the user with access to information from heterogeneous resources [25]. However, implementing this method has several disadvantages, such as the high cost of creating the mediator form and the mapping to its data sources. In addition, matching each input query to its domain is considered a challenging problem. Moreover, defining data on different domains is not easy, since the nature and the boundaries of the data on the web are not clear [23].

The second method is implemented using “surfacing,” which relies on pre-computing the results of each form submission for all forms. Many algorithms were implemented in the literature to surface the deep web content. Computing the resulting URLs is done offline and indexed the same way as an HTML page. Using this approach proved very beneficial in leveraging the search engine infrastructure for the deep web content. Whenever a user selects a search result, the fresh content of the relevant website is directed to him or her [23].

Deep web sites or hidden web sites' contents can be categorized into two categories: document-oriented textual contents and entity-oriented content [24]. The document-oriented textual content represents popular websites such as Wikipedia, Twitter, etc. The entity-oriented content of deep web can be represented by online shopping websites such as eBay, Amazon, etc. Each deep web category has different crawling techniques or algorithms. The following is a description of an entity-oriented crawl system [24] that is specified to crawl entity-oriented deep web content and use it for advertisement purposes. Each structured entity in such deep web content represents a specified product. The objective behind implementing this system is to get the representative coverage of a specified item for a user. This system consists of the following main components, as illustrated in Diagram 2: URL template generation, query generation and URL generation, empty page filter (or web page filter) and URL extraction and deduplication [24].

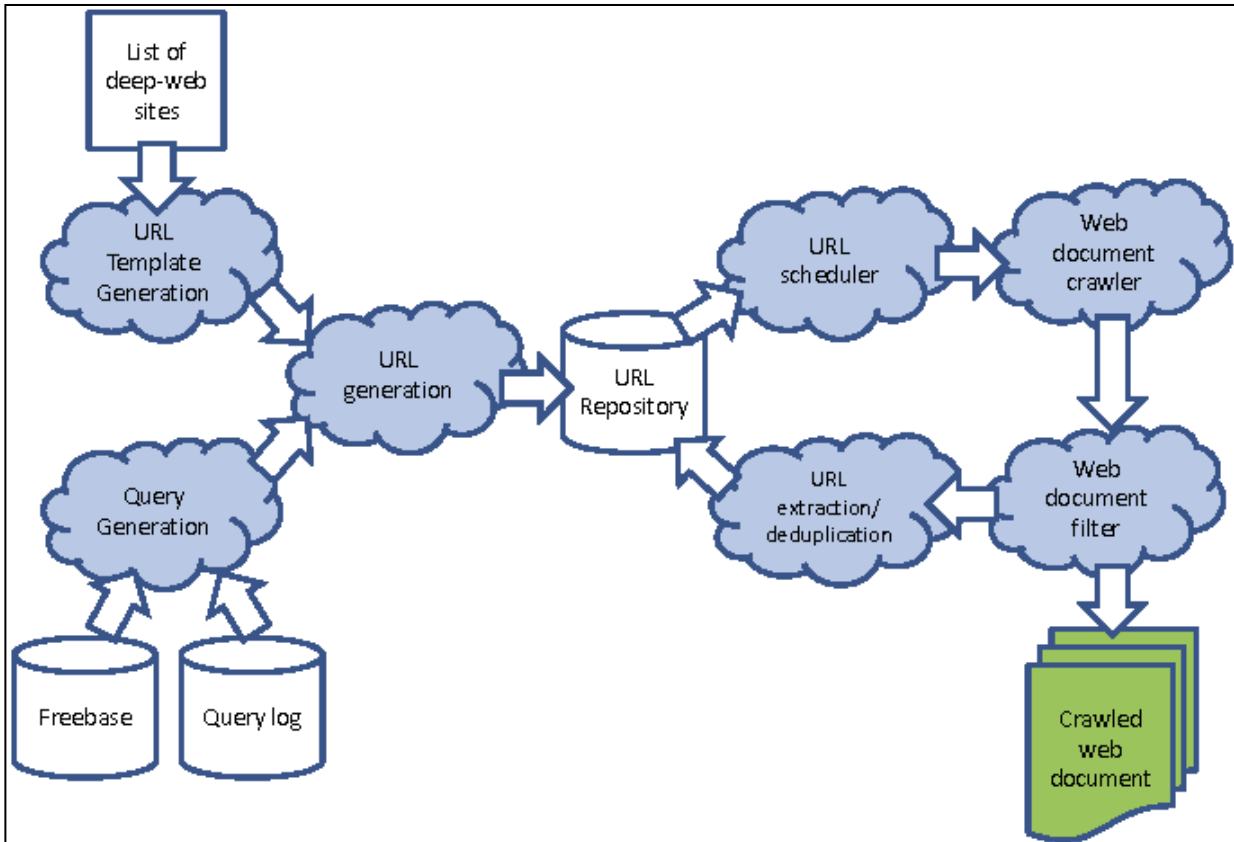


Figure 2 : Overview of the Entity-Oriented Crawl System [24]

- **“URL Template Generation”** : This component produces a list of URL templates after taking the domain name of the deep web sites as an input [24].
- **“Query Generation and URL Generation “**: The query generation component is used to generate a query after getting the input from the freebase and the query log. After that, the generated query will be input to the URL generation for getting the final query that will be stored in the URL repository [24].
- **“Empty Page Filter”**: This component is used to detect the retrieved pages that contain no entities or errors. This component is considered critical for most entity-oriented deep web sites [24].

- **“URL Extraction and Deduplication”**: This component is used to remove the second-level URLs, which are the URLs on the search results pages and different from the URL generated from the URL template and these URLs will not lead to deep web content [24].

In [26], a hidden web crawler was built for document-oriented deep web content. The crawler built was automatically generating query for indexing and discovering hidden web pages. To implement its main task, the crawler has to perform these steps: First, it generates a query .Second, it is delivered to the web site and finally, the required pages are obtained [26].

The most critical task performed in the crawling algorithm is choosing the relevant query word that will obtain the required user pages. Many options were provided for choosing relevant queries. Query words can be chosen randomly from an English dictionary or by using the most frequent keywords after analyzing a corpus of collected documents. Another “adaptive” method is used for selecting the query depending on analyzing the results obtained from a previous query and estimating the promising query that will return a matching result [26].

2.8 Online System Interfacing

Developing a mobile application with an efficient and easy-to-use interface is considered a complicated task. Many researchers found that the user interface code (UI) of the mobile application makes up about 80% of the whole application, and it takes about 50% of the code implementation time [27]. Thus, developing the UI of an application is considered a critical task for the implementation of the application.

A mobile phone device consists of the following main components: user interface (UI) system, operating system and many other hardware devices [27]. The UI system takes an input from the user and executes a corresponding output result. In the market, there are many tools available for creating mobile UI systems such as eMbedded Visual C++, Rapid and Symbian's Eclipse tool. While implementing and designing UIs for mobile applications, some problems could face the programmers, as follows [27]:

- writing appropriate program code;
- the cooperation between the UI designer and the UI programmer; and
- the lack of a UI design template generator in the current mobile application.

To overcome the previously mentioned problems, a UI design template generator was implemented and designed in [27] for mobile application that includes the UI template, the UI template constructor, the UI template manager, the UI template constructor and a UI template database.

2.9 Interfacing Small Screen Devices

There has been much research in the literature on developing an automatic interface for mobile computing devices. In fact, it is not an easy task to display the whole personal computer (PC) interface on one mobile device due to the small size of the mobile phone screen. Many techniques have been proposed to solve the interface division problem. One solution is to use interface tailoring, which adapts only important information and displays it in the interface [28]. In [3], an approach was developed to solve the interface division problem using two methods, which are depth-first and breadth-first principles. The model used was based on a constraint-based PC user

interface description method, and the design process had many stages, as illustrated in the below figure (3)[3] .

Figure 3: The Design Process of a Mobile Device Interface [3]

1- “Design common interface using representation model”	2- “Convert to common constraint model using constraint”	3- “Add constraints in mobile environment and get mobile constraint model”	4- “Divide interface into screens according to principles”	5- “Generate and display every interface of mobile device “
---	--	--	--	---

A main step in developing the mobile interface is to use the constraint-based user interface design method to describe the interface. Two types of constraints are contained in this model, which are “abstract constraints” and “spatial constraints”. “Abstract constraints include logical constraints, dependent constraints and geometric constraints” and are used for describing the logical relation of the interface components. Spatial constraints are used to describe the positional relation of the interface components [3]. The mechanism of the constraint interface relies on the abstract constraint and grouping the interface components into a component tree to describe the user interface information.

The component tree of the mobile interface can be divided using two approaches, which are the depth-first approach and the breadth-first approach. The tree non-leaf node represents component groups, while the leaf nodes represent concrete components. In the depth-first approach, all of the leaf nodes that have the same parent nodes are displayed on the same screen. If there is a need to come back to a displayed leaf node, then all the screens that are between the current screen and

the screen that contained that node are displayed in reverse order [3]. In the breadth-first approach, all of the child nodes of the root node are displayed first, and then the interface waits for a user action. If the action occurs, then the interface will display the entire child node of the chosen node. When a user needs to come back from a screen, the interface will display the parent node of the current node and its sibling [3].

A technique implemented in [39] for improving the interface of online system on small screen device. This technique is based on system decomposition and conceptual browser that: allows extracting main concepts from hidden data, structures search results in a tree view and allows users to move between tree nodes to reach search goal. On the other hand, this technique is not considering functional dependency preservation or data reduction that are targeted in our methodology.

The previous background and related work will be the foundation for constructing our methodology for the enhancement of the interface in chapter 3.

Chapter 3: Data Reduction by Functional Dependencies Preservation: A New Conceptual Approach

Interfacing online systems requires an analysis of real-time search data results and utilization of reduction methods to initially display only a sample of the most representative data. Sampling methods might offer a good alternative for that [29][30]. In this chapter, we describe an original approach for creating a conceptual sample from data (i.e., reduced database), which can preserve functional dependencies between the different attributes existing in the initial search result. To assess the quality of the reduced database, we select a benchmark of a database of objects, and we check the prediction accuracy of the reduced database. This chapter describes in detail the conceptual approach used in this thesis towards achieving our objectives and goal, which consists of converting a database into an approximate formal context, applying an incremental reduction process and extracting functional dependency or mapping the formal context back into a database instance, as shown below in Figure 4:

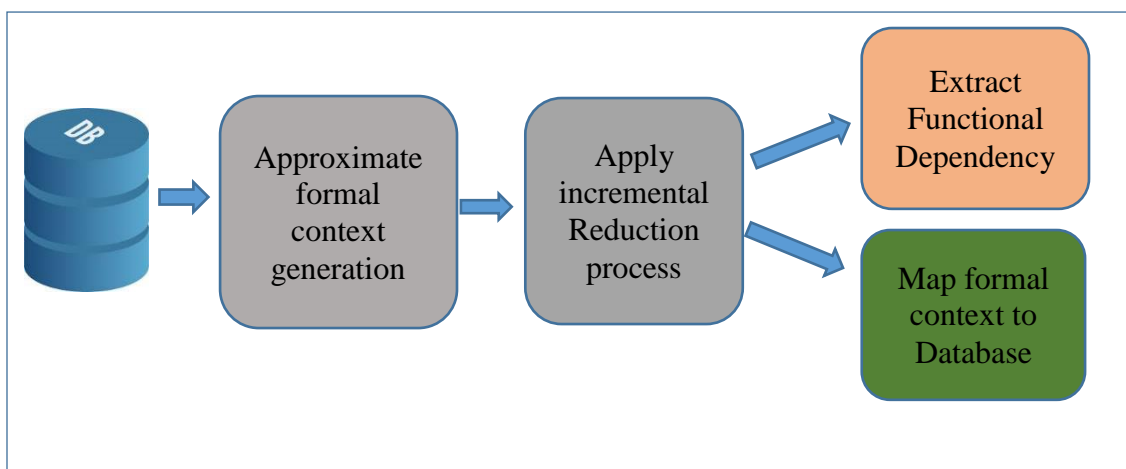


Figure 4: Conceptual Method


3.1 Transform DB into an Approximate Formal Context

The transformation is done using a pairwise comparison between the tuples of the obtained database [6]. In the exact formal context analysis, a value of “1” is assigned whenever there is an exact match between the same values of an attribute for the compared tuples; otherwise, a value of “0” is given. As exact equality is too restrictive, so we will propose different similarity functions to consider approximate functional dependencies.

Example: Figure (5) represents an exact transformation of a database into an exact FC.

Each row in the FC represents a possible pattern of zeros (i.e. disagreements) or ones (i.e. agreements) between two pairs of tuples of the initial dataset (i.e. instance of a database).

ID	A	B	C	D
t ₁	2	1	6	2
t ₂	3	1	5	1
t ₃	2	3	5	1
t ₄	3	2	5	2



ID	A	B	C	D
t ₁ -t ₂	0	1	0	0
t ₁ -t ₃	1	0	0	0
t ₁ -t ₄	0	0	0	1
t ₂ -t ₃	0	0	1	1
t ₂ -t ₄	1	0	1	0
t ₃ -t ₄	0	0	1	0

Figure 5: Converting a Database into a Formal Context (FC)

According to [6], it was proved that the functional dependencies (FDs) extracted from the original database are equivalent to the set of implications that we could extract from the formal context (FC).

The exact transformation into a FC results in a significant loss of information due to the exact comparison of the tuples value, thus we use in this thesis an approximate FC using similarity measures [31] to avoid such information loss. There are many similarity measure functions, which can be categorized based on the checked data to textual, numerical and binary forms [31].

In our case, we use the similarity measure between numerical and textual values. In the numerical value, the similarity between two numbers is calculated based on the difference between two compared numbers that are given a reference number, as shown in the following formula [31]:

$$\text{“Similarity } (n_1, n_2) = [1 - |n_1 - n_2| / \max(n_1, n_2)]\text{” (2),}$$

Where n_1 and n_2 represent the two compared numbers.

The similarity between two textual data is calculated using the algorithm proposed by Simon White [32]. Its functionality is based on comparing adjacent pairs, as in the following formula [31]:

$$\text{“Similarity } (S_1, S_2) = 2 \times |\text{pairs } (S_1) \cap \text{pairs } (S_2)| / |\text{pairs } (S_1) + \text{pairs } (S_2)|\text{” (3),}$$

Where S_1 and S_2 represent two strings.

Example: Consider the two words “Italy” and “Italian.” We divide them into pairs of string as follows: {IT, TA, AL, LY}, {IT, TA, AL, LI, IA, AN}. Then, the similarity between the two strings is calculated using the above similarity formula, as follows:

$$\begin{aligned} \text{Similarity (ITALY,ITALIEN)} &= 2 \times |(IT,TA,LA)| / (|\{IT,TA,AL,LY\}| + |\{IT,TA,AL,LI,IE,EN\}|) \\ &= 2 \times 3 / 4 + 6 = 0.6 \end{aligned}$$

At the step of transferring the data set into an approximate FC, the similarity was calculated in the tuples comparison process using formulas 2 and 3, so that the FC is assigned a value of “1” if their similarity is greater than a certain threshold (in our case, a threshold of 85 was chosen) [33]. Otherwise, “0” is assigned.

Very relevant to our research, in [4], the authors proved that applying the Lukasiewicz reduction to the FC might reduce it with the elimination of many objects without losing implications between

the different attributes of the FC. In the next section, we propose an efficient incremental approach to reduce the FC.

3.2 Apply Incremental Reduction to FC

Data reduction methods play a vital role in FCA due to the large obtained objects in the context, which negatively imposes excessive time and storage needs. The Lukasiewicz reduction [22] is implemented in a new way in this proposed conceptual method, which is incremental on packages of a certain number of FC objects. Thus, this leads us to avoid having an FC with n^2 objects compared to the number of database objects and waiting until the entire formal context is build. Another advantage of implementing the Lukasiewicz reduction incrementally is that it enables the dynamic reduction of formal context as long as new objects are available [22].

After that, the Lukasiewicz reduction was further applied iteratively in the obtained reduced FC to further reduce its size in terms of objects and attributes without loss (attribute implications that are equivalent to functional dependencies). In fact, the reduction is implemented in the approximate FC iteratively multiple times for columns and rows until reaching stability, which means that no more columns or rows can be reduced. The precision level used was $\delta = 1$ to guarantee information preservation while reducing rows and attributes [22].

The steps of the Lukasiewicz reduction algorithm (discussed previously in chapter 2) were applied as follows [22]:

1. For each object in the approximate FC, a set of verifying objects was calculated by using Lukasiewicz implication $h_{\delta}(B)$ (previously defined in chapter 2).

2. The functional dependency was maintained by computing $f(A)$ (previously defined in chapter 2) to the set of verifying objects and comparing it with the candidate object.

The following example will illustrate how to apply the Lukasiewicz reduction algorithm to the FC to obtain a reduced FC.

Example: Apply the Lukasiewicz reduction algorithm at $\delta = 1$ in the following FC Table 9 to reduce both objects and attributes.

Table 9: Formal Context (FC)

	A	B	C	D
O1	1	1	0	1
O2	1	1	0	1
O3	0	0	0	1
O4	0	1	1	1

We will start by applying the algorithm for O1 and then calculate its set of verifying objects. To calculate the sets of verifying objects, we compute the "Lukasiewicz implication $a \longrightarrow_L b$ as $\text{Min}(1, 1-a+b) \geq \delta$ "for each attribute of the compared object. Table 10 will illustrate the computation.

Table 10: Calculating $h_{\delta}(\mathbf{B})$

	O1-O2	O1-O3	O1-O4
A	Check $\text{Min}(1, 1 - a_1 + a_2) \geq 1$	Check $\text{Min}(1, 1 - a_1 + a_3) \geq 1$	Check $\text{Min}(1, 1 - a_1 + a_4) \geq 1$
	Result $\longrightarrow 1 = 1$	Result $\longrightarrow 0 < 1$	Result $\longrightarrow 0 < 1$
		The implication values is less than 1 so, we stop	The implication values is less than 1 so, we stop
B	Check $\text{Min}(1, 1 - b_1 + b_2) \geq 1$		
	Result $\longrightarrow 1 = 1$		
C	Check $\text{Min}(1, 1 - c_1 + c_2) \geq 1$		
	Result $\longrightarrow 1 = 1$		
D	Check $\text{Min}(1, 1 - d_1 + d_2) \geq 1$		
	Result $\longrightarrow 1 = 1$		

For both objects 2 and 3, we stop on attribute A because the value of the Lukasiewicz implication is less than 1, so only O2 is in the set of verifying objects for O1. The next step is to calculate the minimum using $f(A)$ as shown in Table 11 and compare it to the O1; if O1 is not less than the minimum, then it can be removed.


Table 11: Computing f(A)

	A	B	C	D
O1	1	1	0	1
minimum	1	1	0	1

Since O1 is not less than the minimum, it can be eliminated from the formal context.

After applying the Lukasiewicz algorithm to the remaining objects, we obtain the following reduced context, as shown below in Figure 6:

	A	B	C	D
O1	1	1	0	1
O2	1	1	0	1
O3	0	0	0	1
O4	0	1	1	1



	A	B	C	D
O2	1	1	0	1
O3	0	0	0	1
O4	0	1	1	1

Figure 6: Converting FC into Reduced FC

The above algorithm was implemented for reducing FC objects, and the same procedure was used to reduce FC attributes after transposing the formal context (i.e. attributes become objects and vice versa).

3.3 Extracting Functional Dependencies

In this section, we explain how we could extract functional dependencies from the dataset of search results, as we need these to define some order between attributes. As the FDs in the initial dataset are equivalent to the implications in the corresponding FC, we have advantageously used a tool (ConExp) [34] to indirectly find FDs.

The functional dependencies were extracted from the approximate reduced formal context. This process is done using the ConExp tool, which is an Open Source Java application used to extract different dependencies that exist between the attributes of the FC [34]. The ConExp tool [34] provides the following functionalities for the user: context editing, building concept lattices for a context, extracting attribute implications and association rules that are applicable in the context and attribute exploration. In our case, we will use the approximate reduced context obtained in the previous step as an input to the tool for extracting attribute implications that are equivalent to the functional dependencies in the original database [6] .

Example: Consider the following formal context (FC) in ConExp, as shown below in Figure 7.

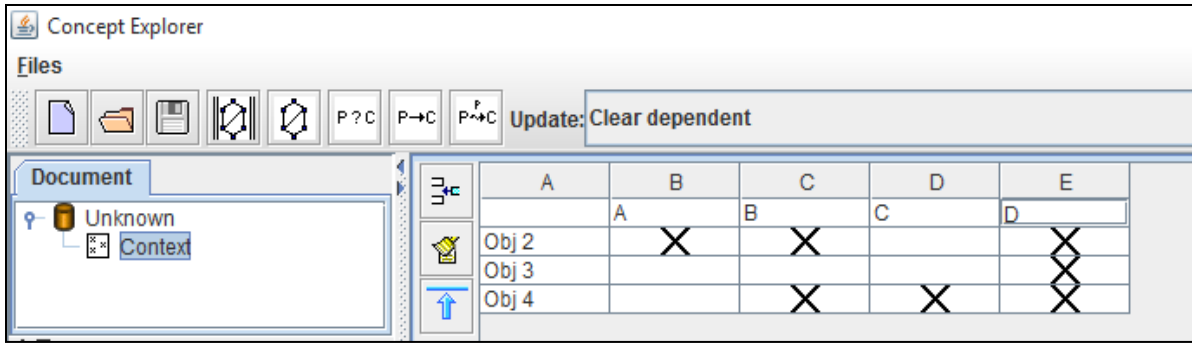


Figure 7: FC in ConExp

Using this tool, we extract the attribute implications, as shown below in Figure 8.

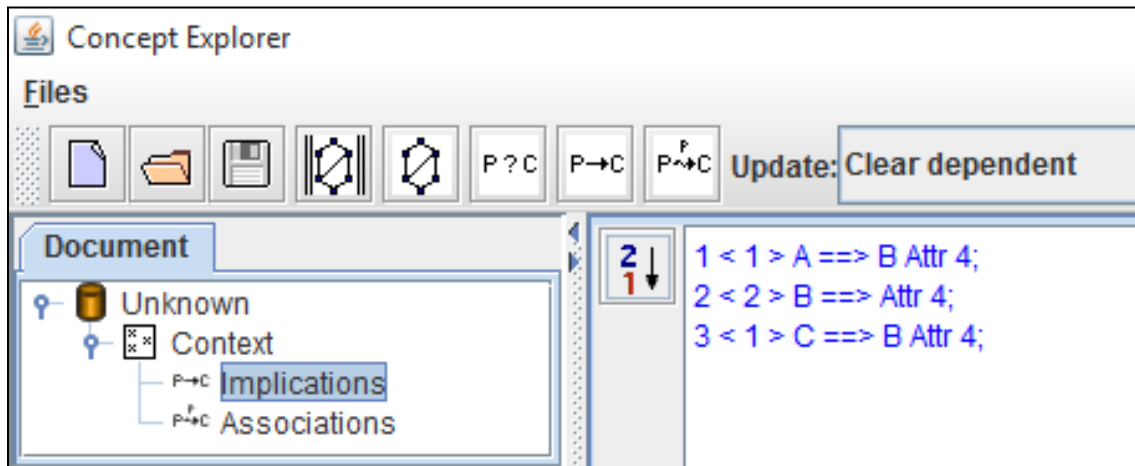


Figure 8: Extracting Attribute Implications using ConExp

By extracting the functional dependencies between the attributes of the hidden database, we will display the most important attributes (dependent attributes) in the interface of the online system.

3.4 Map a Formal Context into a Database

The last step in the proposed conceptual method is to map back the reduced, approximate formal context into a database instance, which will result in a compact version of the original data set. The resulting database preserves all the data characteristics of the original database and will be used for displaying the output result in the improved interface system, which will be discussed in detail in chapter 4.

Figure 9 displays the algorithm that maps a formal context into a database instance:

Algorithm : Convert a Formal Context into a Database Instance
Input: Reduced Formal Context, Original Database R
Output: Reduced Database Instance RD
Begin
For each object in the reduced formal context, which represents a pair of objects in the database, do the following:
1- Compare the formal context object with the original database object.
2- If there is a match and it is not on the RD, put the object in RD.
End for
End

Figure 9: Algorithm to Map Formal Context into a Database Instance.

Example : Consider the following dataset that consists of 100 objects and 11 attributes [37], as shown below in Figure 7:

1	att1	att2	att3	att4	att5	att6	att7	att8	att9	att10	att11
2	1000025	5	1	1	1	2	1	3	1	1 b	
3	1002945	5	4	4	5	7	10	3	2	1 b	
4	1015425	3	1	1	1	2	2	3	1	1 b	
5	1016277	6	8	8	1	3	4	3	7	1 b	
6	1017023	4	1	1	3	2	1	3	1	1 b	
7	1017122	8	10	10	8	7	10	9	7	1 m	
8	1018099	1	1	1	1	2	10	3	1	1 b	
9	1018561	2	1	2	1	2	1	3	1	1 b	
10	1033078	2	1	1	1	2	1	1	1	5 b	
11	1033078	4	2	1	1	2	1	2	1	1 b	
12	1035283	1	1	1	1	1	1	3	1	1 b	
13	1036172	2	1	1	1	2	1	2	1	1 b	
14	1041801	5	3	3	3	2	3	4	4	1 m	
15	1043999	1	1	1	1	2	3	3	1	1 b	
16	1044572	8	7	5	10	7	9	5	5	4 m	
17	1047630	7	4	6	4	6	1	4	3	1 m	
18	1048672	4	1	1	1	2	1	2	1	1 b	
19	1049815	4	1	1	1	2	1	3	1	1 b	
20	1050670	10	7	7	6	4	10	4	1	2 m	
21	1050718	6	1	1	1	2	1	3	1	1 b	
22	1054590	7	3	2	10	5	10	5	4	4 m	
23	1054593	10	5	5	3	6	7	7	10	1 m	
24	1056784	3	1	1	1	2	1	2	1	1 b	
25	1057013	8	4	5	1	2	1	7	3	1 m	
26	1059552	1	1	1	1	2	1	3	1	1 b	
27	1065726	5	2	3	4	2	7	3	6	1 m	
28	1066373	3	2	1	1	1	1	2	1	1 b	
29	1066979	5	1	1	1	2	1	2	1	1 b	
30	1067444	2	1	1	1	2	1	2	1	1 b	

Figure 10: The Input Datasets of the Conceptual Method

After applying the conceptual method on the dataset (i.e. running the developed software on that dataset), as shown below in Figure 11, that is, transferring the dataset into an approximate reduced FC (using 70% approximation level and $\delta = 1$), then iteratively applying the Lukasiewicz reduction on objects and attributes and transferring the FC back into a database instance, we get the following reduced dataset with only 29 objects, as shown in Figure 11 :

1	att1	att2	att3	att4	att5	att6	att7	att8	att9	att10	att11
2	1000025	5	1	1	1	2	1	3	1	1	b
3	1002945	5	4	4	5	7	10	3	2	1	b
4	1015425	3	1	1	1	2	2	3	1	1	b
5	1016277	6	8	8	1	3	4	3	7	1	b
6	1017023	4	1	1	3	2	1	3	1	1	b
7	1018561	2	1	2	1	2	1	3	1	1	b
8	1033078	4	2	1	1	2	1	2	1	1	b
9	1044572	8	7	5	10	7	9	5	5	4	m
10	1048672	4	1	1	1	2	1	2	1	1	b
11	1049815	4	1	1	1	2	1	3	1	1	b
12	1050718	6	1	1	1	2	1	3	1	1	b
13	1057013	8	4	5	1	2	1	7	3	1	m
14	1059552	1	1	1	1	2	1	3	1	1	b
15	1065726	5	2	3	4	2	7	3	6	1	m
16	1066373	3	2	1	1	1	1	2	1	1	b
17	1071760	2	1	1	1	2	1	3	1	1	b
18	1080185	10	10	10	8	6	1	8	9	1	m
19	1096800	6	6	6	9	6	1	7	8	1	b
20	1103722	1	1	1	1	2	1	2	1	2	b
21	1105524	1	1	1	1	2	1	2	1	1	b
22	1110524	10	5	5	6	8	8	7	1	1	m
23	1111249	10	6	6	3	4	5	3	6	1	m
24	1112209	8	10	10	1	3	6	3	9	1	m
25	1115282	5	3	5	5	3	3	4	10	1	m
26	1118039	5	3	4	1	8	10	4	9	1	m
27	1143978	5	2	1	1	2	1	3	1	1	b
28	1147748	5	10	6	1	10	4	4	10	10	m
29	1148278	3	3	6	4	5	8	4	4	1	m
30	1166630	7	5	6	10	5	10	7	9	4	m

breast_100_R

Figure 11: Compact Version of DB after Running the Conceptual Method

The fuzzy formal concept analysis was not used in the design of the proposed conceptual method .Although the validation of the conceptual method was done using fuzzy formal context and crisp formal context as shown on [33].

3.5 Complexity Analysis

Complexity analysis is calculated to all processes of the developed conceptual method, which are transforming the database into approximate formal context, apply incremental reduction, and extract functional dependency or map back the formal context into a reduced database.

1. Transforming the dataset into approximate formal context and apply incremental reduction:

Let have a relation r that contains n objects and m attributes, and then the n objects will iterate n times for comparing the attribute values which makes the complexity $O(n^2 * m)$

. The incremental reduction is applied on packages of fixed size formal context objects (10 objects) while transferring the database into the approximate formal context .The complexity of reduction is $O(n^4 * m)$. However, in reality the time complexity is much better with the usage of the incremental reduction as observed for different datasets.

2. Extracting Functional Dependency:

The functional dependencies were extracted using ConExp tool .Therefore, the best case for building the concept lattice and extract the functional dependencies is polynomial due to incremental reduction which speeds the process of mining functional dependencies while the worst case is exponential[40] .

3. Map the formal context into reduced database:

All pair of objects in the formal context were compared with the database objects in the attributes values in order to map back the formal context into a reduced database.

Therefore, the complexity of this process at the best case is polynomial and at the worst case is equal to $O(n^2 * m)$.

The total complexity of the proposed conceptual method at the best case is polynomial and at the worst case is exponential.

3.6 Validation of the Conceptual Method

An experiment was conducted on “the Wisconsin Breast Cancer database of the UCI machine-learning repository “ to validate the adopted conceptual method [33].This database consists of 699

tuples and 10 attributes. Samples of different sizes like 100, 200, 300 and 350 tuples were taken from this dataset to be an input for the conceptual method, as shown below in Table 12[33].

Table 12: Experiment with Different Sample Sizes and Approximation Levels on the Conceptual Method [33]

Sample Size	FC Size Before	Approximation	FC Size After	Database	Size
	Reduction	Level	Reduction	After Mapping	
100	4950	70 %	24	18	
		80%	34	19	
		90%	38	18	
200	19900	70%	26	24	
		80%	42	24	
		90%	39	24	
300	44850	70%	22	34	
		80%	41	37	
		90%	45	34	
350	61075	70%	22	33	
		80%	43	37	
		90%	48	32	

It is clear that the conceptual method has reached a remarkable output; for example, an FC with size 4950 at an approximation level of 70% was compacted to 24 objects. The precision reduction was set to “1” during the entire experiment to prevent loss of information .

After that, these reduced datasets are used as training examples to train a classifier that uses a machine learning algorithm. The classifier was built using an artificial neural network algorithm, which is a classic algorithm for handling complex problems[35]. Actually, the datasets were split in percentage, which means that the first split is used for testing and the remaining one is used for evaluating. Three evaluation metrics were used in the datasets, which are classification accuracy, root mean square error (RMSE) and reduced data size, as shown below in Table 13[33].

Table 13: Evaluating the Dataset Using Three Evaluation Metrics [33].

Sample Size	Approximation Level	Classification Accuracy	RMSE
100	70 %	94.9%	0.19
	80 %	94%	0.20
	90 %	95%	0.18
200	70 %	78.4%	0.44
	80 %	85.4%	0.35
	90 %	89.4%	0.30
300	70 %	95.7%	0.19
	80 %	96.3%	0.18
	90 %	95.3%	0.20
350	70 %	95.1%	0.20
	80 %	97.7%	0.15
	90 %	89.1%	0.30

From the above table, it is observed that for a sample size of 100 objects, the classification accuracy is almost the same at a different approximation level due to the same reduced number of objects

obtained at a different approximation level, as shown previously in Table 3. The worst classification accuracy was obtained when the sample size was 200, due to the unbalanced reduced objects fed as a training set to the classifier. It can be noticed that there is a strong relation between the classification accuracy and the training set size and class balance.

From the above experiment, it was proved by using a real test case that our proposed conceptual method, which is based on data reduction, achieves a highly accurate result, which shows its efficiency in being used for summarizing dataset results by first displaying only a sample of the most representative data in an online system interface, as explained in chapter 4

Chapter 4: Methodology and Implementation

In this chapter, we explain in detail the utilization of the conceptual method described previously in chapter 3 for designing the proposed solution that consists mainly of two stages for improving the online system interface: offline static data analysis and real time data analysis.

4.1 Offline, Static Hidden-Data Analysis

Most online systems do not provide direct access to their databases. Therefore, the process of offline static data analysis is essential in this thesis to discover the domains of their hidden databases in terms of number of attributes, attribute cardinality and the semantic relation between the attributes (functional dependencies). In addition, the discovered knowledge from this step is essential for enhancing the results displayed to the user. Usually, hidden databases or deep web enable users and researchers to interact with their databases and services using application programming interfaces (APIs).

The process of static hidden-data analysis is conducted offline on the datasets obtained after running an API call for a specific item. Then, the developed conceptual method discussed previously in chapter 3 is applied on several datasets for the domain discovery knowledge and to extract the functional dependency. Figure 12 explains the process of offline static data analysis.

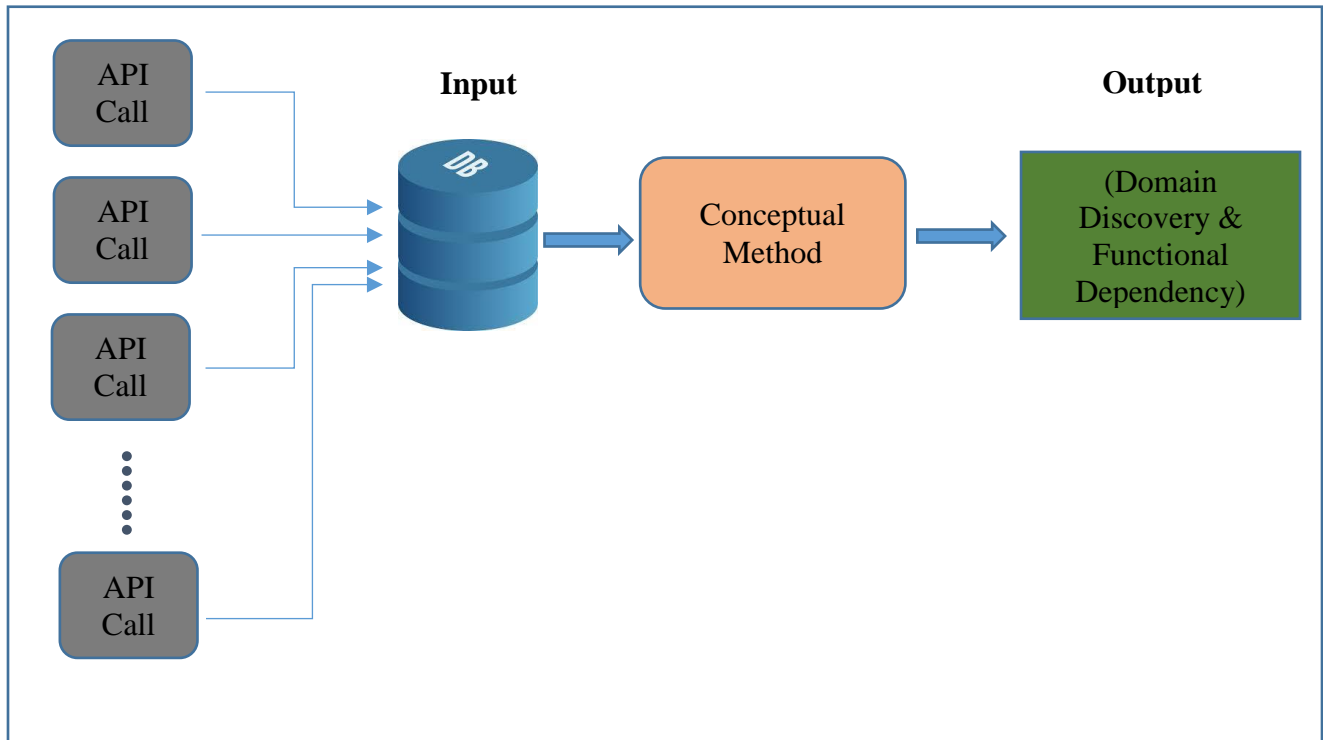


Figure 12: Offline Static Data Analysis

In order to use the API provided for a hidden database, in our case the eBay database, we register for an API authentication using the eBay developer website. Once the API request is sent for a specific item, the returned results are in the form of an XML file. An API request is sent using a uniform resource locator (URL) and the GET Method provided by the Hypertext Transfer Protocol (HTTP). Figure 13 is an example of an eBay API call.

```
http://svcs.ebay.com/services/search/FindingService/v1?OPERATION-  
NAME=findItemsByKeywords&SERVICE-VERSION=1.0.0&SECURITY-  
APPNAME=QatarUni-4b86-4755-a046-f7374027425b&RESPONSE-DATA-  
FORMAT=XML&REST-PAYLOAD&keywords=oracle%206
```

Figure 13: API Call

Each API call consists of the following parts, as shown in Table 14:

Table 14: API Call Parts

“ http://svcs.ebay.com/services/search/FindingService/v1? ”	→	End point URL for the API request
“OPERATION-NAME	→	findItemsByKeywords”
“SERVICE-VERSION	→	1.0.0”
“SECURITY-APPNAME	→	QatarUni-4b86-4755-a046-f7374027425b”
“RESPONSE-DATA-FORMAT	→	XML”
“keywords	→	required item”

After sending an API request, the response file will be in the XML format, as shown in Figure 14:

```

▼<findItemsByKeywordsResponse xmlns="http://www.ebay.com/marketplace/search/v1/services">
  <ack>Success</ack>
  <version>1.13.0</version>
  <timestamp>2016-04-25T09:41:29.803Z</timestamp>
  ▼<searchResult count="100">
    ▼<item>
      <itemId>191851863354</itemId>
      ▼<title>
        SEALED - iPad Mini 4, MK6J2LL/A, 16GB, Space Gray, Apple Warranty, Free Shipping
      </title>
      <globalId>EBAY-US</globalId>
      ▼<primaryCategory>
        <categoryId>171485</categoryId>
        <categoryName>iPads, Tablets & eBook Readers</categoryName>
      </primaryCategory>
      ▼<galleryURL>
        http://thumbs3.ebaystatic.com/m/mubtxApn4i0tVsbL3Wf7tYw/140.jpg
      </galleryURL>
      ▼<viewItemURL>
        http://www.ebay.com/itm/SEALED-iPad-Mini-4-MK6J2LL-A-16GB-Space-Gray-Apple-Warranty-Free-Shipping-/191851863354
      </viewItemURL>
      <productId type="ReferenceID">216534447</productId>
      <paymentMethod>PayPal</paymentMethod>
      <autoPay>true</autoPay>
      <postalCode>10001</postalCode>
      <location>New York,NY,USA</location>
      <country>US</country>
      ▼<shippingInfo>
        <shippingServiceCost currencyId="USD">0.0</shippingServiceCost>
        <shippingType>FlatDomesticCalculatedInternational</shippingType>
        <shipToLocations>US</shipToLocations>
        <shipToLocations>CA</shipToLocations>
        <shipToLocations>GB</shipToLocations>
        <shipToLocations>AU</shipToLocations>
        <shipToLocations>AT</shipToLocations>
        <shipToLocations>BE</shipToLocations>
        <shipToLocations>FR</shipToLocations>
        <shipToLocations>DE</shipToLocations>
        <shipToLocations>IT</shipToLocations>
        <shipToLocations>JP</shipToLocations>
        <shipToLocations>ES</shipToLocations>
        <shipToLocations>TW</shipToLocations>
        <shipToLocations>NL</shipToLocations>
        <shipToLocations>CN</shipToLocations>
      </shippingInfo>
    </item>
  </searchResult>
</findItemsByKeywordsResponse>

```

Figure 14: XML Response for an API Call

Several API requests were sent for various random items such as iPhones, iPads, cars, perfumes, glasses, clothes, cameras and watches. Then, the resulting XML file was processed in the format of an XML table in an Excel worksheet for the domain discovery knowledge process and to extract functional dependency, as shown in Figure 15.

The figure displays two screenshots of API data in the form of Excel worksheets.

The top screenshot shows a list of API responses with the following columns: ns1:ack, ns1:version, ns1:timestamp, count, ns1:itemid, and ns1:title. The data rows show successful responses (Status: Success) with a count of 100 for each item, and a title describing the item, such as "SEALED - iPad Mini 4, MK6J2LL/A, 16GB, Space Gray, Apple Warranty, Free Shipping".

The bottom screenshot shows a list of eBay items with the following columns: ns1:globalid, ns1:categoryid, ns1:categoryName, ns1:galleryURL, and ns1:viewItemURL. The data rows show multiple instances of the same item (Global ID: 171485) categorized as "iPads, Tablets & eBook Readers", with the gallery URL and view item URL for each instance.

Figure 15: API Data in the Form of an Excel Worksheet.

The resulting datasets from an API call contain 56 attributes, such as itemId, title, category name, viewItemURL, etc. and approximately 100 rows depending on the search item. Then, the datasets are analyzed and many unnecessary attributes are eliminated. In addition, duplicated tuples are removed based on their IDs. At the end, we obtain a result that contains 11 attributes and around

100 unique objects for a specific item, as shown in Figure 16. The remaining attributes are mainly itemId, title, categoryId, categoryName, URL, image, price, country, location, shipping type and item condition. The attributes' values were unique numbers for the itemId, text value for title, serial number for categoryID, text value for the categoryName, URL links in the URL attribute, links to the product image in the image attribute, etc.

A	B	C	D	E	F
itemid	title	globalcat	categoryName	galleryURL	
281945378646	Factory Sealed Apple iPad mini 4 16GB, Wi-Fi, 7.9in - Gold (Latest Model)	EBAY-US	171485 iPads, Tablets & eBook Re	http://thumbs3.ebaystatic.com/m/mKa0omECmMAwUbg_ix5mwgg/1	
191812423053	SEALED, iPad Mini 4, WiFi, MK6L2LL/A, 16GB, Gold, Free Shipping	EBAY-US	171485 iPads, Tablets & eBook Re	http://thumbs2.ebaystatic.com/m/mkCbrUJO9LY4EH24A10Ssw/140.jp	
252298872226	New Apple iPad Mini 4 16GB Wi-Fi 7.9in Gold Sealed Overnight shipping available	EBAY-US	171485 iPads, Tablets & eBook Re	http://thumbs3.ebaystatic.com/m/mdXmr6u8M8zt-DN9M8Bupug/140.jp	
182032945223	Apple iPad mini 4 16 GB (Wi-Fi Only) 7.9in GOLD - NEW! FACTORY SEALED- MK6L2LL/A	EBAY-US	171485 iPads, Tablets & eBook Re	http://thumbs4.ebaystatic.com/m/mCQCQ0ERPnMMw10D6rYQvw/140.jp	
291639839189	BRAND NEW SEALED Apple iPad mini 4 16GB Wi-Fi - Gold (Latest Model)	EBAY-US	171485 iPads, Tablets & eBook Re	http://thumbs2.ebaystatic.com/m/mB2Nl0knXfB_Sy7Jb2mslug/140.jp	
291594527075	New Magnetic Smart Cover Leather + Back Case for Apple iPad 2 3 4 Air Mini 1 2 3	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs4.ebaystatic.com/pict/301675175408404000000010_1.jp	
400930651887	360 Rotating Stand Leather Smart Cover Case For Apple iPad 2/3/4 /mini/Air/ Air2	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs4.ebaystatic.com/pict/400930651887404000000006_4.jp	
301675175408	Shockproof Heavy Duty Rubber With Hard Stand Case Cover For iPad Air 2 iPad Mini	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs1.ebaystatic.com/pict/301675175408404000000006_5.jp	
252197017973	SHOCKPROOF HEAVY DUTY RUBBER HARD CASE COVER FOR APPLE IPAD 2 3 4 MINI AIR & PEN	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs2.ebaystatic.com/pict/252197017973404000000001_1.jp	
151774429848	Real Leather Nice Smart Case Stand Cover For Apple ipad 2 3 4 Air / ipad Mini	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs1.ebaystatic.com/pict/151774429848404000000001_1.jp	
262014243262	Top Quality genuine leather protective smart case cover For iPad 2 3 4 /mini/ Air	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs3.ebaystatic.com/pict/262014243262404000000001_1.jp	
111587284214	Luxury Bowknot Leather Smart Case Stand Cover for Apple iPad2 3 4 Air Air 2 mini	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs3.ebaystatic.com/pict/111587284214404000000001_1.jp	
391129480974	Shockproof Military Heavy Duty Rubber With Hard Stand Case Cover For Apple iPad	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs3.ebaystatic.com/pict/391129480974404000000015_1.jp	
121559790574	Ultra Slim Magnetic Leather Smart Stand Case Cover For Apple iPad 2 3 4 Air Mini	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs3.ebaystatic.com/pict/121559790574404000000002_3.jp	
251951939097	BESDATA Slim-Fit Magnetic Leather Smart Cover Back Case for iPad 4 3 2 Mini Air	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs2.ebaystatic.com/pict/251951939097404000000006_2.jp	
321573405752	For Apple iPad 2/3/4 Air Mini/Mini 2 360 Rotating PU Leather Case Cover Stand	EBAY-US	176973 Cases, Covers, Keyboard Fi	http://thumbs1.ebaystatic.com/pict/321573405752404000000012_6.jp	
351646751602	NEW Apple iPad Mini 4 MK6L2LL/A 7.9-Inch, 16GB, Wi-Fi, iOS 9, Sealed Gold	EBAY-US	171485 iPads, Tablets & eBook Re	http://thumbs3.ebaystatic.com/m/mB9c4KPd6LdFZBbc2UmLvvg/140.jp	

viewItemURL	location	count	shippingType
http://www.ebay.com/itm/Factory-Sealed-Apple-iPad-mini-4-16GB-Wi-Fi-7.9in-Gold-Latest-Model-/281945378646	Port Hueneme,CA,USA	US	FlatDomesticCalculatedInternational
http://www.ebay.com/itm/SEALED-iPad-Mini-4-WIFI-MK6L2LL-A-16GB-Gold-Free-Shipping-/191812423053	New York,NY,USA	US	FlatDomesticCalculatedInternational
http://www.ebay.com/itm/New-Apple-iPad-Mini-4-16GB-Wi-Fi-7.9in-Gold-Sealed-Overnight-shipping-available-/252298872226	Garner,NC,USA	US	Flat
http://www.ebay.com/itm/Apple-iPad-mini-4-16-GB-Wi-Fi-Only-7.9in-GOLD-NEW-FACTORY-SEALED-MK6L2LL-A-/182032945223	Brooklyn,NY,USA	US	Free
http://www.ebay.com/itm/BRAND-NEW-SEALED-Apple-iPad-mini-4-16GB-Wi-Fi-Gold-Latest-Model-/291639839189	Katy,TX,USA	US	Free
http://www.ebay.com/itm/New-Magnetic-Smart-Cover-Leather-Back-Case-Apple-iPad-2-3-4-Air-Mini-1-2-3-/291594527075?var=590666307705	USA	US	Free
http://www.ebay.com/itm/360-Rotating-Stand-Leather-Smart-Cover-Case-Apple-iPad-2-3-4-mini-Air-Air2-/400930651887?var=670474089522	Ontario,CA,USA	US	Free
http://www.ebay.com/itm/Shockproof-Heavy-Duty-Rubber-Hard-Stand-Case-Cover-iPad-Air-2-iPad-Mini-/301675175408?var=600521758636	Flushing,NY,USA	US	Free
http://www.ebay.com/itm/SHOCKPROOF-HEAVY-DUTY-RUBBER-HARD-CASE-COVER-APPLE-IPAD-2-3-4-MINI-AIR-PEN-/252197017973?var=551032290001	Hong Kong	HK	Free
http://www.ebay.com/itm/Real-Leather-Nice-Smart-Case-Stand-Cover-Apple-ipad-2-3-4-Air-ipad-Mini-/151774429848?var=50971798798	Hong Kong	HK	Free
http://www.ebay.com/itm/Top-Quality-genuine-leather-protective-smart-case-cover-iPad-2-3-4-mini-Air-/262014243262?var=560808350990	Rowland Heights,CA,USA	US	Flat
http://www.ebay.com/itm/Luxury-Bowknot-Leather-Smart-Case-Stand-Cover-Apple-iPad2-3-4-Air-Air-2-mini-/111587284214?var=410588102969	China	CN	Flat
http://www.ebay.com/itm/Shockproof-Military-Heavy-Duty-Rubber-Hard-Stand-Case-Cover-Apple-iPad-/391129480974?var=660552915959	Flushing,NY,USA	US	Free
http://www.ebay.com/itm/Ultra-Slim-Magnetic-Leather-Smart-Stand-Case-Cover-Apple-iPad-2-3-4-Air-Mini-/121559790574?var=420611249693	La Puente,CA,USA	US	Free
http://www.ebay.com/itm/BESDATA-Slim-Fit-Magnetic-Leather-Smart-Cover-Back-Case-iPad-4-3-2-Mini-Air-/251951939097?var=550820977784	Rowland Heights,CA,USA	US	Free
http://www.ebay.com/itm/Apple-iPad-2-3-4-Air-Mini-Mini-2-360-Rotating-PU-Leather-Case-Cover-Stand-/321573405752?var=510730248651	Pompano Beach,FL,USA	US	FlatDomesticCalculatedInternational
http://www.ebay.com/itm/NEW-Apple-iPad-Mini-4-MK6L2LL-A-7.9-Inch-16GB-Wi-Fi-iOS-9-Sealed-Gold-/351646751602	Gainesville,FL,USA	US	FlatDomesticCalculatedInternational

H	I	J	K	L	M	N
location	count	shippingType	shipToLocations	currentPrice	currencyId2	conditionDisplayName
Port Hueneme,CA,USA	US	FlatDomesticCalculatedInternational	US	334 USD		New
New York,NY,USA	US	FlatDomesticCalculatedInternational	US	334.89 USD		New
Garner,NC,USA	US	Flat	Worldwide	339.89 USD		New
Brooklyn,NY,USA	US	Free	US	339.99 USD		New
Katy,TX,USA	US	Free	US	349.99 USD		New
USA	US	Free	US	8.49 USD		New
Ontario,CA,USA	US	Free	US	9.8 USD		New
Flushing,NY,USA	US	Free	Worldwide	14.99 USD		New
Hong Kong	HK	Free	Worldwide	11.6 USD		New
Hong Kong	HK	Free	Worldwide	14.02 USD		New
Rowland Heights,CA,USA	US	Flat	US	15.19 USD		New other (see details)
China	CN	Flat	AU	8.45 USD		New
Flushing,NY,USA	US	Free	Worldwide	14.24 USD		New
La Puente,CA,USA	US	Free	US	8.16 USD		New
Rowland Heights,CA,USA	US	Free	US	9.57 USD		New
Pompano Beach,FL,USA	US	FlatDomesticCalculatedInternational	US	3.99 USD		New
Gainesville,FL,USA	US	FlatDomesticCalculatedInternational	US	339.95 USD		New

Figure 16: Data after Preprocessing the API Request.

After the domain discovery process, the dataset is converted into a CSV file format to become the input for the conceptual method, which will convert it into an approximate reduced FC, iteratively apply iteratively the Lukasiewicz reduction on FC rows and attributes and extract the functional

dependency using the ConExp tool (see the description of the conceptual method in chapter 3), as shown in Figure 17.


```

Reading dataset : Book1-IPAD.csv
Size of dataset : 16
Start conversion to approximated reduced FC
Threshold = 85.0
0,
Alpha : 1.0
STEP : 1
Reducing the Attributes of FC
No of reduced columns = 0
[]

```

```

Size of FC : 16
Building reduced DB from reduced FC
DB Objects : [1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16]
Size of reduced DB : 15
Writing reduced DB to file : Book1-IPAD_R.csv

```

	A	B	C	D	E	F	G	H	I	J	K	L
	itemId	title	categoryId	categoryName	galleryURL	viewItemURL	location	country	shippingType	currentPrice	conditionD...	
obj1-obj2				X	X				X	X		
obj1-obj10	X			X					X			X
obj2-obj3	X			X					X			
obj2-obj4			X	X					X	X	X	X
obj2-obj7	X		X	X			X	X	X	X	X	X
obj3-obj6				X					X	X	X	X
obj3-obj9	X			X					X	X	X	X
obj3-obj11				X					X	X	X	X
obj3-obj13				X					X	X	X	X
obj4-obj15			X	X			X	X	X	X	X	X
obj5-obj10				X					X	X	X	X
obj5-obj14	X			X					X	X	X	X
obj10-obj11				X		X			X	X	X	X
obj11-obj12				X				X	X	X	X	X
obj13-obj14				X		X		X	X	X	X	X
obj16-obj16	X	X	X	X	X	X	X	X	X	X	X	X

```

Update: Clear dependent
1 < 16 > {} ==> categoryId;
2 < 10 > categoryId shippingType ==> categoryName;
3 < 4 > title categoryId ==> categoryName location country shippingType currentPrice conditionDisplayName;
4 < 5 > itemId categoryId categoryName ==> conditionDisplayName;
5 < 2 > categoryId galleryURL ==> categoryName country conditionDisplayName;
6 < 3 > categoryId viewItemURL ==> title categoryName location country shippingType currentPrice conditionDisplayName;
7 < 7 > categoryId location ==> categoryName country conditionDisplayName;
8 < 9 > categoryId currentPrice ==> categoryName conditionDisplayName;
9 < 4 > itemId categoryId categoryName country conditionDisplayName ==> shippingType;
10 < 1 > categoryId categoryName galleryURL location country conditionDisplayName ==> itemId title viewItemURL shippingType currentPrice;
11 < 4 > itemId categoryId categoryName shippingType conditionDisplayName ==> country;
12 < 1 > categoryId categoryName galleryURL country shippingType conditionDisplayName ==> itemId title viewItemURL location currentPrice;
13 < 3 > itemId categoryId categoryName location country shippingType conditionDisplayName ==> title currentPrice;
14 < 5 > categoryId categoryName location country currentPrice conditionDisplayName ==> shippingType;
15 < 1 > categoryId categoryName galleryURL country currentPrice conditionDisplayName ==> itemId title viewItemURL location shippingType;
16 < 3 > itemId categoryId categoryName country shippingType currentPrice conditionDisplayName ==> title location;

```

Figure 17: Applying the Conceptual Method on the Dataset

After receiving and analyzing the implication sets using the ConExp tool, we observe that most attributes depend on the title attribute. Therefore, we use it in the interface to display the search results of a query to the user.

The functional dependencies extracted at the stage of offline data analysis from the different dataset (i.e. dataset coming from API request) are only the shared FD among the datasets and may change or updated if schema of the hidden database is updated.

4.2 Real Time Data Analytic

The objective of this step is to use data analytics to organize the online system interfacing by respecting some criteria of a good interface, such as first displaying a summary of the search results then repeating the same process on the remaining search results. The real-time data analysis stage is considered the essential stage in the implemented proposed solution, because all operations are done at that stage after the user issues a search query for a specific product. A real-time data analysis consists of the following activities:

1. Obtain the user query as input for the search product.
2. Use the eBay API to request the search product from the online marketplace (eBay).
3. Preprocess the resulting data by removing duplicated tuples.
4. Convert the data instance into an approximate reduced formal context and then map it back to a database instance to obtain a compact version of the search result.
5. Display the reduced data to the user.
6. A displayed further reduced dataset is available to the user upon his or her request by pressing on the next button on the screen.

7. When the user clicks on the desired record, the web browser will open the link of the desired product.

Figure 18 illustrates the different activities of real-time data analysis:

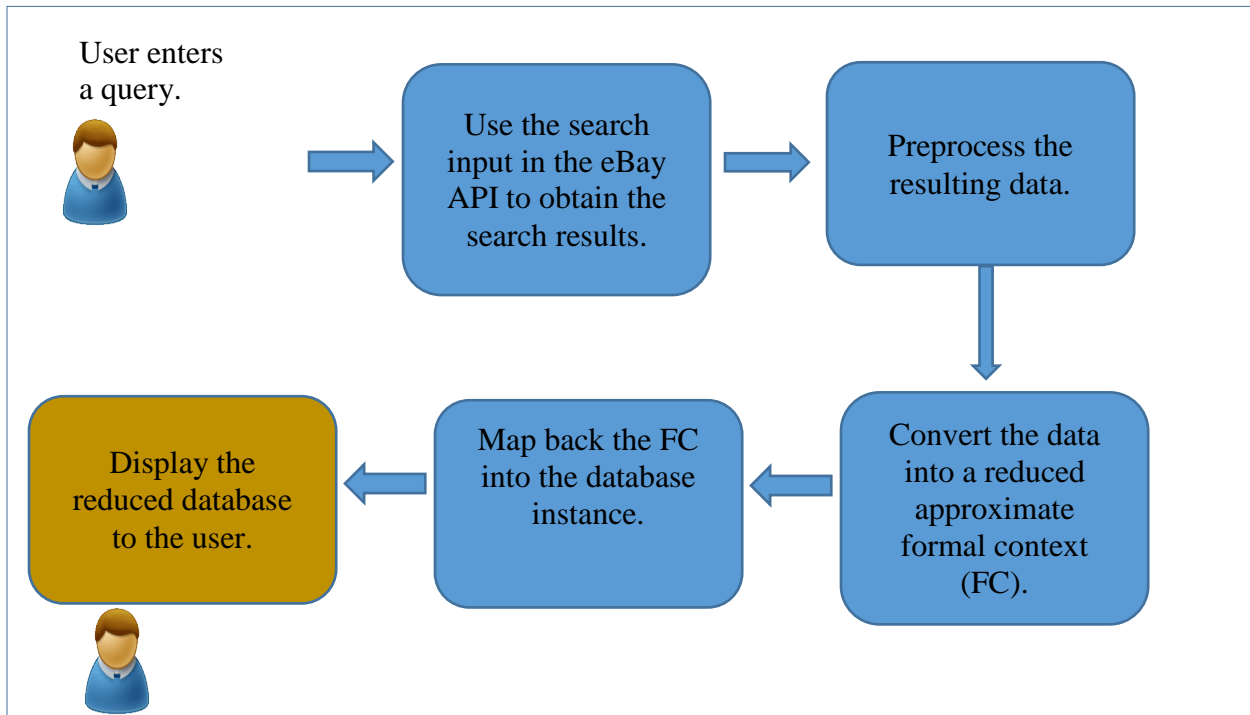


Figure 18: Real-Time Data Analysis

Real-time data analysis tasks are done automatically using the developed software in the Java programming language. These tasks are done sequentially, starting when the user issues a search query and continuing until he or she obtains the desired product, as will be explained in detail in the next section.

4.3 Implementation

The developed app was named as an FD eBay search app. It was developed using Android Studio Integrated Development Environment (IDE) and Java. By using the API authentication account in eBay, we were able to send a request for a specific item and get back the response in the format of an XML file.

The app consists of the following classes:

1. Main Activity Class:
2. eBay XML Helper Class
3. Post Value Class
4. Record Item Class
5. RecordListAdapter Class
6. Transformer Class

Figure 19 illustrates the workflow of the developed application.

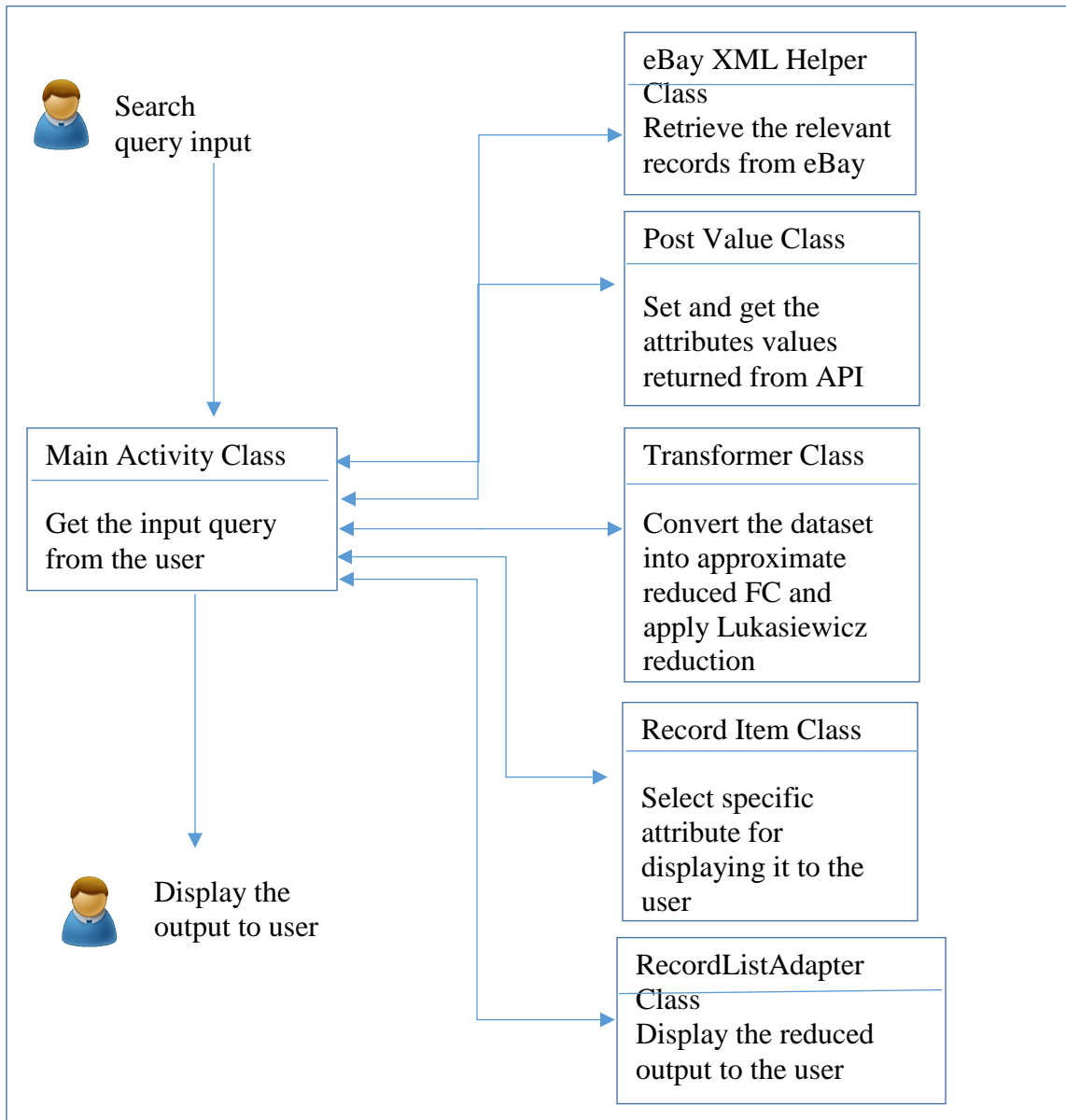


Figure 19: Application Workflow

After implementing the app that is based on the proposed conceptual method, the next step is to evaluate its performance and compare it with other online system as will be explained in the next chapter.

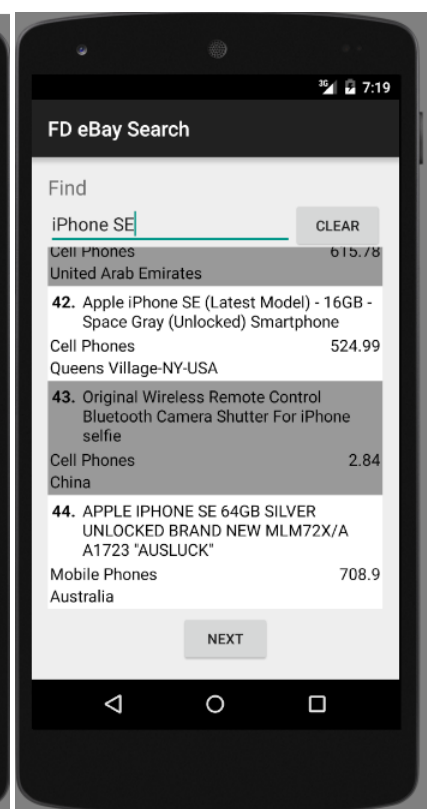
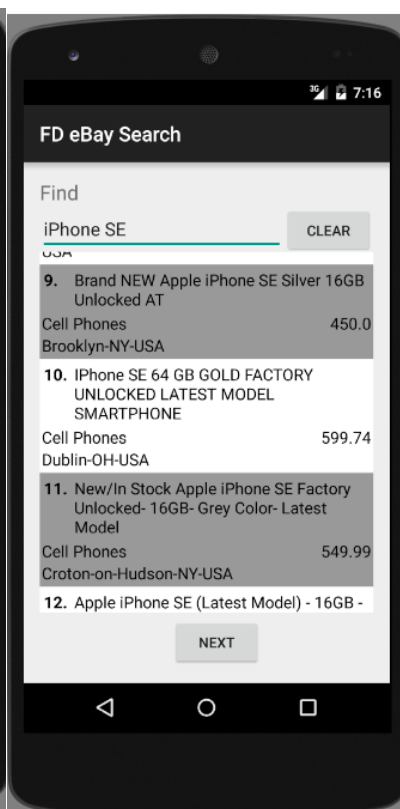
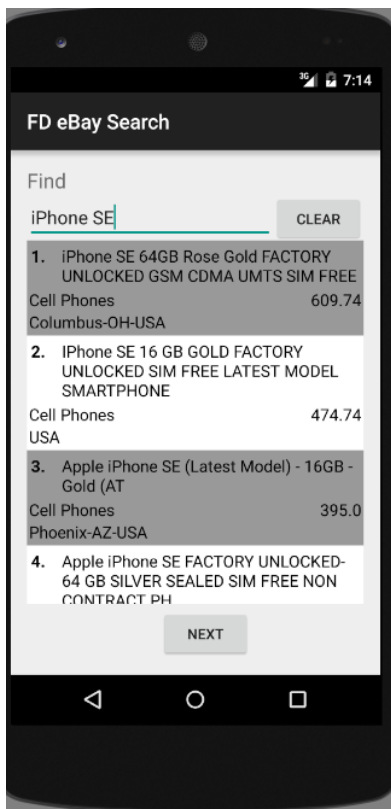
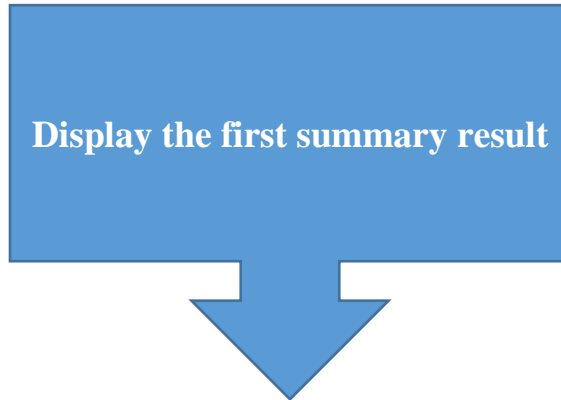
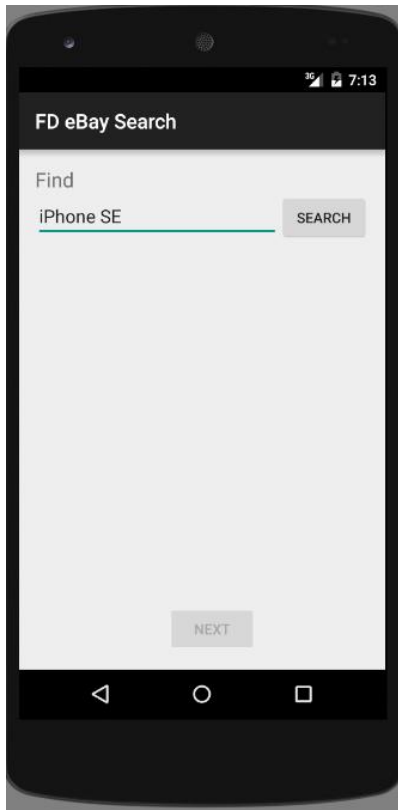
Chapter 5: Experimental Results, Validation and Evaluation

In this chapter, we will talk about the validation of the developed online system interface. Therefore, the experiments conducted were for two purposes: one to evaluate the proposed app and the other to compare it with other available online systems to show its efficiency.

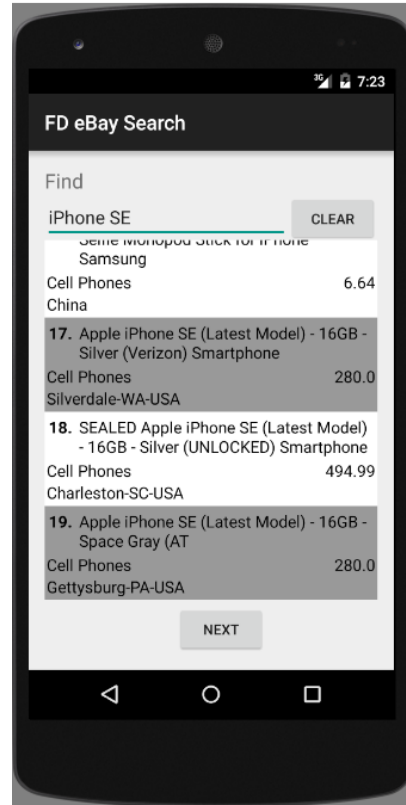
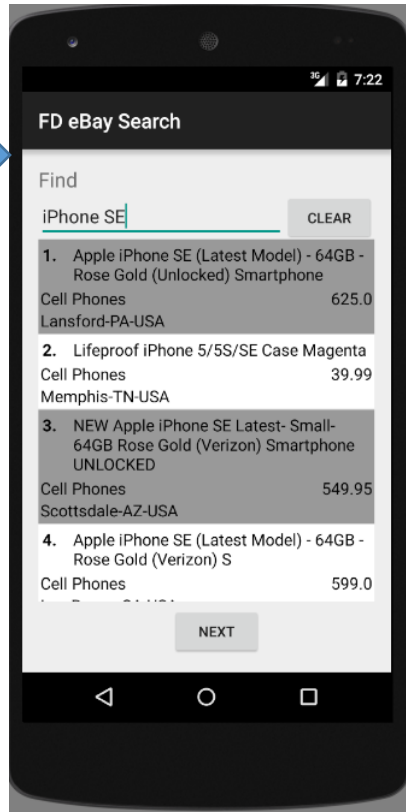
5.1 Testing the Developed App

The developed app was tested and evaluated using real smart phones instead of using the emulator provided by the PC for making the experiments real. Ten different cases were tested (i.e. 10 different search products) using the developed app on a smart phone. This step was mainly conducted to prove the efficiency of the developed app by recording its performance and revealing its points of strengths and weaknesses. Five evaluation metrics were used: number of clicks, number of navigation screens, processing time, reduction percentage and efficiency.

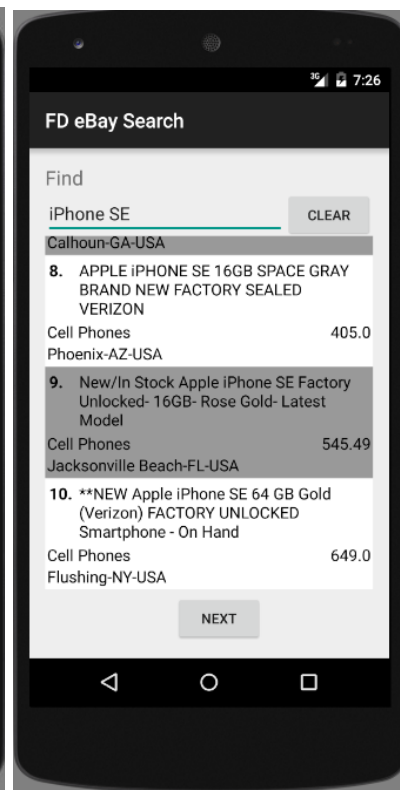
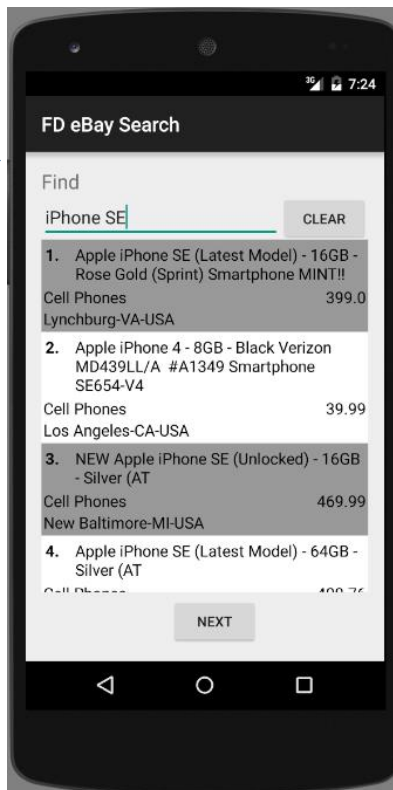
Testing and evaluating the proposed app starts after the user presses the search button for a specific product, as shown in Figure 20 for the following test case: A user enters a search query for iPhone SE. Then, an initial summary of results will be displayed on the screen. The displayed result contains the matched records for the input query from the eBay website, the description of the product, its price, the category name and the location. If the user finds his or her product from the first displayed summary, then he or she clicks on the desired description, and a link for the product will be opened. Otherwise, he or she presses on the next button to display another summary set and continues his or her search until he or she finds the desired product:



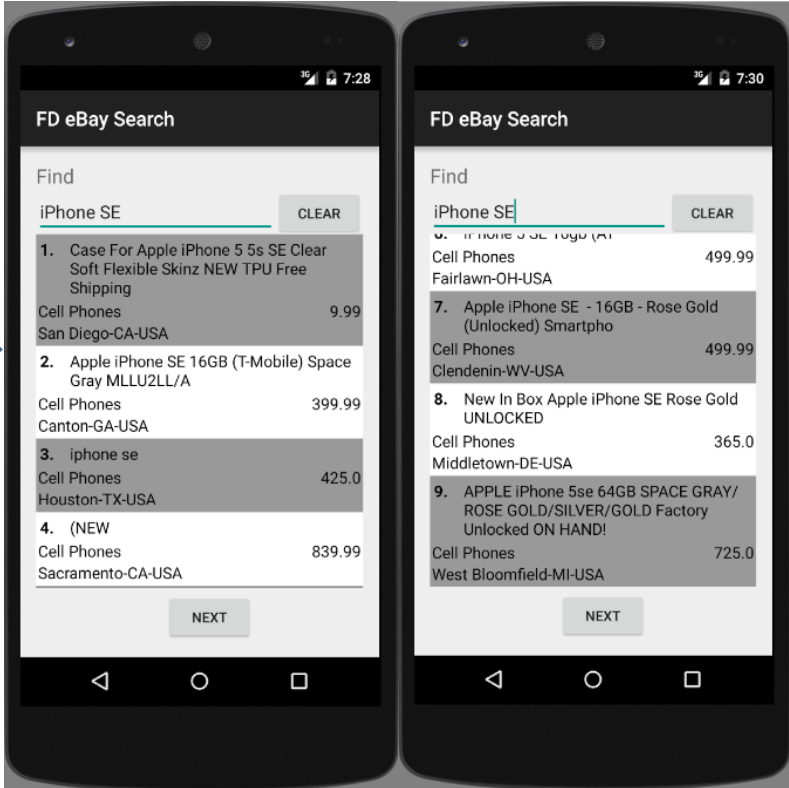
Display the second summary level



Display the third summary level



Display the fourth summary level



Display the last summary level

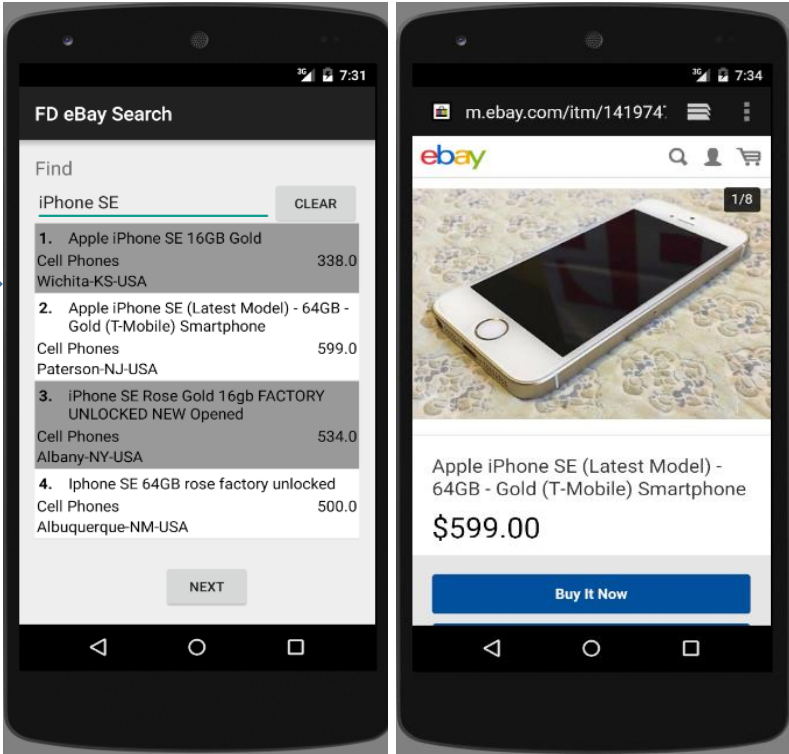


Figure 20: Testing the Developed App

5.2 Compare Developed App with eBay App

In order to evaluate the developed app's performance, ten users compared its performance with another online market system like eBay using five evaluation metrics:

1. **Number of clicks:** the number of clicks needed after pressing the search button to reach the desired product
2. **Number of navigation screens:** the number of search levels we have to go through before finding the product
3. **Response time:** the time taken for the search results to appear on the screen after pressing the search button
4. **Reduction percentage:** Shown as a group of percentages, this shows the percentage of reduced objects shown at each reduction level compared to the total number of search results.
5. **Relevance to the user (satisfaction):** how satisfied the user was with the search results.

This is put into a rating from 1 to 5, where each number stands for the following:

- a. 1 – Very dissatisfied
- b. 2 - Somewhat dissatisfied
- c. 3 – Neither satisfied nor dissatisfied
- d. 4 – Somewhat satisfied
- e. 5 – Very satisfied

Tables 15 and 16, display the average results of evaluating the developed app versus the eBay online system using the five metrics above.

Table 15: Average Results of Evaluating the Developed App.

Items	Number of Clicks	Number of Navigation Screens	Response Time	Reduction Percentage	Relevance to the user Satisfaction
1. Samsung Galaxy (J1)	3	4	21 secs	43/30/19/7/1	5
2. Dior Perfume (Miss Dior)	0	1	24 secs	36/19/21/11/5/5/1	5
3. Aldo Handbag (Faux Leather)	0	1	23 secs	19/17/11/10/10/9/9/10/5	5
4. Loung Chair (patio)	1	2	14 secs	64/29/6/2	5
5. Grandfather Clock (Tempus)	2	3	25 secs	38/27/18/13/3	5
6. Water Floss (Waterpick)	0	1	30 secs	33/31/18/13/4/1	5
7. Bvlgari Scarf (mens)	1	2	17 secs	23/33/35/9	5
8. Compass (wrist)	2	3	28 secs	27/22/17/20/9/4/1/	5
9. Geometry Box (Faber Castell)	0	1	22 secs	55/27/13/5	5
10. Rolex (Oyster Watch)	0	1	27 secs	25/24/14/9/7/7/8/5/1	5

Table 16: Average Results of Evaluating the EBay Application

Items	Number of Clicks	Number of Navigation Screens	Response Time	Reduction Percentage	Relevance to the user Satisfaction
1. Samsung Galaxy (J1)	0	1	12 secs	Not Applicable	4
2. Dior Perfume (Miss Dior)	0	1	4 secs	Not Applicable	5
3. Aldo Handbag (Faux Leather)	0	1	5 secs	Not Applicable	5
4. Loung Chair (patio)	0	1	30 secs	Not Applicable	3
5. Grandfather Clock (Tempus)	0	1	45 secs	Not Applicable	3
6. Water Floss (Waterpick)	0	1	52 secs	Not Applicable	3
7. Bvlgari Scarf (mens)	0	1	38 secs	Not Applicable	3
8. Compass (wrist)	0	1	31 secs	Not Applicable	3
9. Geometry Box (Faber Castell)	0	1	10 secs	Not Applicable	4
10. Rolex (Oyster Watch)	0	1	18 secs	Not Applicable	3

It is clear from the above evaluation metrics and the comparison with the eBay online system that the developed application is much faster than eBay, as shown by the time-consuming metrics. However, the user may need to go through many search summary sets (navigation screens) to reach his or her desired product, which is not the case when using the eBay online system.

Chapter 6: Conclusion and Future Work

This chapter will conclude the thesis work and give some recommendations for future enhancement.

6.1 Conclusion

Improving online system interfaces that are built on hidden databases has become interesting to many researchers due to the increase in global usage of those online systems. In this thesis, a new data-analysis method is proposed that is based on functional dependency preservation and data reduction methods. The proposed conceptual method consists of two stages, which are offline static data analytic and real time data analytic. In the process of offline static data analysis, we were able to discover the domain of the hidden database and then extract the functional dependencies. Extracting the functional dependencies enables us to select the most dependent attributes that we consider in the interface. During real-time data analysis, Formal Concept Analysis and the data reduction method, which is based on the Lukasiewicz implication, were utilized to display multiple levels of the most representative data objects on the screen. Experiments were conducted on the developed mobile application, which proves its efficiency and shows promising results. Evaluation of the performance of the developed mobile application and comparison with other online market systems reveal the reduction power implemented in this research for improving the online system interface. In addition, Comparing the developed application that is based on the proposed conceptual method with the application developed in [39] shows the advantages of using functional dependencies preservation and data reduction method.

6.2 Future Works

This thesis could be further developed and enhanced in a number of ways:

- enabling the user to get multiple facets of summarized data and to be able to return back to any summarized sets;
- improving the user interface of the developed app by displaying product's image and gives more detail about the product; and/or
- enhancing the speed of the application by improving the implemented algorithm by reducing the complexity of the algorithm.

References

- [1] X. Jin, A. Mone, N. Zhang, and G. Das, “MOBIES: mobile-interface enhancement service for hidden Web database,” *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 1263–1266, 2011.
- [2] M. K. Swamy, P. K. Reddy, R. U. Kiran, and M. V. Reddy, “Interface tailoring by exploiting temporality of attributes for small screens,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5999 LNCS, no. April, pp. 284–295, 2010.
- [3] Y. Niu, X. Li, X. Meng, J. Sun, and H. Dong, “A Constraint-based User Interface Design Method for Mobile Computing Devices,” *2006 First Int. Symp. Pervasive Comput. Appl.*, pp. 343–347, 2006.
- [4] B. Ganter, G. Stumme, and R. Wille, *Formal Concept Analysis foundation and application*. Springer International Publishing, 1887.
- [5] K. Tunde, J. Rancz, and V. Varga, “A method for mining functional dependencies in relational database design using fca,” *Stud. Univ. Babeş-Bolyai, Inform.*, vol. LIII, no. 1, pp. 17–28, 2008.
- [6] J. Baixeries and M. Kaytoue, “Characterizing Functional Dependencies in Formal Concept Analysis with Pattern Structures,” *Ann. Math. Artif. Intel.*, vol. 72, no. 1, pp. 129–149, 2014.
- [7] H. Mannila and K. J. Raiha, “Algorithms for inferring functional-dependencies from relations,” *Data Knowl. Eng.*, vol. 12, no. 1, pp. 83–99, 1994.
- [8] H. Yao and H. J. Hamilton, “Mining functional dependencies from data,” *Data Min. Knowl.*

- Discov.*, vol. 16, no. 2, pp. 197–219, 2008.
- [9] D. M. Kroenke and D. J. Auer, “Functional dependencies and normalization,” *Database concepts*, pp. 64–75, 2011.
- [10] Stéphane Lopes, J.-M. Petit, and L. Lakhal, “Efficient Discovery of Functional Dependencies and Armstrong Relations,” *Proc. 7th Int. Conf. Extending Database Technol. (EDBT 2000)*, vol. 1777, pp. 350–364, 2000.
- [11] J. Baixeries and M. Kaytoue, “Characterizing Functional Dependencies in Formal Concept Analysis with Pattern Structures,” *Ann. Math. Artif. Intel.*, vol. 72, no. 1, pp. 129–149, 2014.
- [12] T. Papenbrock, T. Neubert, J. Rudolph, M. Sch, J. Zwiener, and F. Naumann, “Functional Dependency Discovery : An Experimental Evaluation of Seven Algorithms,” *Proceeding VLDB Endow.*, vol. 8, no. 10, pp. 1082–1093, 2015.
- [13] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen, “TANE: An efficient algorithm for discovering functional and approximate dependencies,” *Comput. J.*, vol. 42, no. 2, pp. 100–111, 1999.
- [14] N. Novelli and R. Cicchetti, “FUN: An Efficient Algorithm for Mining Functional and Embedded Dependencies,” *Proc. 8th Int. Conf. Database Theory (ICDT 2001)*, vol. 1973, pp. 189–203, 2001.
- [15] Y. Ye and C. C. Chiang, “A parallel apriori algorithm for frequent itemsets mining,” *Proc. - Fourth Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2006*, pp. 87–94, 2006.
- [16] Z. Abedjan, P. Schulze, and F. Naumann, “DFD: Efficient Functional Dependency Discovery,” *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manag. (CIKIM 2014)*, pp. 949–

- 958, 2014.
- [17] C. Wyss, C. Giannella, and Edward Robertson, “Fastfds: A heuristic-driven, depth-first algorithm for mining functional dependencies from relation instances extended abstract,” *Data Warehous. Knowl.*, pp. 101–110, 2001.
- [18] P. a. Flach and I. Savnik, “Database dependency discovery: a machine learning approach,” *AI Commun.*, vol. 12, no. 3, pp. 139–160, 1999.
- [19] M. Farach-Colton and Y. Huang, “A linear delay algorithm for building concept lattices,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5029 LNCS, pp. 204–216, 2008.
- [20] A. Kumar, “New Techniques for Data Reduction in a Database System for Knowledge Discovery Applications,” *J. Intell. Inf. Syst.*, vol. 48, pp. 31–48, 1998.
- [21] J. Tuya, C. de la Riva, M. J. Suarez-Cabal, and R. Blanco, “Coverage-Aware Test Database Reduction,” *IEEE Trans. Softw. Eng.*, vol. 5589, no. c, pp. 1–1, 2016.
- [22] S. Elloumi, J. Jaam, A. Hasnah, A. Jaoua, and I. Nafkha, “A multi-level conceptual data reduction approach based on the Lukasiewicz implication,” *Inf. Sci. (Ny)*, vol. 163, no. 4, pp. 253–262, 2004.
- [23] J. Madhavan, D. Ko, \Lucja Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, “Google’s Deep Web crawl,” *Proc. VLDB Endow. Arch.*, vol. 1, no. 2, pp. 1241–1252, 2008.
- [24] Y. He, D. Xin, V. Ganti, S. Rajaraman, and N. Shah, “Crawling deep web entity pages,” *Web Search Data Min.*, pp. 355–364, 2013.

- [25] R. Yerneni, C. Li, H. Garcia-Molina, and J. Ullman, "Computing capabilities of mediators," *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 443–454, 1999.
- [26] a. Ntoulas, P. Pzerfos, and J. C. J. Cho, "Downloading textual hidden web content through keyword queries," *Proc. 5th ACM/IEEE-CS Jt. Conf. Digit. Libr. (JCDL '05)*, pp. 100–109, 2005.
- [27] M. J. Tsai and D. J. Chen, "Generating user interface for mobile phone devices using template-based approach and generic software framework," *J. Inf. Sci. Eng.*, vol. 23, no. 4, pp. 1189–1211, 2007.
- [28] G. Menkhous and W. Pree, "User interface tailoring for multi-platform service access," *Proc. 7th Int. Conf. Intell. user interfaces - IUI '02*, p. 208, 2002.
- [29] S. T. Buckland, E. a. Rexstad, T. a. Marques, and C. S. Oedekoven, *Distance Sampling: Methods and Applications*. Springer International Publishing, 2015.
- [30] A. Doucet, S. Godsill, and C. Andrieu, "On sequential {Monte Carlo} sampling methods for {Bayesian} filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.
- [31] M. J. Lesot, M. Rifqi, and H. Benhadda, "Similarity measures for binary and numerical data: a survey," *Int. J. Knowl. Eng. Soft Data Paradig.*, vol. 1, no. 1, p. 63, 2009.
- [32] E. D. Bolker and M. M. Mast, "Relative and Absolute Change Percentages," pp. 1–6, 2007.
- [33] E. Rezk, S. Babi, F. Islam, and A. Jaoua, "Uncertain Training Data Set Conceptual Reduction : A Machine Learning Perspective," *FUZZ-IEEE*, 2016.
- [34] C. Explorer, F. C. Analysis, R. Wille, and T. Taran, "Concept Explorer . The User Guide

- ConExp installation Working with Concept Explorer,” *Environment*, pp. 1–13, 2006.
- [35] I. Baqui, I. Zamora, J. Mazón, and G. Buigues, *Artificial Neural Networks*. Springer Science Business Media, 2011.
- [36] R. Elmasri, S. Navathe, " Database Systems:Models, Languages, Design, And Application Programing ".Pearson International Edition , Sixth Edition.
- [37] <http://archive.ics.uci.edu/ml/datasets.html>.
- [38] S. Khalili, “A Combined Conceptual Approach for Extracting Functional Dependencies and Completing Missing Data,” Qatar University, 2014.
- [39] A. Aqle, “Mobile App for Hidden Data Analytics of Online Marketplace Systems,” 2016.
- [40] S. O. Kuznetsov and S. a. Obiedkov, “Comparing performance of algorithms for generating concept lattices,” *J. Exp. Theor. Artif. Intell.*, vol. 14, no. 2–3, pp. 189–216, 2002.