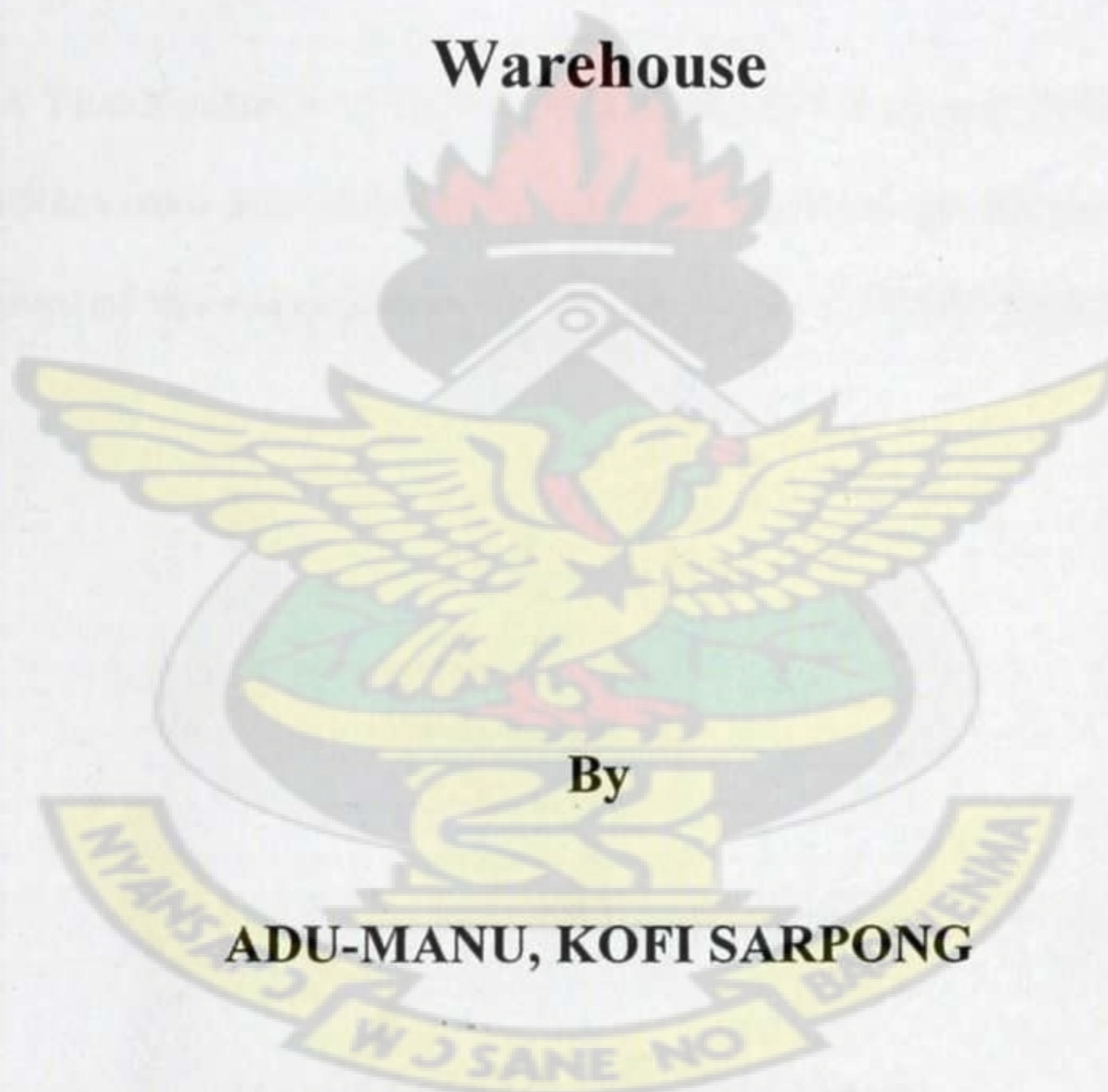


Kwame Nkrumah University of Science and Technology,

Kumasi Ghana

A Conceptual Framework for Data Cleansing in a Data

Warehouse



By

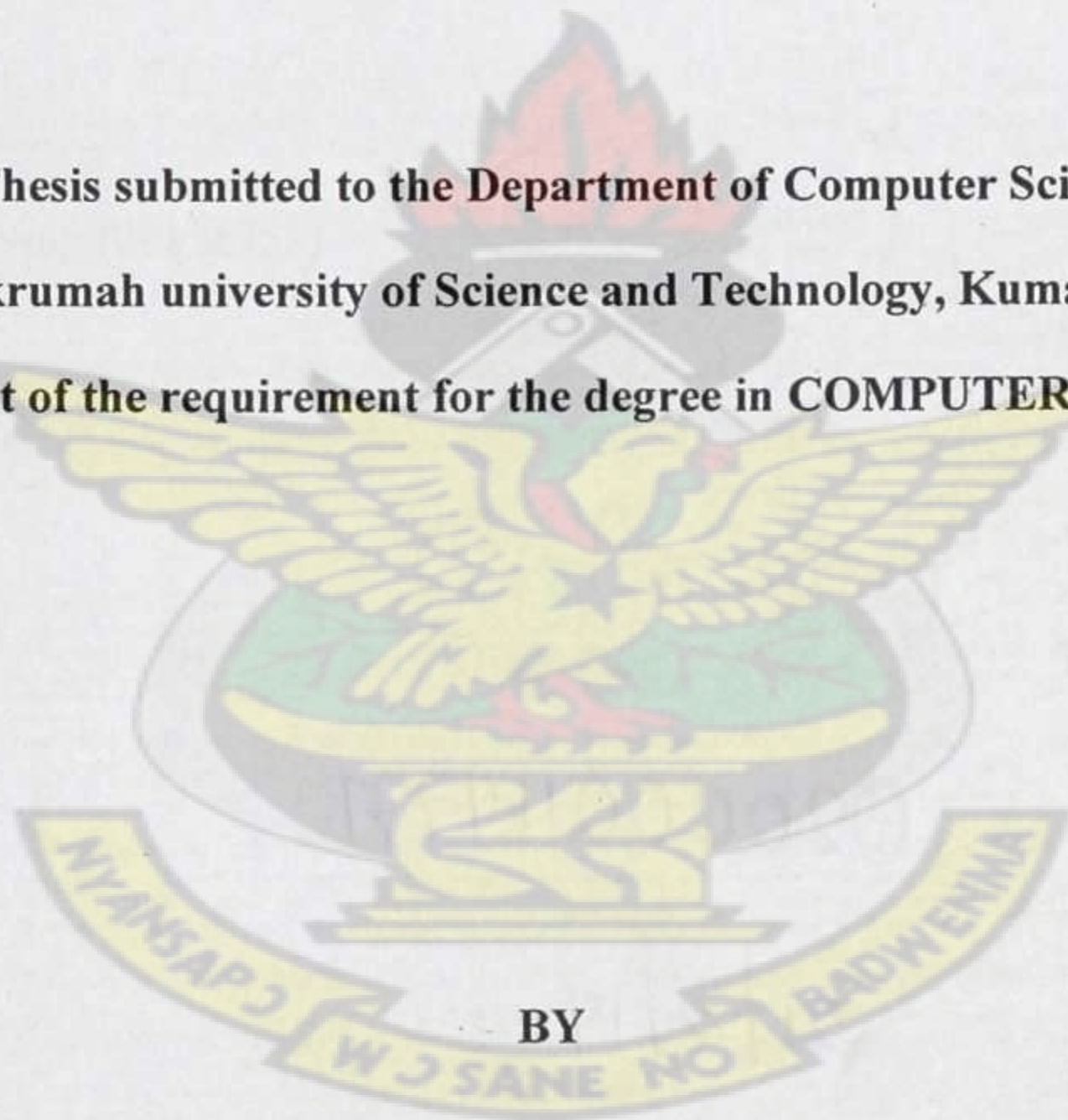
ADU-MANU, KOFI SARPONG

February, 2013

A Conceptual Framework for Data Cleaning in a Data Warehouse

KNUST

**A Thesis submitted to the Department of Computer Science,
Kwame Nkrumah university of Science and Technology, Kumasi in partial
fulfillment of the requirement for the degree in COMPUTER SCIENCE**



**ADU-MANU, KOFI SARPONG
B.SC. COMPUTER SCIENCE**

February, 2013

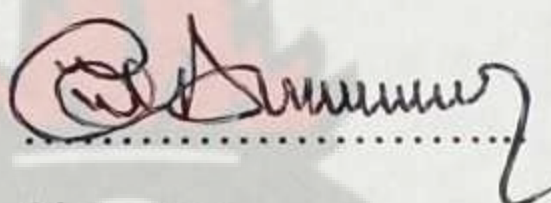
CERTIFICATION

I hereby declare that this submission is my own work towards the MPhil and that, to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgement has been made in the text.

KNUST

Kofi Adu-Manu Sarpong (20136753)

Student



Signature

18-4-2013

Date

Certified by:

Mr. J.G. Davis

Supervisor



Signature

24/4/13

Date

Mr. J.K. Panford

Supervisor



Signature

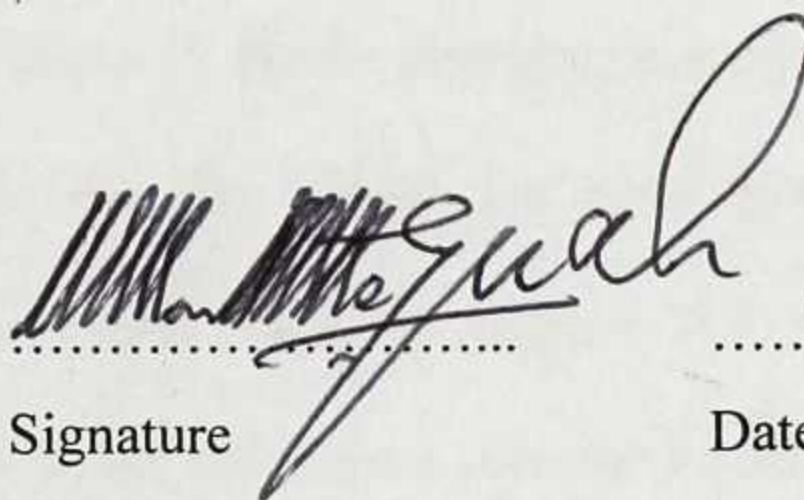
24/4/2013

Date

Certified by:

Dr. J.B Hayfron-Acquah

Head of Department



Signature

26/4/13

Date

ABSTRACT

Data Cleansing is an activity involving a process of detecting and correcting the errors and inconsistencies in data warehouse. It deals with identification of corrupt and duplicate data inherent in the data sets of a data warehouse to enhance the quality of data.

This research seeks to facilitate the data cleaning process by addressing the problem of duplicate records detection pertaining to the various attributes (name, place of birth, etc) of the data sets. It provides an algorithm through a new framework for identifying errors in the attributes of the data sets of an already existing data warehouse. The key feature of the research includes its proposal of a new framework through a well defined algorithm. Primarily is the acknowledgment that the main objective involves the development an algorithm to detect and remove duplicates and missing data. Duplicates and missing data in a set of records results in inconsistencies in reports and analyses and affects the performance of an organization and their decision making. Atpmts were directed at investigating some existing approaches and came up with their strengths and weaknesses.

System analysis and design methodology tools were used in the development and testing the performance of the algorithm. Secondary data was obtained from a students' database and was used to input source to test the systems performance. Several data sets of varied sizes (ranging from Kilobyte to Gigabyte) were used. The research results strongly revealed that maintenance of the clean data within periodic intervals, the file format, the error type, the number of processors and memories are major factors in data cleansing process. It is suggested that the strategy of storing all the strings that result from the addition of columns be stored in a dictionary for ease of searching and to enhance the cleaning process.

ACKNOWLEDGEMENT

To God be the Glory, great things He has done. Gracious God to Thee we raise this sacrifice of praise.

I would like to extend my earnest appreciation to my supervisors Mr. J. G. Davis and Mr. J.K Panford for their support and encouragement throughout the period of this research.

I would like to specially acknowledge my family for all their love and support. In particular, I would like to thank my beautiful wife Adwoa Agyeiwaa for selflessly sharing me with school and work when she needed me most. Emmanuel, Kofi, Ama my siblings thanks so much for even going hungry for me to reach this height. Also, thanks to my mother, Margaret Adu for been integral part of my academic journey.

I want to gratefully acknowledge the following special faculty at Valley View University for their positive influence and guidance which further motivated me to pursue a master's degree: Mr. Henry Quarshie (Dean, ICS – VVU), Mr. Abraham Bamfo Boakye, Mr. Dominic Damoah, Dr (Mrs.) Josephine Ganu (Dean, School of Business – VVU) and Mr. John Kingsley Arthur. I would like to express my warmest thanks to Mr. Tonto Baffour for his support and encouragement not forgetting Mr. Winfred Larkotey my reader.

As for the Hage family (Rev. & LP Hage, Nabil, Nazar Boy, Narda, and Nadia) God bless you for your hospitality and love.

DEDICATION

This thesis is dedicated to my lovely wife and son, Mrs. Adwoa Agyeiwaa Adu-Manu and Nana Yaw Sarpong and the entire family especially my mother.

KNUST



LIBRARY
KWAME NKRUMAH
UNIVERSITY OF SCIENCE & TECHNOLOGY
KUMASI

TABLE OF CONTENT

CERTIFICATION	iii
ACKNOWLEDGEMENT	v
DEDICATION	vi
ABSTRACT	iv
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS AND ACRONYMS	xiv
TABLE OF CONTENT	vii
Chapter One	1
Introduction.....	1
1.0 Introduction	1
1.1 Background of the Study.....	1
1.2 Motivation of the Study.....	4
1.3 Research problem	5
1.4 Aims and Objectives of the Research	6
1.5 Research Questions	6
1.6 Study Justification/Rationale	7
1.7 Scope of the Study.....	7
1.8 Definition of Data Warehouse.....	8
1.8.1 Advantages of Data Warehousing	9
1.8.2 Problems with Data Warehousing	10
1.9 Definition of Data Cleansing.....	10
1.10 Data Cleaning Background	11
1.11 High Quality Data	13
1.12 Data cleaning process perspective.....	14
1.13 The Existing System.....	15

1.13.1	Millicom Ghana Limited.....	15
1.14	Organization of the Thesis	16
Chapter two	18
Literature Review.....		18
2.1	Review of Existing Approaches for Data Cleaning	18
2.1.1	Potter's Wheel: An Interactive Data Cleaning System.....	18
2.1.2	IntelliClean: A Knowledge-Based Intelligent Data Cleaner.....	26
2.1.3	ARKTOS: A Tool For Data Cleaning and Transformation in Data Warehouse Environments.....	31
2.1.4	AJAX	33
2.2	Reviewed Papers on Data Cleaning	35
2.2.1	Problems, Methods, and Challenges in Comprehensive Data Cleansing	36
2.2.2	Data Cleaning: Problems and Current Approaches	37
2.2.3	Matching Algorithms with a Duplicate Detection System	38
2.2.4	Open User Involvement in Data Cleaning for Data Warehouse Quality.....	39
2.3	Conceptual Framework to support data cleaning process.....	47
Chapter Three	52
Methodology		52
3.1	Selecting a Research Methodology.....	52
Chapter four	54
Analysis and Design		54
4.0	Analysis and Design in perspective	54
4.1	Analysis.....	55
4.1.1	Presentation of data	55
4.1.2	Cleansing process	56
4.2	Users of the proposed system	57
4.3	Benefits of the proposed system	57
4.4	Data collection	58
4.5	The proposed data cleansing method.....	59
4.6	Requirements Analysis	59
4.6.1	Functional Requirements	59

4.6.2	Non-Functional Requirements	59
4.6.1.1	Hardware Requirement	62
4.6.1.2	Software Requirement	62
4.6.3	Input Requirements	63
4.6.4	Processing Requirements	63
4.7	Design of Proposed System	63
4.7.1	Processing Modeling	65
4.8	The Pseudo-code Algorithm	65
4.8.1	Data from the Student table	67
4.8.2	The Algorithm: Method to Detect Errors	68
4.8.3	Applying Data Cleansing	71
4.8.4	Procedure for Detection Errors	75
4.8.5	Code Snippet (Application of the algorithm): Duplicate data	77
4.8.6	Code Snippet (Application of the algorithm): Missing data	80
4.9	Determining the Algorithm Complexity	82
4.9.1	Performance and complexity of algorithm	82
4.9.2	The running time of the algorithm (Time Complexity)	83
4.9.3	Hardware Complexity due to the proposed framework	86
4.9.4	Space Complexity	86
4.9.5	System Evolution	86
4.9.6	Modeling Flowchart representation for the cleansing system	87
4.9.7	Modelling Use Case diagram for the integrated cleansing system	88
4.9.8	System model diagram for the integrated cleansing system	90
4.9.9	Data Flow diagram (Level 1) for the integrated cleansing system	91
4.9.10	Context Level diagram for the integrated cleansing system	91
Chapter five	92
5.0	Implementation	92
5.1	System Environment	93
5.2	Graphical User Interfaces (GUIs) for the System implementation	94
5.2.1	The Main Interface	95
5.2.2	Populating data from MSSQL using Select Database Dialog box	96

5.2.3	Populating data from MySQL using <i>MySQL Data</i> Dialog box	96
5.2.4	Data import interface	97
5.3	Testing.....	97
5.3.1	Expected performance for detecting duplicates	98
5.3.2	Expected Performance for Missing data	101
5.3.3	Test Phases.....	104
5.3.4	Testing Descriptions	104
5.4	Deployment.....	106
Chapter six	107
6.0	Summary and Conclusion	107
6.1	Contributions to the research.....	108
6.2	Recommendation and Future work	109
REFERENCES	111



LIST OF TABLES

Table 2.1: Comparative analysis of Data Cleaning Frameworks	35
Table 2.2: Critical factors aiding in data cleansing	49
Table 4.1: Concatenated rows	69
Table 4.2: Equality among strings	74
Table 4.3: Running Time for detecting duplicates	84
Table 4.4: Running time for detecting missing data	85
Table 4.5: Definition of the Use Case	89
Table 4.6: Definition of the Use Case Continued	89
Table 5.1: Test with two CPUs	99
Table 5.2: duplicate test five CPUs	100
Table 5.3: test with two CPUs for missing data	102
Table 5.4: missing data test with five CPUs	103
Table 5.5: System Integration Testing	104
Table 5.6: System Integration Testing (Continuation)	105
Table 5.7: Comparison between reviewed systems and the proposed system	105

LIST OF FIGURES

Figure 1.1: Data Warehouse Architecture [source: tech-faq (2012)]	9
Figure 2.1: Potter's Wheel Architecture (Source: Raman and Hellerstein, 2001)	19
Figure 2.2: Snapshot of Potter's Wheel user interface	21
Figure 2.3: Snapshot of discrepancy detection	24
Figure 2.4: An Overall Conceptual model of Intelliclean	27
Figure 2.5: A Knowledge-based Cleaning Framework cited by Lee et al., (2000)	29
Figure 2.6: The conceptual framework	48
Figure 4.1a: Some tables in the database	55
Figure 4.1b: Tables at the schema level	56
Figure 4.1c: Oracle interface for connecting to data source	61
Figure 4.2: diagram for cleansing data records	Error! Bookmark not defined.
Figure 4.3: A table of records with labels	67
Figure 4.4: table of records	69
Figure 4.5: column identification	71
Figure 4.6: Comparing two fields	73
Figure 4.7: The Flowchart	87
Figure 4.8: The Use Case Diagram of the pseudo code	88
Figure 4.9: The System Model	90
Figure 5.1: Use Case for user accessibility	93
Figure 4.6: Comparing two fields	73
Figure 5.3: User operates on data	94

Figure 5.4: Main interface for iCleaner 95

Figure 5.5: MSSQL connection interface 96

Figure 5.6: MySQL connection interface 96

Figure 5.7: Main interface for cleansing errors 97

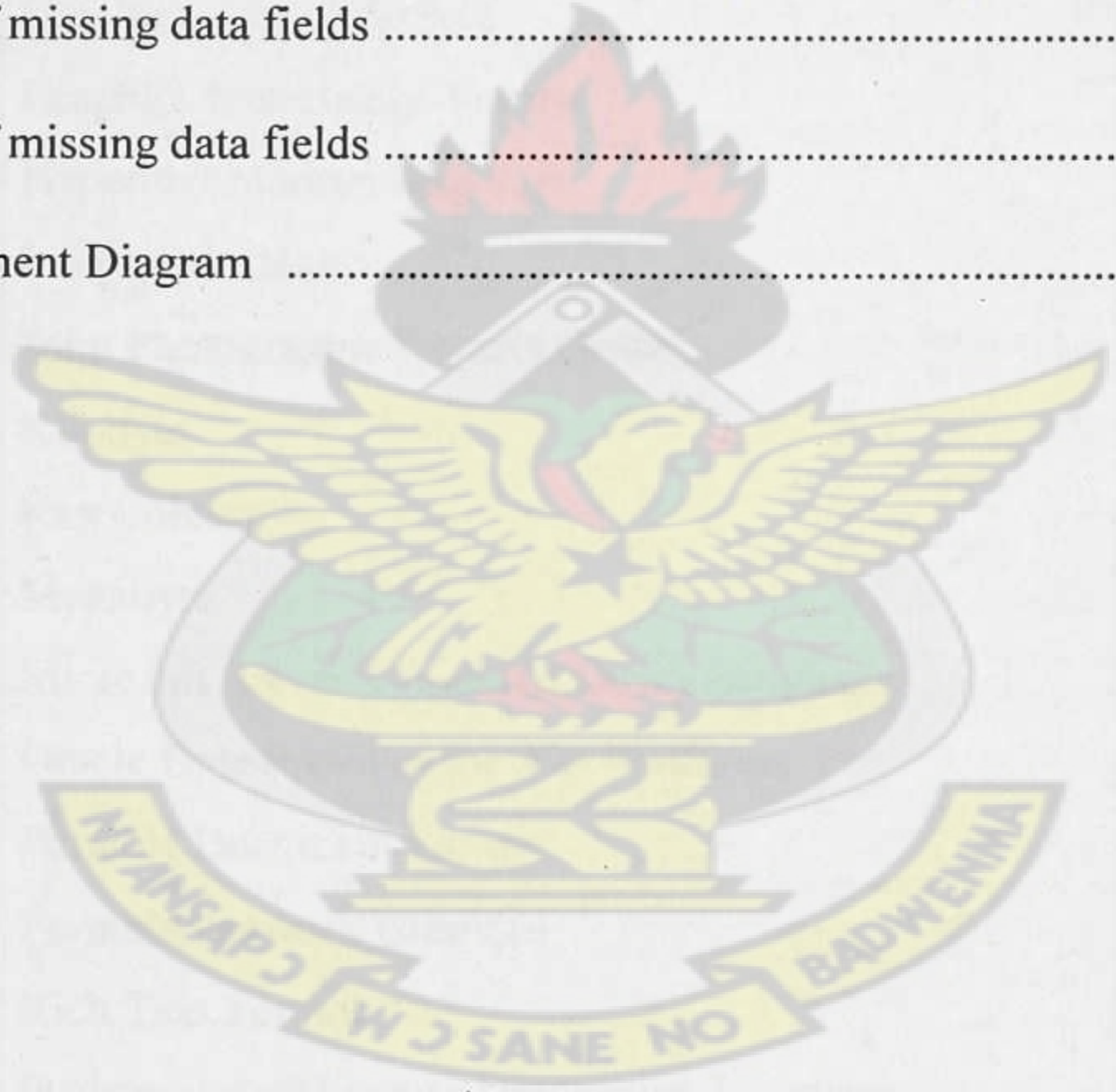
Figure 5.8: Graph of duplicate test with two CPUs 99

Figure 5.9: Graph of duplicate test with five CPUs 101

Figure 5.10: graph of missing data fields 102

Figure 5.11: graph of missing data fields 103

Figure 5.12: Deployment Diagram 106



LIST OF ABBREVIATIONS AND ACRONYMS

ASCII	American Standard Code for International Interchange
CDR	Call Detail Records
DC	Data Cleansing
DW	Data Warehouse
DM	Data Mining
DQ	Data Quality
ETL	Extract-Transform-Load
GB	Gigabyte
GUI	Graphical User Interface
GIF	Graphics Interchange Format
HTML	Hypertext Markup language
iCleaner	Integrated Cleaner
JPEG	Joint Photographic Experts Group
KB	Kilobyte
KC	Key Column
MB	Megabyte
MSSQL	Microsoft Server Structured Query Language
ODP.Net	Oracle Data Provider for .Net Platforms
PDF	Portable Document Format
PNG	Portable Network Graphics
RTF	Rich Text Format
SADL	Structural Architecture Description Language
SSADM	Structured Systems Analysis and Design Methodology
SOR	Sum of Rows
SQL	Structured Query Language
TIFF	Tag Image File Format
VVU	Valley View University
WAV	Waveform Audio Format
WMA	Windows Media Audio
XML	Extensible Markup Language

Chapter One

Introduction

1.0 Introduction

1.1 Background of the Study

The advances in technological development have seen rise in business transactions in Ghana, and new methods of preserving data have become the issue for businesses to consider today. Muller and Freytag (2000) in their work stated that, the application and exploitation of huge amounts of data take an ever increasing role in the modern economy, government, and research.

According to the Bank of Ghana's annual report and accounts (2007), the bank has seen the need of data migration and aggregation of the numerous data the bank receives from the various banks, microfinance corporations and other organizations they transact business with by implementing a data warehouse infrastructure. Though it is not yet implemented.

Bohan (2012) identifies two purposes this could achieve: first, the productivity of the current operational transaction processing is not disturbed by analysis queries and second, the data can be logically consolidated into coherent, searchable entities, e.g. by trend/value, customer, etc.

According to Deku et al., (2012) data warehouse (DW) is a modern proven technique of handling and managing diversity in data sources, format and structure. Recent advances in database technologies are leading to the proliferation of different kinds of information design

with independent supporting hardware and software. This technology was developed to integrate heterogeneous information sources for analysis purposes.

According to Bradji and Bonfaida (2011), data warehouses integrate data from various operational sources (databases) all over the organizations. The Ghanaian organizations are gradually moving away from the traditional forms of data capturing and performance analysis into a more refined methodology.

Companies over the last couple of decades have done more data logging and capturing with the advent of computers with database capabilities. Many have found that these data are quite useful to augment or focus market groups if only the information was available for statistical analyses.

Data warehousing is a technology that seeks to create a repository for all sources of data needed by the organization so that, information could be accessed and used from one point. In computing, a data warehouse (DW) is a database used for reporting and analysing. The data stored in the warehouse is uploaded from the operational systems such as database applications and other external data sources.

Hellerstein (2008), in his paper stated that data collection has become a ubiquitous function of large organizations not only for record keeping, but to support a variety of data analysis tasks that are critical to the organization's mission. As a result researching into various aspects of data cleansing to find procedures to automatically or semi-automatically identify and correct errors in large data sets become relevant.

Poor data quality (DQ) continues to be an important issue for companies. Erroneous, incomplete or duplicate data leads to bad and poor business decisions which cost a lot (Diallo et al., n.d). The quality data can only be produced by cleaning the data and pre-processing it prior

to loading it in the data warehouse. As not all the algorithms address the problems related to every type of dirty data, one has to prioritize the need of its organization and use the algorithm according to their requirements and occurrence of dirty data (Porwal and Vora, 2013).

Chapman (2005) in his work Principles of Data Quality, explained that, data cleansing is an essential part of the information management chain, thereby defining data cleansing as the process used to determine inaccurate, incomplete, or unreasonable data and then improving the quality through correction of detected errors and omissions.

There is a high need of data cleansing since it is centered on the quality of data and to make the data “fit for use” by its user’s data through the removal of errors and having proper documentation.

Though many researchers have sought to come up with a framework for the general cleansing process, there still remain gaps which need to be addressed and tackle. It is to this background that the research is been conducted to come up with solutions (such as an algorithm and a framework) to cater for the gaps and contribute to the ongoing research within the data cleansing community.

1.2 Motivation of the Study

The concept of data warehouse and data cleansing is gaining popularity in the Ghanaian business environment. Companies are challenged with the increasing number of data collected from its customers and employees. This information is relevant for decision making by management for reporting purposes and data analysis.

The presence of data is important to organizations. But this alone does not ensure that all the managerial activities or functions and decisions could be smoothly undertaken. There is a compulsive requirement for the data to be meaningful or, in other words, data quality is of utmost importance if management is to take any advantage of the data at its disposal.

This research is motivated by several objectives. First of all, data collected from different sections or units in organizations representing any such entity as name, date employed, height, weight etc are collected and processed for management to keep track of customers and employees and track the volume of data from its numerous branches, analyze and interpret the data to support decision making processes and support the business process.

During data collection, several errors occur. Among these are omissions, data entry problems and double entry problems rendering databases redundant. According to Brown (1997) and Gardyn (1997) as cited by Mohan et al., (n.d), the data residing in operational databases, though perhaps of sufficient quality for transactional procession, contain duplications, inconsistencies, errors, missing data, and other issues that make them unsuitable for a decision support data warehouse, without considerable "cleansing" of the data.

The researcher is motivated to undertake this research so as to develop a data cleansing tool to enable companies, government agencies, institutions and other related agencies that collect and process huge data for reporting and analysis to get rid of some common errors. Muller and Freytag (2000) in their work states that erroneous data leads to unnecessary costs and probably loss of reputation when used to support business processes.

1.3 Research problem

Businesses in Ghana collect several data from their customers from different sources as well as other educational institutions. Most of these companies place much importance on the quality of information in order to effectively perform analysis and report to their stakeholders. Hamza et al., (2012), emphasizes on the fact that, the quality of data is the key to the success of the fast growing industry. For tactical decision making, the data extracted from the data warehouse should be efficient and useful for effective reporting.

Muller and Freytag (2004) in their studies of Problems, Methods, and Challenges in Comprehensive Data Cleansing, identified accuracy, integrity, completeness, validity, consistency, schema performance, uniformity, density and uniqueness as the set of data quality criteria that could be used within optimization of data cleansing by specifying priorities for each of the criteria. Today, companies employ individuals to go through their huge data to indentify anomalies and remove them.

Marcus and Maletic (2000) in their studies identify methods of error detection and discussed three methods for data cleansing – define error type, search error instances and correct errors in order to ensure the quality of data. The quality of data is often evaluated to determine usability and to establish the processes necessary for improving data quality (Dongre, 2004).

Data cleansing is a tedious and time taking exercise which requires a sound systematic approach, wise option in the process to be used and a basic understanding of the applicable existing system and the target system (i.e. the new application). It is therefore, a necessity to research into how to address the problems related to every type of “dirty data” in datasets and also research into the deficiencies of other frameworks and algorithms. The research seeks to solve the problems state above.

1.4 Aims and Objectives of the Research

The research seeks to establish weakness in the existing data cleansing tools and methods and propose a framework as well as an algorithm to address the weaknesses.

The specific objectives are;

1. To investigate the existing tool(s) for detecting missing data and removing duplications of data in data warehouse and determine their strengths and weaknesses.
2. To develop an algorithm to detect and remove missing and duplicated data.
3. To propose a new conceptual framework for detecting missing data and for removing duplicate data in data warehouse.
4. To measure the efficiency of the proposed algorithm

1.5 Research Questions

Many questions arise as a result of this research and they are as follows:

1. Are the existing tool(s) for detecting missing data and removing duplicated data from the warehouse robust?

2. What steps will be used in detecting missing data and removing duplicated data efficiently?
3. What factors will be required for the development of the framework?
4. How efficient is the proposed algorithm against the weaknesses of the existing tools?

1.6 Study Justification/Rationale

The development of humans and the society at large has caused or advanced the frontiers of researchers, educators and business operators to make investigations into several methods and techniques in order to be at par with technology. This research is intended to contribute to the existing collected works of research for use in academia, business organizations and government institutions since each of these bodies places much importance on the quality of information.

Results from the study will particularly provide an effective tool for managements of business organizations to use in order to maintain their good reputation and save time and money as well. This will enable them gain competitive advantage over other companies.

The study is also justified since a new framework will be proposed to support the data cleansing process.

1.7 Scope of the Study

The foundation of the research under study is in the area of data cleansing to determine duplicates and missing data in databases and files using an algorithm. Structured System Analysis and Design Methodology (SSADM) tools were carried to develop a system and student database of Valley View University will be used to validate the tool.

Accordingly, the findings will reflect only the data behavior of the attributes and parameters set by the database administrator at the University and moreover represent other data sets obtained from other sources. The researcher is limited by time, resources and opportunities to design a cleansing tool to deal with other anomalies (errors) such as incomplete data, etc.

1.8 Definition of Data Warehouse

There has been couple of definitions shared by intellectuals on the concept of Data warehousing. They come from varying perspective but making closely same point on what Data warehousing is. Few of these definitions are shared here.

According to Surajit and Dayal (1997), Data Warehousing is a collection of decision support technologies, aimed at enabling the knowledge worker (executive, manager, and analyst) to make better and faster decisions. Matyia (2007) in his paper also explains that, Data Warehouses (DW) integrate data from different various operational data sources all over the organizations. Typically, this helps organizations to create coherent and aggregate data for decision-making purpose.

Finally, Vassiliadis et al., (2001), considered a data warehouse to be a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources.

These definitions are quite different, but have something in common when they are considered as a group. We infer from Galhards et al., (2000) and simply put say that a data

warehouse contains a lot of data. Some of the data comes from operational sources in the organization and some of it may come from outside the organization.

The software process that facilitates the initial loading and the refreshment of DW contents is commonly known as Extract, Transform and Load (ETL) process. ETL is responsible for extracting the data from the different sources, transforming them and then for loading them into the DW with the probability that some of these sources contain dirty data is enough high. So the improvement of data is vital to make perfect and accurate conclusions (Fox et al., 1994; Chaudhuri and Dayal, 1997). The architecture is as shown in figure 1.1.

Below is the basic data warehouse architecture.

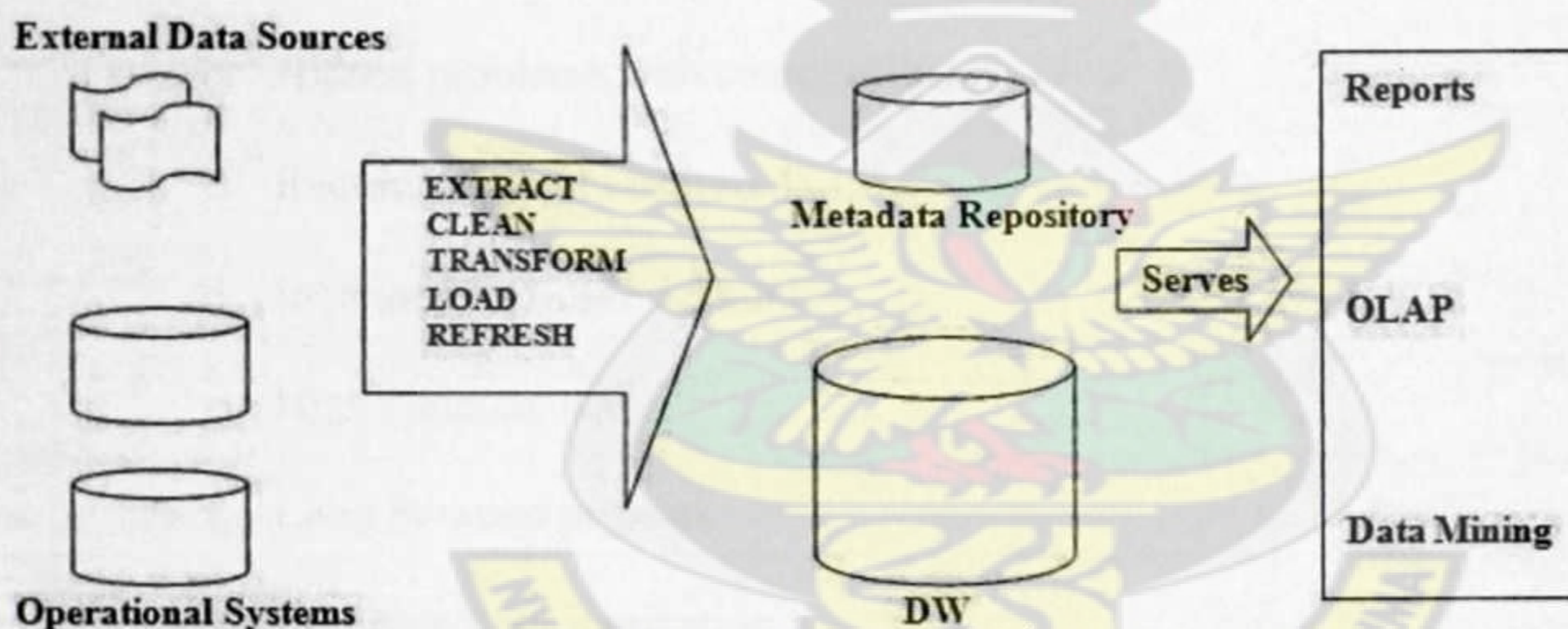


Figure 1.1: Data Warehouse Architecture [source: tech-faq (2012)]

1.8.1 Advantages of Data Warehousing

Data warehouse has gained popularity and has been hyped due to these advantages since 1995 (Galhards et al., 2000). According to Tec-faq (2012), the following are some of the advantages of data warehouse:

- Potential high Return on Investment

- Competitive Advantage
- Increased Productivity of Corporate Decision Makers

1.8.2 Problems with Data Warehousing

As much as data warehouse concepts have been widely accepted by enterprises and businesses due to its advantages (analysis driven, subject-oriented, holds holistic data and many more), it also comes with some associated problems. Among these are as discussed by Tech-faq (2012):

- Underestimation of resources for data loading
- Hidden problems with source systems
- Required data not captured
- Increased end-user demands
- High maintenance
- Long duration projects
- Complexity of integration

1.9 Definition of Data Cleansing

Data cleaning, also called data cleansing or scrubbing, deals with detecting and removing errors and inconsistencies from data in order to improve the quality of data. Data quality problems are present in single data collections, such as files and databases (Navathe and Elmasri, 2004). It is deeply domain-specific.

Data Cleaning (DC) is one of the critically important steps in DQ improvement programs in both the DW and Data Mining (DM) processes. It is used to improve the quality of data by

using various techniques. Many research works use the methods of Data Mining in the DC programs. Missing and incomplete values of data represent especially difficult to build good knowledge and draw any conclusions (Kim et al., 2003; Maletic and Marcus, 2010).

Data quality problems are quite trivial, complex and inconsistent. There is no international common standard for reference. So the process of data cleaning vary from domain to domain but basically a process used to determine inaccurate, incomplete, or unreasonable data and then improving the quality through correction of detected errors and omissions (Elmasri et al., 2004).

Organizations accumulate much data that they want to access and analyze as a consolidated whole. However the data often has inconsistencies in schema, formats, and adherence to constraints, due to many factors including data entry errors and merging from multiple sources (Mallach, 2000; Ali and Warraich, 2010).

These processes usually result in flagging, documenting and subsequent checking and correcting of suspect records. Validation checks may also involve checking for compliance against applicable standards, rules, and conventions (Elmasri et al., 2004). In data cleaning various methods are employed to find and detect errors, omissions and inconsistencies in data.

1.10 Data Cleaning Background

There are many issues in data cleansing that researchers are attempting to tackle. Of particular interest here, is the search context for what is called in literature and the business world as "dirty data" (Mallach, 2000; Vassilaids, 2009; Nemani and Konda, 2011).

Recently, Kim et al., (2003) proposed taxonomy for dirty data. It is a very important issue that will attract the attention of the researchers and practitioners in the field. It is the first step in defining and understanding the data cleansing process.

There is no commonly agreed definition of the data cleansing. Various definitions depend on the particular area in which the process is applied. The major areas that include data cleansing as part of their defining processes are: data warehousing, knowledge discovery in databases (also termed data mining), and data/information quality management (e.g., Total Data Quality Management) (Rahm and Do, 2000; Kumar and Chadrsekran, 2011).

According to Herbert and Wang (2007), Data cleaning addresses the issues of detecting and removing errors and inconsistencies from data in order to improve the quality of the data. In general, the architecture of DC process involves five phases. These steps according to Hernandez and Stolfo (1998) are as follows:

- Data Analysis in order to detect the inconsistencies;
- Definition and choice of transformations;
- Verification of the correctness and effectiveness of transformations;
- Execution of the transformation;
- Feedback of cleaned data.

In data warehousing, data cleaning becomes relevant when data is drawn from several databases to be merged at one point. There are a number of ways to represent or refer to the same entity in different data formats. Erroneous data in databases can also be reported and captured in a data warehouse.

Data cleaning has three components: auditing data to find discrepancies, choosing transformations to fix these, and applying the transformations on the dataset. They come in two

forms: auditing tools and transformation tools. The data often has many hard-to-find special cases, so this process of auditing and transformation must be repeated until the “data quality” is good enough. This approach has two problems – *lack of interactivity and need for much user efforts* (Naumann, 2002).

1.11 High Quality Data

Data can be described as high quality when it satisfies a set of criteria. In general, quality is defined as an aggregated value over a set of quality criteria (Wang, 1995). High quality of data warehouse is a key to make smart strategic decisions. Hence, correctness of data is essential for well-informed and reliable decision making (Galhards et al., 2001).

The data cleaning is a program that performs to deal with the quality problems of data extracted from operational sources before their loading into data warehouse (Hernandez and Stolfo, 1998). This section will discuss the set of data quality criteria. According to Guide (2002), data quality is data that is fit for use by data consumers.

According to Lin and Haug (2008), data cleaning addresses the issues of detecting and removing errors and inconsistencies from data in order to improve the quality of the data. Data quality has many attributes.

According to Bradji and Boufaida (2011); Kimbal (1996), an aggregated value over the criteria of integrity, consistency and density is referred to as accuracy. Intuitively this describes an exact, uniform and complete representation of data collection not containing any defined anomalies except for duplicates Kimbal (1996).

Integrity is an aggregated value over the criteria of completeness and validity and completeness is achieved by correcting data containing anomalies which occurs when all the values for a certain variable are recorded (Guide, 2012).

Elmasri and Navathe (2004) defined validity as being approximated by the amount of data satisfying integrity constraints and consistency as contradictions and syntactical anomalies. Also, Bradji and Boufaida (2011) define the uniformity attribute to be directly related to irregularities and density to be the quotient of missing values in the data and the number of total values ought to be known. Finally, uniqueness is explained by Elmasri and Navathe (2004) as related to the number of duplicates in the data.

1.12 Data cleaning process perspective

According to Rahm and Do (2000 as cited in Muller and Freytag (2000), data is audited with the use of statistical methods to detect anomalies and contradictions. This eventually gives an indication of the characteristics of the anomalies and their locations.

The detection and removal of anomalies is performed by a sequence of operations on the data known as the workflow. It is specified after the process of auditing the data and is crucial in achieving the end product of high quality data (Galhardas et al., 2001 as cited in Muller and Freytag, 2000). In order to achieve a proper workflow, the causes of the anomalies and errors in the data have to be closely considered. If for instance we find that an anomaly is a result of typing errors in data input stages, the layout of the keyboard can help in manifesting possible solutions (Rahm and Do, 2000).

The workflow is executed after its specification is complete and its correctness is verified. The implementation of the workflow should be efficient even on large sets of data

which inevitably poses a trade-off because the execution of a data cleansing operation can be computationally expensive (Muller and Freytag, 2000).

After executing the cleansing workflow, the results are inspected to verify correctness. Data that could not be corrected during execution of the workflow are manually corrected if possible. The result is a new cycle in the data cleansing process where the data is audited again to allow the specification of an additional workflow to further cleanse the data by automatic processing (Muller and Freytag, 2000).

1.13 The Existing System

There is only one company in Ghana who has developed an in-house data warehouse infrastructure to support its operations. The system is briefly discussed in the next section.

1.13.1 Millicom Ghana Limited

Millicom Ghana Limited is a telecommunication company in Ghana. It is popularly called '*Tigo*'. According to the Data warehouse report (2011), there is an in-house developed system (data warehouse) for the following purposes:

- Loading, Processing and storing Call Detail Records (CDR) and other relevant data,
- Data mining,
- Developing various reports,
- Publishing reports to a Reporting Portal for the perusal of management and to aid various staff in decision making,

- To present accurate, timely and relevant data for SOX Control execution and verification.

The system consists of modules and packages which collect all decoded files or CDRs from Callgate and all nodes on:

- Mobile Switching Centres,
- Intelligent Network,
- Short Messaging Service Centres,
- General Packet Radio Service & Wireless Application Protocol.

1.13.1.1 Critique of Millicom Ghana Framework

From the review of this system, it was realized that it had some strengths and weaknesses.

1.13.1.1.1 Strengths

- The data cleansing tool only works on call data record
- It does comparative analysis of file logs to generate report on the status of files

1.13.1.1.2 Weaknesses

- The system does not have a cleansing system. Therefore, the company fall on individuals to clean the errors on daily basis.
- The system does not remove duplicated data automatically
- The system does not support any other file format
- It does not have an interactive interface.

1.14 Organization of the Thesis

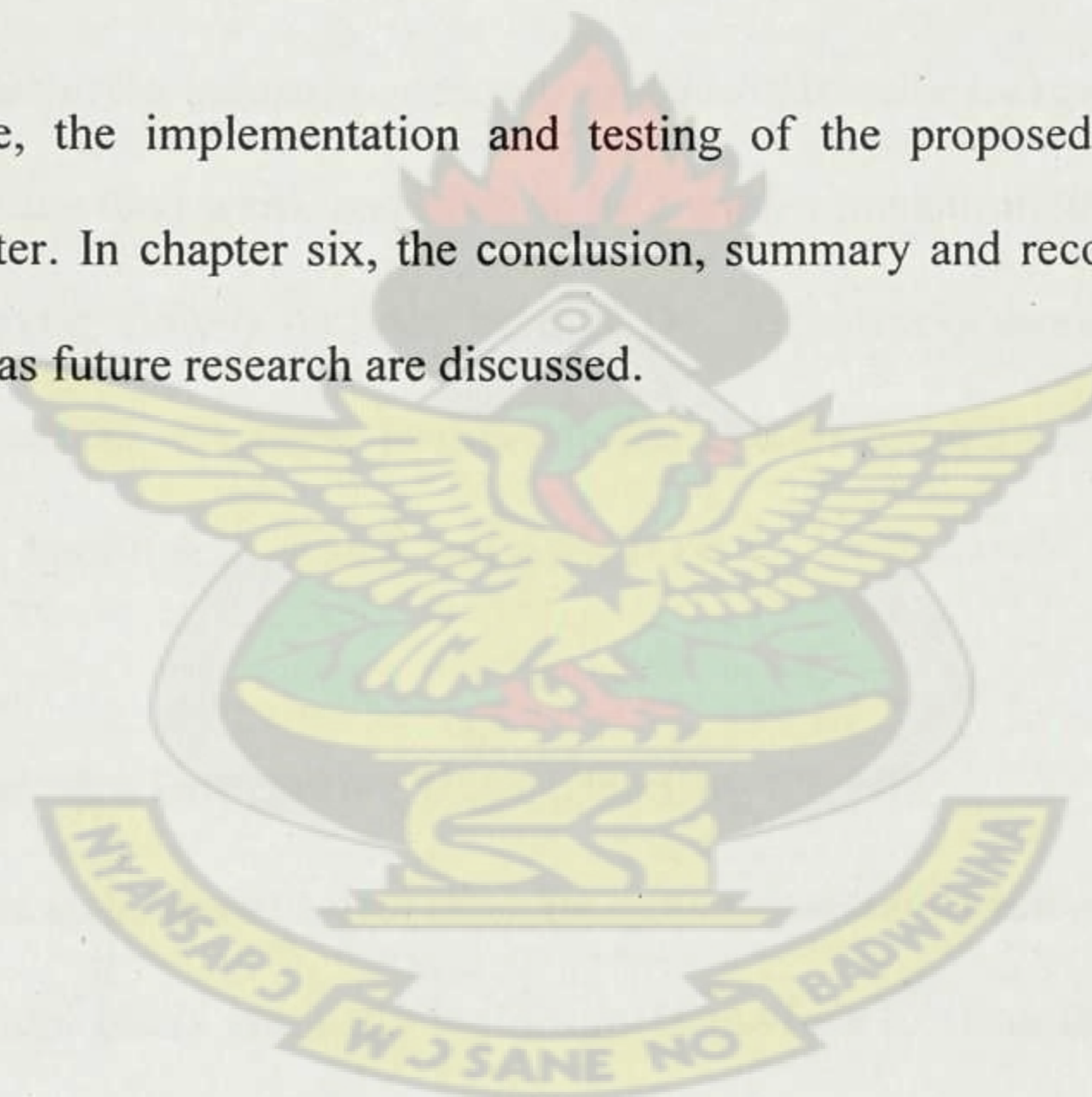
The thesis is organized into six (6) chapters. Chapter one entails the general introduction concerning the background of the study, statement of the problem, research objectives (both

general and specific), and significance of the study, scope and limitations of the study, definition of terms and organization of the chapters.

Chapter two involves a study of literature on data cleansing methods and techniques. This gives detailed information of some existing data cleansing methods used for cleansing dirty data from databases and presents the research framework for the study.

Chapter three explains the methodological approach adopted to collect data and to reach research findings. Chapter four focuses on analysis and results based on the proposed framework.

In chapter five, the implementation and testing of the proposed algorithm will be discussed in this chapter. In chapter six, the conclusion, summary and recommendation of the research paper as well as future research are discussed.



Chapter two

Literature Review

2.1 Review of Existing Approaches for Data Cleaning

Cleaning data of errors in structure and content is important for data warehousing and integration. Current solutions for data cleaning involve many iterations of data “auditing” to find errors, and long-running transformations to fix them. There are various approaches used in cleaning data in manufacturing industries, schools/colleges/universities, organizations and many more. Users need to endure long waits, and often write complex transformation scripts.

Muller and Freytag (2000) outlined several DC frameworks among which the most popular ones are reviewed in this paper in order to know their strengths and weakness. These include Potter’s Wheel, IntelliClean, AJAX and ARKTOS.

2.1.1 Potter’s Wheel: An Interactive Data Cleaning System

Potter’s Wheel is an interactive data cleaning system that tightly integrates transformation and discrepancy detection. Users gradually build transformations to clean the data by adding or undoing transforms on a spreadsheet-like interface; the effect of a transform is shown at once on records visible on screen (Raman and Hellerstein, 2001 as cited in Muller and Freytag).

These transforms are specified either through simple graphical operations, or by showing the desired effects on example data values. In the background, Potter’s Wheel automatically infers structures for data values in terms of user-defined domains, and accordingly checks for

constraint violations. Thus users can gradually build a transformation as discrepancies are found, and clean the data without writing complex programs or enduring long delays (Naumann, 2002).

2.1.1.1 Potter's Wheel: Design Goals

- Eliminate wait time during each step
- Eliminate programming, but keep user "in the loop"
- Unify detection and transformation
- Extensibility

The architecture below represents the way data flows in Potter's Wheel architecture.

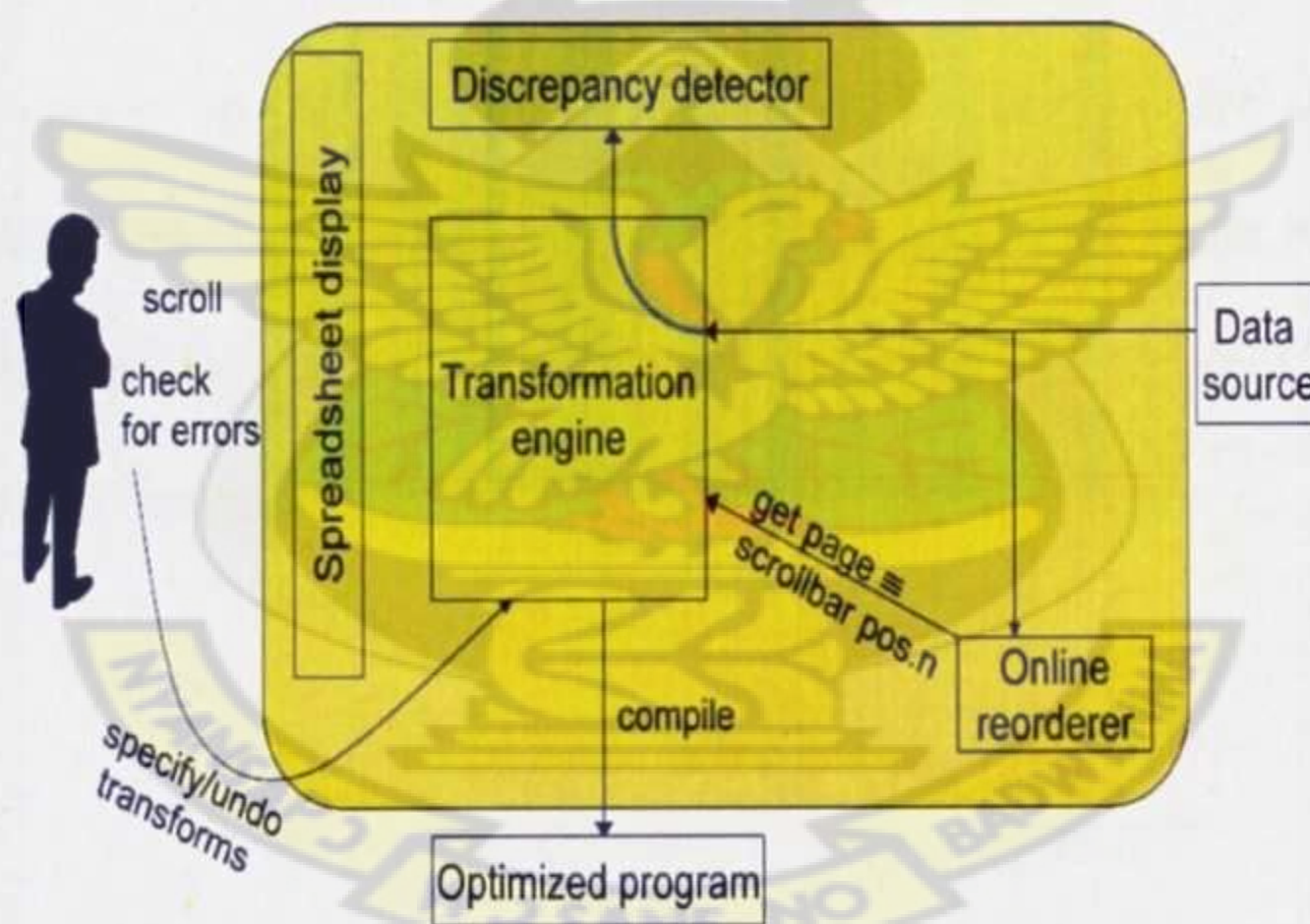


Figure 2.1: Potter's Wheel Architecture (Source: Raman and Hellerstein, 2001)

The main components of the Potter's Wheel architecture (Figure 2.1) are a Data Source, a Transformation Engine that applies transforms along 2 paths, an Online Reorderer to support interactive scrolling and sorting at the user interface, and an Automatic Discrepancy Detector (Raman and Hellerstein, 2001).

2.1.1.2 Data Source

Potter's Wheel accepts input data as a single, pre-merged stream, which can come from an ODBC source or any ASCII file descriptor (or pipe). The ODBC source can be used to query data from DBMSs, or even from distributed sources via middleware.

When reading from ASCII files, each record is viewed as a single wide column. The user can identify column delimiters graphically and split the record into constituent columns.

2.1.1.3 Interface used for Displaying Data

Data read from the input is displayed on a Scalable Spreadsheet interface that allows users to interactively re-sort on any column, and scroll in a representative sample of the data, even over large datasets (Chaudhuri and Dayal, 1997). When the user starts Potter's Wheel on a dataset, the spreadsheet interface appears immediately, without waiting until the input has been completely read. This is important when transforming large datasets or never-ending data streams.

Delay	Carrier	Num	Date	Day	Deot Sch	Deot Act	Arr_Sch	Arr_Act	Status	Random
-60	AMERICAN	0185					20:45	19:45	NORMAL	102203
-49	AMERICAN	0701					21:18	20:29	NORMAL	100402
-46	AMERICAN	0527	ORD	SAT	1997/12/13	06:00	22:02	21:16	NORMAL	10192
-45	AMERICAN	0779	ORD	TUS	1998/03/22	Su	14:49	14:04	NORMAL	102447
-44	DELTA	0035	JFK to DFW		1997/09/04	Th	15:35	15:32	NORMAL	100555
-41	AMERICAN	0194	SFO to BOS		1998/06/28	Su	15:30	15:30	NORMAL	100189
-41	UNITED	0629	ORD	OAK	1997/02/08	Sa	19:45	19:45	NORMAL	102020
-41	AMERICAN	0194	SFO	BOS	1998/09/24	Th	15:30	15:27	NORMAL	100708
-41	AMERICAN	1893	ORD	LAX	1998/04/27	M	16:45	16:43	NORMAL	100891
-39	AMERICAN	1891	ORD	LAX	1998/02/20	F	13:10	13:08	NORMAL	100860
-37	AMERICAN	2015	ORD	PHX	1998/04/27	M	17:35	17:34	NORMAL	102172
-35	AMERICAN	1475	ORD	PSP	1998/03/20	F	15:00	14:56	NORMAL	100463
-35	AMERICAN	1389	ORD to SJC		1997/06/03	Tu	17:30	17:27	NORMAL	100402
-35	AMERICAN	0827	ORD	SFO	1998/09/11	F	08:55	08:55	NORMAL	10070
-35	AMERICAN	1545	ORD	SFO	1997/08/19	Th	20:45	20:40	NORMAL	101379
-35	TWA	0741	JFK	SEA	1997/06/04	W	18:25	18:24	NORMAL	102386
-35	UNITED	0180	SFO to BOS		1998/06/29	M	16:15	16:13	NORMAL	101654
-34	DELTA	1104	JFK	ATL	1998/10/12	M	06:25	06:22	NORMAL	101684
-34	UNITED	0124	SFO	IAD	1997/07/01	Tu	16:30	16:29	NORMAL	101806
-34	AMERICAN	0721	ORD to LAS		1998/02/20	F	19:15	19:14	NORMAL	10192
-33	DELTA	0035	JFK	DFW	1997/09/08	M	15:35	15:26	NORMAL	102325
-31	AMERICAN	0417	ORD	MSY	1998/11/23	M	18:40	18:37	NORMAL	101623
-30	UNITED	0477	ORD	SAN	1997/03/25	Tu	12:14	12:12	NORMAL	101470
-30	UNITED	0129	ORD	SFO	1998/02/20	F	15:35	15:34	NORMAL	10070
-30	CONTINE...	0024	SFO to EWR		1998/08/22	Sa	12:15	12:13	NORMAL	100433
-30	NORTHW...	0928	SFO to MSP		1998/12/24	Th	06:30	06:25	NORMAL	100860
-29	DELTA	0511	ORD	ATL	1998/10/25	Su	06:10	06:09	NORMAL	100128
-29	AMERICAN	1523	ORD	SLC	1998/08/12	W	08:35	08:32	NORMAL	102081
-28	AMERICAN	0059	JFK to SFO		1997/10/17	F	08:00	07:57	NORMAL	101684

Figure 2.2: Snapshot of Potter's Wheel user interface

When the user scrolls to a new region, the reorderer picks a sample of tuples from the bucket corresponding to the scrollbar position and displays them on screen. Thus, users can explore large amounts of data along any dimension. Exploration helps users' spot simple discrepancies by observing the structure of data as values in the sort-column change. The Potter's Wheel user interface is shown in figure 2.2 above.

2.1.1.4 Transformation Engine

Transforms specified by the user need to be applied in two scenarios. First, they need to be applied when records are rendered on the screen. With the spreadsheet user interface this is done when the user scrolls or jumps to a new scrollbar position. Second, transforms need to be applied to records used for discrepancy detection because as argued earlier, we want to check for discrepancies on transformed versions of data.

2.1.1.5 Automatic Discrepancy Detector

While the user is specifying transforms and exploring the data, the discrepancy detector runs in the background, applying appropriate algorithms to find errors in the data. The discrepancy detector first parses values in each field into sub-components according to the structure inferred for the column as cited by Muller and Freytag (2000).

Then suitable algorithms are applied for each sub-component, depending on its domain. For example, if the structure of a column is <number><word><time> and a value is 19 January 06:45, the discrepancy detector finds 19, January, and 06:45 as sub-components belonging to the <number>, <word>, and <time> domains, and applies the detection algorithms specified for those domains.

2.1.1.6 Domains in Potter's Wheel

Domains in Potter's Wheel are defined through the interface shown code segment below (Figure 2.3). The only function required to be implemented is an inclusion function match to identify values in the domain.

```
public abstract class Domain {    /** Required Inclusion Function— Checks
    if value satisfies domain constraints. */

    public abstract boolean match(char *value);    /** Optional function – finds
    the number of values in this domain with given length. This could vary based on
    parameterization. */

    public int cardinality(int length);    /** Optional function – updates any state for
    this domain using the given value. */
```



```

public void updateStats(char* value);          /** Optional function – checks if a
given value is a discrepancy, with a certain probability. Typically needs to know
the total number of tuples in the data set*/

public float matchWithConfidence(char *value, int dataSize);      /** Optional
function – checks if one pattern is redundant after another.*/

public boolean isRedundantAfter(Domain d);
}

```

2.1.1.7 Structure Extraction

A given value will typically be parseable in terms of the default and user-defined domains in multiple ways. For example, March 17, 2000 can be parsed as ξ^* , as $[A-Za-z]^* [0-9]^*$; $[0-9]^*$, or as $[achrM]^* [17]^*$; $[20]^*$, to name a few possible structures. Structure extraction involves choosing the best structure for values in a column. Formally, given a set of column values v_1, v_2, \dots, v_n and a set of domains d_1, d_2, \dots, d_m , we want to extract a suitable structure $S = ds_1 ds_2 \dots, ds_p$, where $1 \leq s_1 \dots s_p \leq m$.

2.1.1.9 Critique of Porter's Wheel framework

From the above review, it was realised that Potter's Wheel had some strength as well as some weaknesses. In this section, some of the strengths and weaknesses in Potter's Wheel are critically looked at.

2.1.1.9.1 Strengths

- i. Potter's Wheel is an interactive system for data transformation and cleaning. By integrating discrepancy detection and transformation, Potter's Wheel allows users to gradually build a transformation to clean the data by adding transforms as discrepancies are detected. Users can specify transforms through graphical operations or through examples, and see the effect instantaneously, thereby allowing easy experimentation with different transforms.
- ii. Parsing strings using structures of user defined domains result in a general and extensible discrepancy detection mechanism. Such domains provide a powerful basis for specifying Split transformations through example values.
- iii. End user power: there are graphical representations of transformations on data such that users do have a chance to add or even undo effects on transformations.

2.1.1.9.2 Weaknesses

- i. Scrolling by the end-user is directly related to the scope of the automatic discrepancy detection. Hence if the end user does not scroll up to that record, a particular discrepancy may be available but will not be detected.
- ii. Data to be tested for errors is fetched in samples of the source data; however, the sequence of the fetch remains unknown. That is whether the fetch is done sequentially or randomly or by this implies that if there are errors in the data. The situation of random may call for the third situation as described in point 'a'.
- iii. Delay of time – as stated in the second concept, it is possible that same sampled data could be fetched and retested over and over again. When this situation occurs, greater time will be used in detecting discrepancies within the data warehouse
- iv. Although the user interface is said to be interactive, however, its effectiveness was not measured. Therefore, an appropriate tool such as an interactive querying system needs to be used to measure the effectiveness of the user interface.

2.1.2 IntelliClean: A Knowledge-Based Intelligent Data Cleaner

Intelliclean is a knowledge-based Intelligent Data cleaner. The system proposes a generic knowledge-based framework for effective data cleaning that implements existing cleaning strategies. Intelliclean employs a new method to compute transitive closure under uncertainty

which handles the merging of groups of inexact duplicate records. Experimental results show that this framework can identify duplicates and anomalies with high recall and precision.

2.1.2.1 Conceptual model and benchmarking metrics for data cleaning

The model starts by receiving “dirty” datasets with variety of errors. Cleaning strategies are applied to the dataset with the objective of obtaining consistent and correct data as the output. The effectiveness of the cleaning strategies will thus be the degree by which data quality is improved through cleaning. Figure 2.4 below shows the conceptual model of Intelliclean.

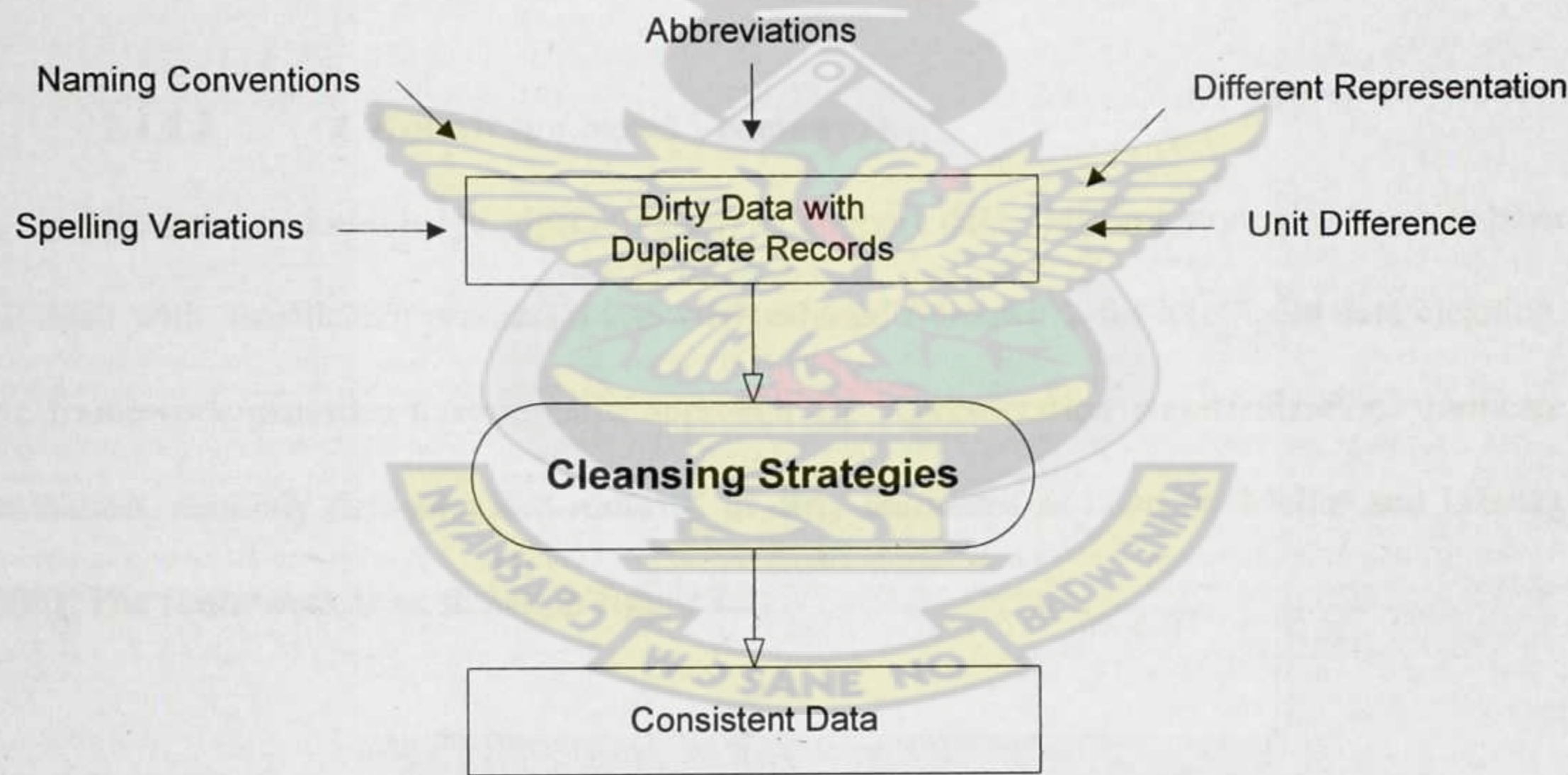


Figure 2.4: An Overall Conceptual model of Intelliclean

Two metrics that benchmark the effectiveness of data cleaning strategies are defined.

Recall: This is also known as percentage hits. It is defined as the percentage of duplicate records being correctly identified.

False-Positive Error: This is the antithesis of the precision measure and sometimes referred to as false merges. It is defined as the percentage of records wrongly identified as duplicates, i.e.

KNUST

$$\frac{\text{No. of wrongly identified duplicates}}{\text{Total no. of identified duplicates}} * 100\%$$

2.1.2.2 A Knowledge-based Framework

The issue of knowledge representation to support data cleaning strategies has not been well dealt with. Intelliclean presents a knowledge-based framework for intelligent data cleaning. This framework provides a systematic approach for representation standardization, duplicate elimination, anomaly detection and removal in dirty databases as cited by Muller and Freytag (2000). The framework is as shown in figure 2.5.

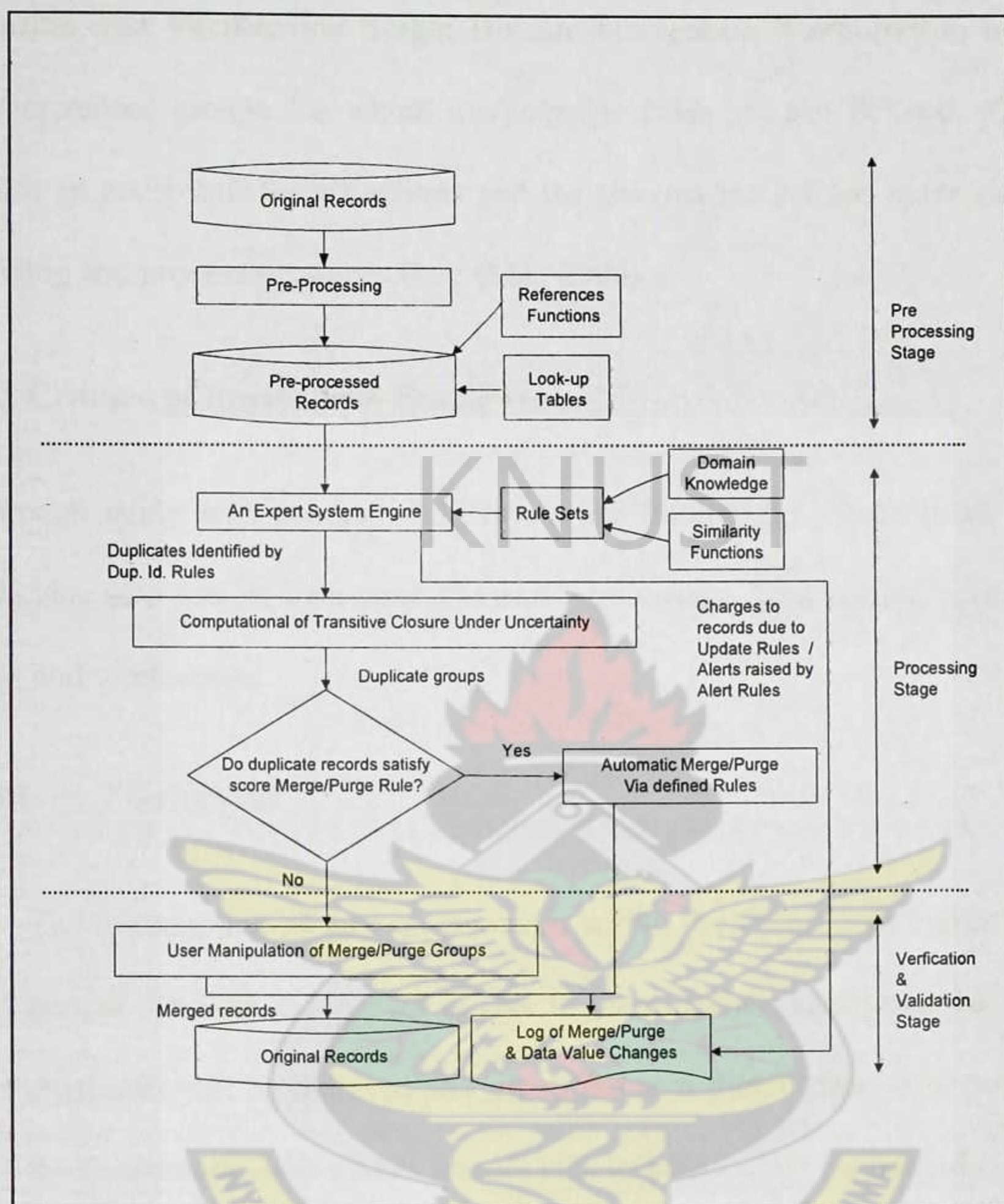


Figure 2.5: A Knowledge-based Cleaning framework cited by Lee et al., (2000)

The framework consists of three stages:

- **Pre-processing Stage:** Data records are first conditioned and scrubbed of any anomalies we can detect at this stage. Data type checks and format standardization can be performed (e.g. 01/1/2000, 1st January 2000 can be standardized to one format).
- **Processing Stage:** Conditioned records are next fed into an expert system engine together with a set of rules. Each rule will fall into one of the following categories; Duplicate Identification Rules. Merge and purge rules, update rules and alert rules.

- **Validation and Verification Stage:** Human intervention is required to manipulate the duplicate record groups for which merge/purge rules are not defined. The log report provides an audit trail for all actions and the reasons for actions made during the pre-processing and processing stages (Lee et al., 2000)

2.1.2.3 Critique of IntelliClean Framework

A thorough study was conducted in reviewing Intelliclean. Since there is no perfect system, Intelliclean also has its own strengths and weaknesses. This section considers some of these strengths and weaknesses.

2.1.2.3.1 Strengths

- i. The introduction of an expert system within the framework makes it much more generic because the system is able to learn. Hence, changes made in the textual databases will be detected and identified as a transformed data automatically by the framework. This makes it more efficient than other frameworks.
- ii. It provides effective rules for resolving the recall-precision dilemma which is a short coming in other frameworks; hence, it enhances the effectiveness of the framework.

2.1.2.3.2 Weaknesses

- i. The IntelliClean considered only textual forms of data in the data warehouse, implying that duplicates of data of other formats will not be identified. For

example, image, video, graphics and other file formats are not applicable with the IntelliClean.

- ii. Data source from the web is not applicable to the Intelliclean system. It was not considered as a source. Therefore, a generic and data source independent system when developed can resolve this problem.

2.1.3 ARKTOS: A Tool For Data Cleaning and Transformation in Data Warehouse Environments

ARKTOS as cited by Vassiliaadis et al., (2001) is a framework capable of modelling and executing the Extraction-Transformation-Load process (ETL process) for data warehouse creation. The authors consider data cleansing as an integral part of this ETL process which consists of single steps that extract relevant data from the sources, transform it to the target format and cleanse it, then loading it into the data warehouse.

A meta-model is specified allowing the modeling of the complete ETL process. The single steps (cleansing operations) within the process are called activities. Each activity is linked to input and output relations.

The logic performed by an activity is declaratively described by a SQL-statement. Each statement is associated with a particular error type and a policy which specifies the behavior (the action to be performed) in case of error occurrence.

Six types of errors can be considered within an ETL process specified and executed in the ARKTOS framework. PRIMARY KEY VIOLATION, UNIQUENESS

VIOLATION and REFERENCE VIOLATION are special cases of integrity constraint violations. The error type NULL EXISTENCE is concerned with the elimination of missing values. The remaining error types are DOMAIN MISMATCH and FORMAT MISMATCH referring to lexical and domain format errors as cited by Muller and Freytag (2000).

The policies for error correction simply are IGNORE, but without explicitly marking the erroneous tuple, DELETE as well as WRITE TO FILE and INSERT TO TABLE with the expected semantics. The latter two provide the only possibility for interaction with the user.

The success of data cleansing can be measured for each activity by executing a similar SQL statement counting the matching/violating tuples. The authors define two languages for declaratively specifying of the ETL process. This is accompanied with a graphical scenario builder.

2.1.3.1 Critique of ARKTOS framework

2.1.3.1.1 Strengths

- i. The Arktos, is capable of modeling and executing practical ETL scenarios by providing explicit primitives for the capturing of common tasks (like data leaning, scheduling and data transformations).
- ii. The system makes it easier for authoring by providing three ways to describe an ETL scenario: a graphical point-and-click front end and two declarative languages: XADL

(an XML variant), which is more verbose and easy to read and SADL (an SQL-like language) which has a quite compact syntax.

2.1.3.1.2 Weakness

- i. Although the research could identify the optimization problems (Identification of a small set of algebraic operators, Local optimization of ETL activities and, Global (multiple) optimization of ETL activities) of ETL processes but was silent on the solution to the identified problems.

2.1.4 AJAX

According Galhards et al., (2000), Muller and Freytag (20001) and Wang and Herbert (2007), AJAX is an extensible and flexible framework attempting to separate the logical and physical levels of data cleansing. The logical level supports the design of the data cleansing workflow and specification of cleansing operations performed, while the physical level regards their implementation.

The main goal of AJAX is to facilitate the specification and execution of data cleaning programs either for a single source or for integrating multiple data sources (Galhards et al., 2000). In this paper the main issue is transforming existing data from several data collections into a target schema and eliminating duplicates within this process.

In order to achieve this, five major transformations are discussed in the paper. These transformations are mapping, viewing, matching, clustering, and merging. According to Galhards et al., (2000), the mapping transformation standardizes all data formats and simply

produces a more appropriate data format by applying two major operations such as splitting and merging. The matching compares several records and finds pairs that match specified criteria. Another transformation called clustering groups together matching pairs with high similarity by applying a given group criteria. The merging transformation is then applied to each of these clusters in order to eliminate duplicates or produce a new record based on the integrated results.

AJAX tends to be more complex as compared to other approaches.

2.1.4.1 Critique of AJAX

2.1.4.1.1 Strengths

- i. It allows customization because it is extensible. For example extension of functions from other libraries, combination with primitives with SQL etc. all of this adds dynamism to the functionality of the AJAX.
- ii. Interactivity of system is high.

2.1.4.1.2 Weaknesses

- i. There is higher level of dependency on human expert for the resolution of exceptional cases that arise as a result of executing a micro-operator. This is a demanding situation as compared with the IntelliClean system which has a an expert module which caters for such kinds of exception
- ii. The approach used in creating quality data (cleaned data) is very complex as compared to other frameworks. This will supposedly suggest that, it will take greater time to clean data as compared with other frameworks.

Table 2.1: Comparative analysis of Data Cleaning Frameworks

Parameter	Porters Wheel	AJAX	IntelliClean	ARKTOS
Interactivity	It is very interactive; hence easy to use	Complex interface and hence not friendly to non-technical persons.	Interactive with end user. However, requires little input from end-users	Highly interactive; it has graphical interfaces for loading and executing validations on loaded files.
Data format/Structure	Text	Text	Text	Text
Human dependency	High human dependency for exceptional errors.	High human dependency. Example; evaluation and validation of errors are fully dependent on human expert.	Very minimal because of the expert module embedded in the system.	Although the system has complex modules for dealing with duplicates, however, there is a high dependency on human experts for error correction.
Maintenance	Not considered	Not considered	Not considered	Not considered

After the thorough criticizing of the data cleansing frameworks, a comparative analysis was conducted to compare the four data cleansing frameworks basing on four parameters as shown in table 2.1 above. It was realized that none of the frameworks considered the maintenance aspect of the data cleansing process.

2.2 Reviewed Papers on Data Cleaning

There has been quite a number of works and researches done on data cleaning with its application to enterprise data warehouse as well as data marts for operational systems for many organizations. This research reviews some of them with special reference to the methods used and the frameworks proposed to clean dirty data in operational databases from different sources.

2.2.1 Problems, Methods, and Challenges in Comprehensive Data Cleansing

In this paper, Muller and Freytag (2000) classified data quality problems into syntactical anomalies which concern data formats and values for data representation (e.g. lexical errors, domain format errors and irregularities). Lexical errors name discrepancies between the structure of data items and the specified format. Domain errors specify where the given value for an attribute A does not conform with the anticipated domain format $G(dom(A))$. Irregularities deal with non-uniformed use of values and other abbreviations which normally are noticeable when different currency format is used to specify employee salary.

Müller and Freytag (2000) also discussed the Semantic anomaly which hinders data collection from being comprehensive and non-redundant representation of the world such as integrity constraints, contradictions, duplicates and invalid tuples. Finally, Coverage anomaly is discussed by the authors in regard to data quality problems. Here the amount of entities and their properties are represented as missing values and missing tuples from real world data collections. In the work of Müller and Freytag no appropriate framework was designed to support the cleansing process.

2.2.1.1 Critique of Müller and Freytag's paper

2.2.1.1.1 Strength

The research was able to identify several problems and challenges such as the inability of researchers to state the details of the implementation of cleaning in comprehensive data cleaning. This will help many researchers to be able look into the identified problems and come up with solutions appropriately.

2.2.1.1.2 Weaknesses

- i. Maintenance is not considered in the framework for cleaning of data. Hence after cleaning of data, the question is what next? Data needs to be maintained.
- ii. Although several cleaning tools were compared in terms of the file formats that they can clean, however, how effective they do the cleaning was swept under the carpet. The effectiveness of these tools should be clearly known.

2.2.2 Data Cleaning: Problems and Current Approaches

According to Rahm and Do (2000), the classification of data quality problems can be divided into two main categories: single-source and multiple-source problems. At the single-source, Rahm and Do divide these into schema level and instance level related problems without considering the occurrence in a single relation. The single-source problems deal with attribute, record, record type and source whereas the multiple-source problems deal with naming conflicts, schema-level conflicts and the identification of overlapping data which refers to same real-world entity.

2.2.2.1 Critique of Rahm and Do's paper

2.2.2.1.1 Strengths

- i. The research clearly provided a classification of data quality problems in data sources differentiating between single- and multi-source and between schema- and instance-level problems.
- ii. The research also, further outlines the major steps for data transformation and data cleaning and emphasized the need to cover schema- and instance-related data transformations in an integrated way.

- iii. The research provided an overview of commercial data cleaning tools. It was identified that, while the state-of-the-art in these tools is quite advanced, they do typically cover only part of the problem and still require substantial manual effort or self-programming.

2.2.2.1.2 Weaknesses

- i. The research work is insufficient of the design and implementation details of the best language approach for supporting both schema and data transformations. For instance, operators such as Match, Merge or Mapping Composition have either been studied at the instance (data) or schema (metadata) level but may be built on similar implementation techniques.
- ii. Data cleaning is not only needed for data warehousing but also for query processing on heterogeneous data sources, e.g., in web-based information systems. This environment poses much more restrictive performance constraints for data cleaning that need to be considered in the design of suitable approaches; however, they were untouched in this work.

2.2.3 Matching Algorithms with a Duplicate Detection System

According to Monge (2000), detecting database records that are approximate duplicates, but not exact duplicates, is an important task. These duplicates concern the same real world entity due to data entry errors, abbreviations that are not standardized or differences in schema level records from multiple databases. In this paper, the approximate duplicate detection problem was explored and the following contributions were made:

- The author proposed how to compute the transitive closure of “is duplicate of” relationships incrementally by using the union-find data structure (R_i and R_j) operation where R_j is compared to R_i using the matching algorithm.
- The other contribution made by the author was the implementation of heuristic method for minimizing the number of expensive records while comparing individual records.

2.2.3.1 Critique of Monge's paper

2.2.3.1.1 Strength

The research contributed to the existing algorithms for detecting duplicates of records by introducing an integrated framework, which draws from the existing algorithms.

2.2.3.1.2 Weaknesses

The research was able to measure the effectiveness of the developed algorithm against the standard algorithm, however, with very minimal iterations. It is also identified that minimal iterations could result in inability to identify duplicate records. Therefore, there is doubt on the outcome of this research.

2.2.4 Open User Involvement in Data Cleaning for Data Warehouse Quality

The issue of data cleaning caught the attention of researchers not so long ago. According to Marcus and Maletic (2000) data cleansing is a relatively new research field. Although after Maletic and Marcus in the year 2000 who proposed a framework which other authors/researchers have looked into and proposed slight different architectures, the outcome still needs further

investigation. The process is computationally expensive on very large data sets and thus it was almost impossible to do with old technology (Bradji and Boufaida, 2011).

The new faster computers allow performing the data cleansing process in acceptable time on large amounts of data. There are many issues in the data cleansing area that researchers are attempting to tackle. They consist of dealing with missing data, determining record usability, erroneous data, etc. Different approaches address different issues (Bradji and Boufaida, 2011).

The following problems were identified in the paper:

- Data warehouse projects mostly integrate the data quality phase into the extract-transform-load (ETL) process but do not allocate enough time for efficient data validation with respect to data cleansing.
- When data has been extracted from operational data sources, feedback of data cleaning results is not taken into account even though it's important to avoid redoing the same data cleaning tasks.
- It can be realized that some erroneous data require manual intervention and others require combining the manual and automatic data cleansing approaches as cited by (Monge, 2000).

Some of the issues Bradji., *et al.*, discussed in their work was to extend the data cleaning process for users at the operational sources to validate the data, and they also considered the adaptation of ETL to support their data cleaning process.

The purpose of this research is to establish that involvement of the user in the data cleaning process for the purposes of validating the data at the operational sources before propagating the cleaned data.

Reviewing the work done in this paper, the following were realized:

- i. Data extracted from different operational sources could not be easily sent for re-validation for the fact that some of the errors could be at the schema or instance level and could not be easily taken care of.
- ii. The proposed framework only work for small amount of data being propagated and was not tested for large data sets.

2.2.4.1 Critique of Bradji., *et al.*, paper

2.2.4.1.1 Strengths

- i. The research extended the data cleaning and extract-transform-load (ETL) processes to better support the user involvement in data quality management.
- ii. Systems user interface is graphical and hence easy to be operated by end-users.

2.2.4.1.2 Weaknesses

- i. The data warehouse that was used for the testing of results was small. Huge data warehouse will bring out the exact outcome of the research.
- ii. There is an extensive involvement of users. For example, after detection of duplicates (errors), the end user is asked to validate. Continues human validation may result in other human errors in the data warehouse.

2.2.5 Three Tier level Data Warehouse Architecture for Ghanaian Petroleum Industry

According to Deku et al., (2012), data warehouse (DW) is a modern proven technique of handling and managing diversity in data sources, format and structure and he added that recent

advances in database technologies are leading to the proliferation of different kinds of information design with independent supporting hardware and software.

This technology was developed therefore to integrate heterogeneous information sources for analysis purposes. In this paper they realized that, Information sources are progressively becoming autonomous and they change their content rapidly due to perpetual transactions (data changes) and may change their structure due to continual users' requirements evolving (schema changes).

However, handling correctly all type of these changes was the real challenge the research is supposed to comprehend. Strategic decision making is an ongoing process and part of human life. Businesses all over the world are faced with challenging decision making which is the very livelihood to their very existence.

There has been intensive research in the field of databases including the grandfather of data warehousing Inmon which they offered tremendous insight into how the power of computing can be harnessed in strategic decision making. The achievement in the petroleum industry has been highlighted and written about by Nimmagadda and his fellow professionals. However, how this can be implemented in Africa remains to be solved.

In this paper, Deku et al., (2012) investigated the various data capturing systems which are currently in use at the Ghana National Petroleum Cooperation (GNPC) – a state agency responsible for the overall management and supervision of the hydrocarbon endowment of the sub-region and made recommendation that a tree tier level petroleum data warehouse architecture for cutting- edge decision making on petroleum resources should be adapted as the architecture for a data warehouse.

2.2.5.1 Critique of Deku et al., paper

2.2.5.1.1 Strength

It provides a comprehensive and effective framework for managing information of GNPC.

2.2.5.1.2 Weakness

The case for this study is narrowed and hence, the findings of this research to the other companies in the process of commencing exploratory work in their respective fields due to issues of underlying geological rock differences and the contents of such fields (that is gas or oil or both).

2.2.6 Gap Analysis

In this section, the various weaknesses are discussed and possible solutions are provided to fill the gap identified in the frameworks and concepts of cleaning data in the data warehouse as identified above.

In the review of the Porter's wheel framework, IntelliClean, ARKTOS, AJAX, the following defects were identified

- i. Scrolling by the end-user is directly related to the scope of the automatic discrepancy detection. Hence if the end-user does not scroll up to that record, a particular discrepancy might be available but will not be detected.
- ii. Data to be tested for errors is fetched in samples of the source data; however, the sequence of the fetch remains unknown. That is whether the fetch is done sequentially or randomly or by this implies that if there are errors in the data. The situation of random may call for the third situation as described in point 'a'.

- iii. Delay of time – as stated in the second concept, it is possible that same sampled data could be fetched and retested over and over again. When this situation occurs, greater time will be used in detecting discrepancies within the data warehouse
- iv. Although the user interface is said to be interactive, however, its effectiveness was not measured. Therefore, an appropriate tool such as an interactive querying system needs to be used to measure the effectiveness of the user interface.
- v. The IntelliClean considered only textual forms of data in the data warehouse, implying that duplicates of data of other formats will not be identified. For example, image, video, graphics and other file formats are not applicable with the IntelliClean.
- vi. Data source from the web is not applicable to the Intelliclean system. It was not considered as a source. Therefore, a generic and data source independent system when developed can resolve this problem.
- vii. Although the research could identify the optimization problems (Identification of a small set of algebraic operators, Local optimization of ETL activities and, Global (multiple) optimization of ETL activities) of ETL processes but were silent on the solution to the identified problems.
- viii. There is higher level of dependency on human expert for the resolution of exceptional cases that arise as a result of executing a micro-operator. This is a demanding situation as compared with the IntelliClean system which has an expert module which caters for such kinds of exception.
- ix. The approach used in creating quality data (cleaned data) is very complex as compared to other frameworks. This will supposedly suggest that it will take greater time to clean data as compared with other frameworks.

- x. Maintenance is not considered in the framework for cleaning of data. Hence after cleaning of data, the question is what next? Data needs to be maintained.
- xi. Although several cleaning tools were compared in terms of the file formats that they can clean, however, how effective they do the cleaning was swept under the carpet. The effectiveness of these tools should be clearly known.
- xii. The research was able to measure the effectiveness of the developed algorithm against the standard algorithm, however, with very minimal iterations. It is also identified that minimal iterations could result in inability to identify duplicate records. Therefore, there is doubt on the outcome of this research.
- xiii. The data warehouse that was used for the testing of results was small. Huge data warehouse will bring out the exact outcome of the research.
- xiv. There is an extensive involvement of users. For example, after detection of duplicates (errors), the end user is asked to validate. Continues human validation may result in other human errors in the data warehouse.

2.2.6.1 Summary of gap analysis

Majority of the frameworks do have a common problem of interactivity with the end-user, consideration of only textual contents of databases and hence multimedia databases, procedure for testing is undefined(no standardization), higher human dependency, hidden details of cleaning processes and lastly, maintenance lost in all the framework.

From the above discussion the following will be adopted to fill the gap.

- **User friendly interface:** A customizable user interface is developed which makes it easier for both technical and non technical users to load and clean data. Data cleaning here does not depend on scrolling area, it picks the file directly.
- **Multimedia databases:** The proposed framework will consider multimedia databases. Problems of databases do not always arise from text formats of data. Therefore, in this framework, databases consider text, image, videos and graphics.
- **Showcase of detailed cleaning process:** in the existing systems the details of the cleaning processes are not showed. However, in the proposed system, a comprehensive detail is provided.
- **Maintenance:** Periodically, the cleaned data will be automatically reloaded and cleaned. This will make it possible such that most of the time there will be a clean data in the warehouse.

2.2.7 Parameters for standard testing

From the review of the existing tools and papers, the following parameters stand out among others when it comes to the usability of the tools.

- Interactivity
- File format
- Loading rate
- Data size
- Processor speed
- Memory size

2.3 Conceptual Framework to support data cleaning process

From the above review, a proposed conceptual framework is proposed and incorporates the factors that will enhance the data cleansing process and the entire data cleansing structure within a data warehouse. The process is categorized into three parts: extract, load/clean and validate/export. Listed below are the steps involved in the entire cleansing process.

- Data is extracted from different sources (internal and external sources)
- Data type is obtained (files/multimedia databases)
- Data is then loaded and a copy is stored in a central repository (cache) and shows loading details
- An interactive graphical user interface serves as the ease of use interface
- The data is obtained from the cache and we search and remove the error instance being worked on.
- The cleaning process is applied based on the algorithm to be proposed and shows the cleaning details
- Data is validated depending on the user's perception and data warehouse is updated with clean data
- The clean data is exported to a periodic automatic cleaning for data maintenance
- The unclean data is exported to a knowledge base engine for routine checking before forwarding it to the periodic automatic cleaning and vice versa for data maintenance.

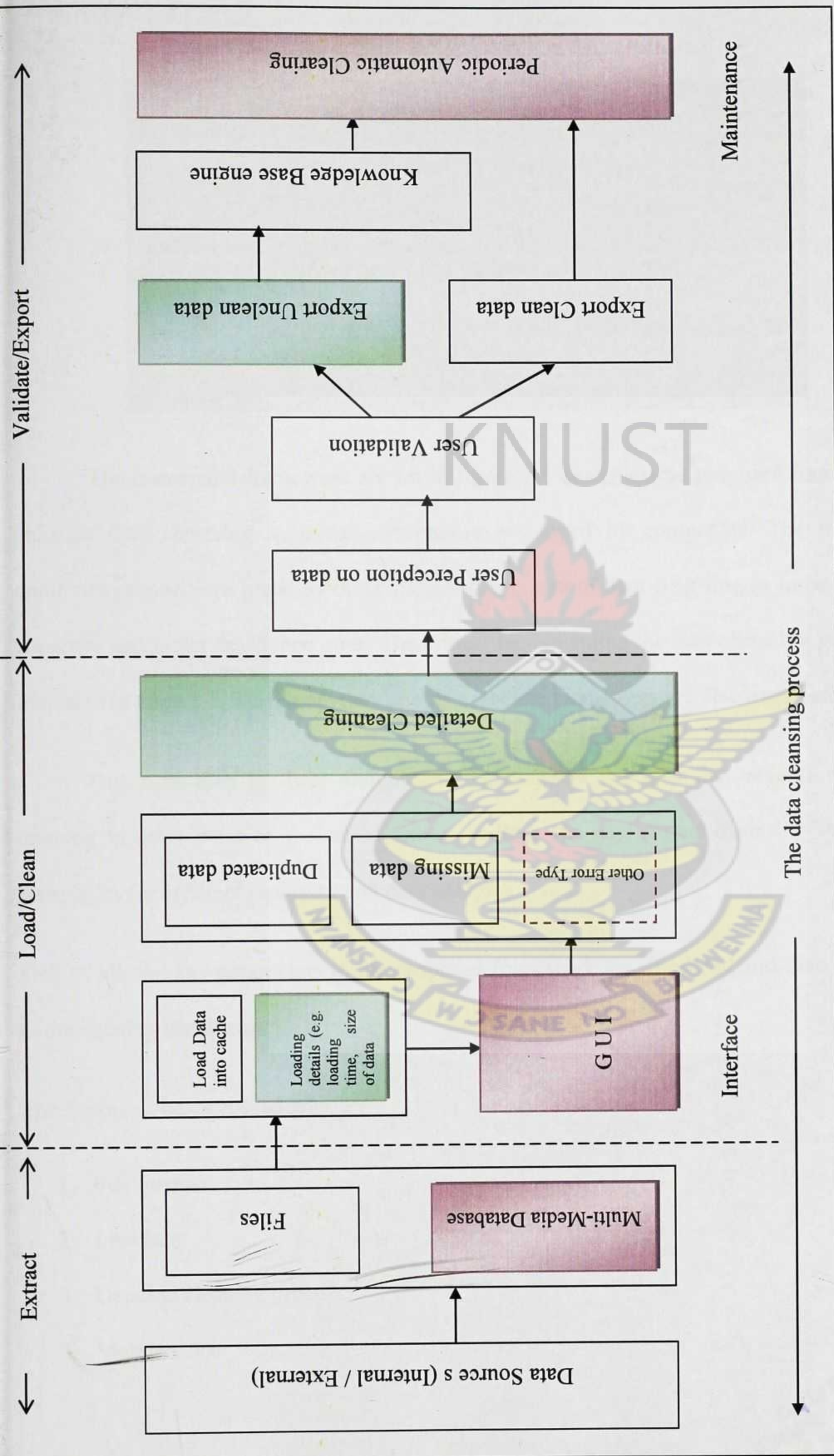


Figure 2.6: The conceptual framework

Table 2.2: Critical factors aiding in data cleansing

Key Parameters	Indicators
File format	Files
	Multimedia databases
Interface	Graphical user interface for interactivity and ease of use
Detailed cleaning process	Loading time
	Cleansing time
	Data statistics
Maintenance	Update clean data periodically

The conceptual framework shown in figure 2.6 describes the proposed framework to enhance data cleansing in a data warehouse employed by companies. The framework combines propositions made by other authors in the area of data cleansing in some advanced countries and under developed ones. The critical factors aiding the data cleansing process are indicated in table 2.2. The parameters and its associated indicators are discussed below.

Our extension to these frameworks necessitates the idea that several researches ongoing in other parts of the world are to facilitate the use of data cleansing systems by companies for efficient reporting and data analysis. .

First of all, the key parameters in the proposed framework are discussed and then compared to the existing frameworks.

The factors to be discussed are

1. File Format
2. Interface
3. Detailed cleaning process
4. Maintenance

2.3.1 File format

One of the most challenging tasks to is writing a program to consider all file formats. According to Mariappan and Parthasarathy (2009), in their paper, an analysis of data storage and retrieval of file format system, explained that in the modern GUI world the software and hardware developers are facing a challenging trouble to use different types of file formats and storage devices.

This is because there are several file formats which include image file formats (examples GIF, JPEG, PNG, TIFF), Audio File formats (examples WAV, WMA, MP3), Video File format (examples AVI, DAT, MP3), Text File format (examples PDF, HTML, RTF) and Source File format and Database file format etc Mariappan and Parthasarathy (2009). That is why we realised from literature that most of the data cleansing tools supported only text files. It is therefore necessary to research into how other file formats could be included in the data cleansing process.

2.3.2 User Interface

The user interface employed by most of the tools reviewed is not interactivity. Once the user must get involved, an interactive interface is required when building a data cleansing tool.

2.3.3 Detailed cleansing process

From the literature review, it was realised that almost all the data cleansing tools were silent on the data cleansing process. Details such as, the time it took for the cleansing to take place, number of records found with errors and the number of clean records, etc. had been given in the works reviewed in literature. This is an important aspect of the cleansing process and as such must be given attention.

2.3.4 Maintenance

One critical question that remained unanswered in literature is how the cleaned data would be maintained? The maintenance of data is important since a clean data might be corrupted when it is kept in the same repository with other files. Hence, it is an essential part of the data cleansing process. In maintenance, particular attention is given to the existing cleaned data that might become erroneous after some time when dirty data is introduced to it.

To maintain the clean data, database administrators are required to set the period for the maintenance schedule. This will enable the automatic maintenance feature to perform a routine check within the specified time frame to maintain the clean data within the central repository. Depending on the period for data analysis and period set for reporting, the periodic automatic update could be done every forth night, monthly, quarterly, or yearly. Data maintenance begins when the clean data is exported to the central repository for management to use for reporting and analyzing purposes.

Chapter Three

Methodology

3.1 Selecting a Research Methodology

The type of research undertaken by the researcher is applied since this is done to solve specific, practical question. One of the most important aspects of research is the methodology. It serves as the pivot point and much consideration are given to it. Miama (n.d), explains that qualitative research aims at understanding whereas quantitative research aims at (casual) explanation. Both qualitative and quantitative research can aim at description of social reality.

Degu and Yigzaw (2006), define qualitative research to be concerned with developing explanations of social phenomena. It aims to help us to understand the world in which we live and why things are the way they are. In their paper, they further explained it simply as finding the answers to questions which begin with why? How? Whereas quantitative research is more concerned with the questions about: how much? How many? How often?

The research methodology involves the use of both primary and secondary data but the data used specifically for this research is obtained from secondary sources. This data is relevant since it will be used to facilitate the analysis and the performance of the system.

In this research, an algorithm is presented and a system is developed using System Analysis and Design tools. Bhushan and Parikshit (2010), in their paper stated that system analysis and design is a crucial phenomenon in the development of information systems. There are various system analyses and design methodologies used in the development of the

software systems. Some of the tools that this research will adopt include data flow diagrams, entity relationship diagrams, and functional diagram.

The system will be built using the C sharp programming language. Various methods and functions and other object oriented techniques are employed in the coding principles. For the purposes of data analysis, secondary data will be obtained from Valley View University students' database to test the system for its performance. The testing of the system will also reveal the system's strength and weakness. The researcher will obtain the secondary data on a flash drive for it to be used for the testing of the system.

In this research, the first step develops and describes the conceptual framework for the data cleansing tool that is utilized to define, analyze, and provide guidance to improve data cleansing quality. In order to accomplish this step, concepts, ideas, and paradigms will be drawn from existing data cleansing framework as well as data quality community.

Secondly, the measure of success in this research will be analyzed with a specific set (s) of data in order to facilitate the outcome of the research. The system that will be constructed will have no data point but the data will be obtained from external sources and loaded. The data will be corrupted with errors for testing purposes. Thirdly, the data cleansing tool is adapted to test other files for its performance apart from the databases.

Chapter four

Analysis and Design

4.0 Analysis and Design in perspective

This chapter presents a clear analysis of the data collected from the students' database from Valley View University to analyse the performance of the application to be developed. An easy-to-use (with a well designed graphical user interface) data cleansing tool to support the cleansing process, using the C# programming language is developed as part of this research to aid in the analysis. The system is designed in such a manner as to load data from internal or external sources to particularly aid in detecting irregularities related to files and databases with any inputs size.

The basic parameters required as input into the system comprise files and databases. The system basically takes these inputs and concatenates each field into a string and searching through to detect if there exists such string within the list.

Furthermore, in this chapter the analysis of the proposed system, thus, the system to implement the algorithm based on the conceptual framework will be considered. The analysis will include both functional and non-functional requirements, stating the hardware and software requirements, input and processing requirements.

The design of the system and methods will be looked at in this chapter. This will include the design of the algorithm of the proposed system, the Unified Modeling Language (UML) diagram, the system model, and the Level 1 data flow diagram which will be based on the selected processing model.

4.1 Analysis

The feasibility study on how to achieve the development of the cleansing tool was developed. A model will be developed in order to know how the component part of the system interacts. A study and a plan were adopted to study the various requirements for the system in order to inform the researcher's decision making.

4.1.1 Presentation of data

The database consists of one hundred and twenty two (102) tables built with Microsoft SQL Server. It has over fifty thousand (50000) records.

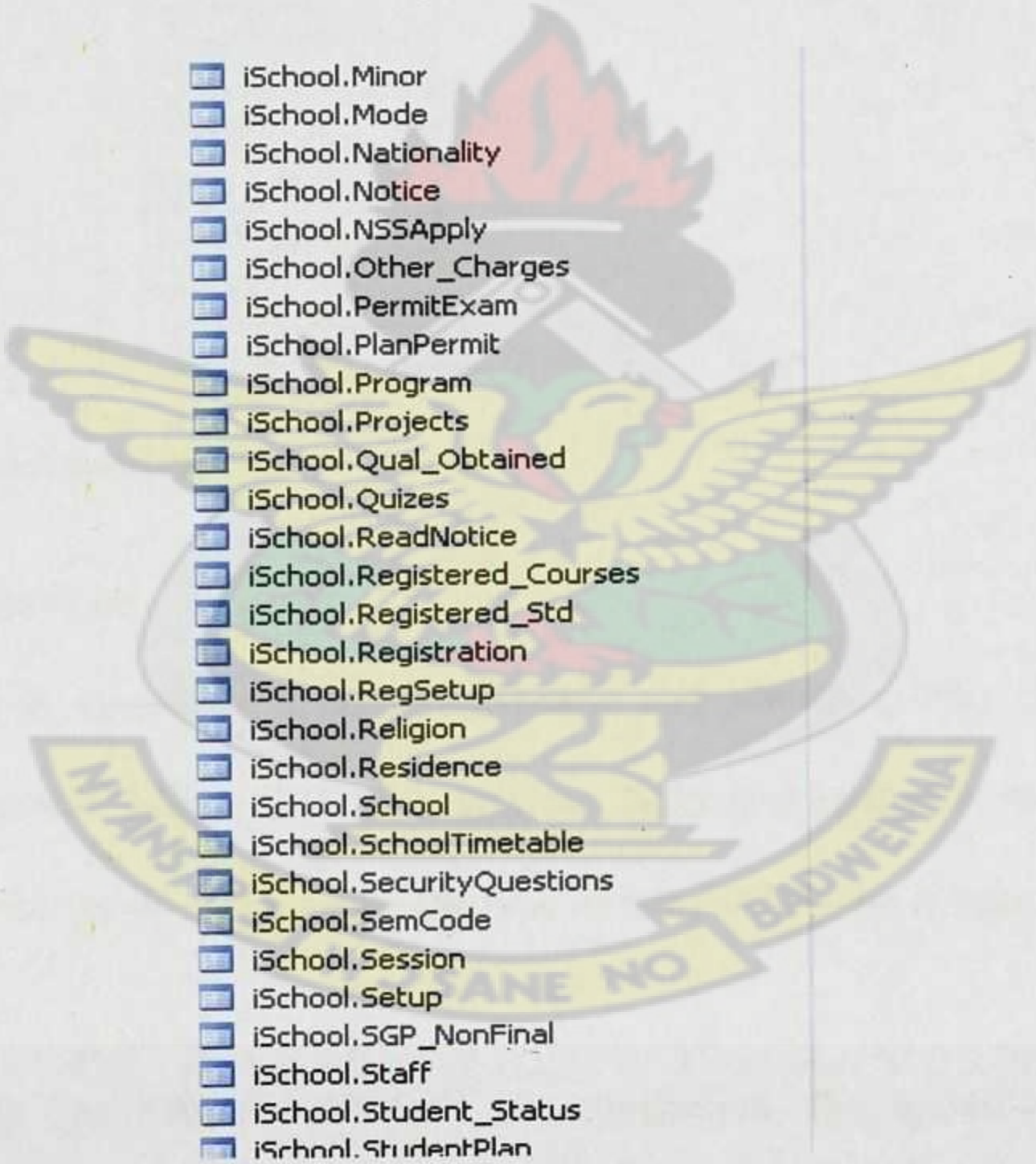


Figure 4.1a: Some tables in the database

Each of the fields in the table has different forms of attributes. The student table will be used in performing the analysis. Some of the tables within the database are shown in figure 4.1b (the schema). The schema shows the list of fields and primary keys within the table. Several tables as shown in figure 4.1a are obtained from the database.

Students	
	StudentID Title LastName FirstName OtherName Sex DOB POB P_Address C_Address Mature_Std NationalityID Passport_No Social_Sec_No NationalID_No Fax_No Tel_No Mobile_No Email Marital_Status ModelD Religion Denomination ConferencID AccountNo ProgramID MajorID MinorID DeptID Admit_Year HallAffiliate Status AcStatusID finAggregate Image ContentType CreateDate

Registered_Std	
PK	<u>StudentID</u>
PK	<u>SemCode</u>
	ID Reg_Date Total_Credit RegType StdYear StdSem

Registered_Courses	
PK	<u>StudentID</u>
PK	<u>SemCode</u>
PK	<u>CourseID</u>
	Grade G_Point SessionID ModelD selfReg timesTaken CreateDate

Figure 4.1b: Tables at the schema level

4.1.2 Cleansing process

Data cleansing is viewed as a process (Maletic and Marcus, 2000). The cleansing process in this research begins when the data has been imported into the application. Here, the user has the choice of selecting the type of error within the database he will like to work on.

When the data has been loaded into the application, the searching process is implemented and then the list of clean data and duplicated or missing data items are displayed in separate windows. The clean data is then exported to another file for maintenance.

4.2 Users of the proposed system

The new system proposed in thesis will be used by administrators, staff and any authorized person mandated with the credentials of the databases at the operational sources or the resident computers.

4.3 Benefits of the proposed system

Data is very important to business organizations, educational institutions, and health institutions and useful for reporting purposes. Business organizations are very much interested in their customer information, employee data and inventory on various assets, educational institutions will do everything possible to obtain the correct data about their students, staff, faculty and other workers. Patient record is very important in all health centers and hospitals across the globe.

The integrated cleansing model will have several benefits to organizations that implement it.

- **It will increase the accuracy and consistency of records:** data obtained from various departments will be accurate and consistent after the cleansing process before moving it into the data warehouse for analysis, data mining and reporting. For example, if you obtain accurate address of all your customers, it will improve the response rate which is likely to enhance or increase revenue. When duplicates or incomplete data is standardized in the database or warehouse, it will reduce most of the mails that do not get to their intended users.
- **Improves cost of company integrity:** the accurate and correct data obtained in the company's data warehouse enhances its image and credibility. Their stakeholders,

customers and employees will see them to be very reliable and increasing the customer base.

- **Enhance Analysis:** the data cleansing tool will generate correct information which will enhance the results for reporting and graphical analysis.
- **Easy-to-use:** this cleansing tool is easy to use since it comes with a flexible graphical user interface (GUI). It is very simple to use by people with or without the technical mindset.
- **Allows data import:** this cleansing tool can import files from Excel and Text and databases such as MSSQL, MySQL and Oracle. The proposed system can identify duplicates (such as names, addresses, emails, etc) and incomplete data (when some fields are missing).
- **Varied end users:** this data cleansing tool is suitable for businesses such as retail, marketing and banking; educational institutions such as universities, polytechnics, training colleges, and other schools. Hospitals can also benefit from this cleansing tool.
- **Time saving:** it is time saving since the system will use a little time to clean the duplicates as compared to hiring an IT personnel to walk through a huge database.

4.4 Data collection

The data to be used for the testing of the system in the proposed system is students' records from any university (but for the purposes of demonstration, Valley View University students' database will be used).

4.5 The proposed data cleansing method

There are several data cleansing methods that have been reviewed in the previous chapter. This algorithm though not perfect, could be adopted. Scripts on other types of error could be added to the knowledge base by other developers in order to obtain a more consistent and accurate data sent into the warehouse for reporting purposes within organizations and institutions.

4.6 Requirements Analysis

4.6.1 Functional Requirements

There are important aspects of the data cleansing system that must meet the user specifications. These are referred to as functional requirements. In other words, according to Bruade (2001) as cited by Reaves (2003), they refer to the functionality of the system, that is, the services the data cleansing tool will provide to the user. The functional requirements under this system include:

- Access the *iCleaner* system main interface
- The user chooses to import database or a file
- Choose type of error to remove (duplicated errors or missing data)
- Select to perform cleansing process
- The user exports the clean data to a storage location

4.6.2 Non-Functional Requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within (Bruade, 2001) as cited by (Reaves, 2003). Since the

data source will be collected from various databases, the following constraints must be taken into consideration.

The Integrated Cleaner (*iCleaner*) data cleansing system needs an existing Internet connection or Telephone line supports in order to connect to the relational database when it is not resident on the same machine as the cleansing tool. The *iCleaner* can connect to any MySQL database when the *MySQL Connector Net* is installed on the resident machine in order to obtain access to servers with MySQL databases. It is also very important to check the port number entered at the *Advance Options* button on the *MySQL Data dialog box* (**figure 5.6**) if it is the same as the one entered there (**that is 3306**). Change this port number to satisfy the port number of the connecting server.

On the other hand, to connect to servers with Oracle databases, ensure that the Oracle Data Provider for .NET platforms (ODP.NET) is installed on the connecting server in order to access the various tables in the Oracle database. Moreover, it is required that the port number entered at the *Advance Options* (**that is 1521**) on the *Connect Oracle dialog box* (**figure 4.1**) after clicking the Oracle logo on the main page be verified with that of the server you are connecting to.

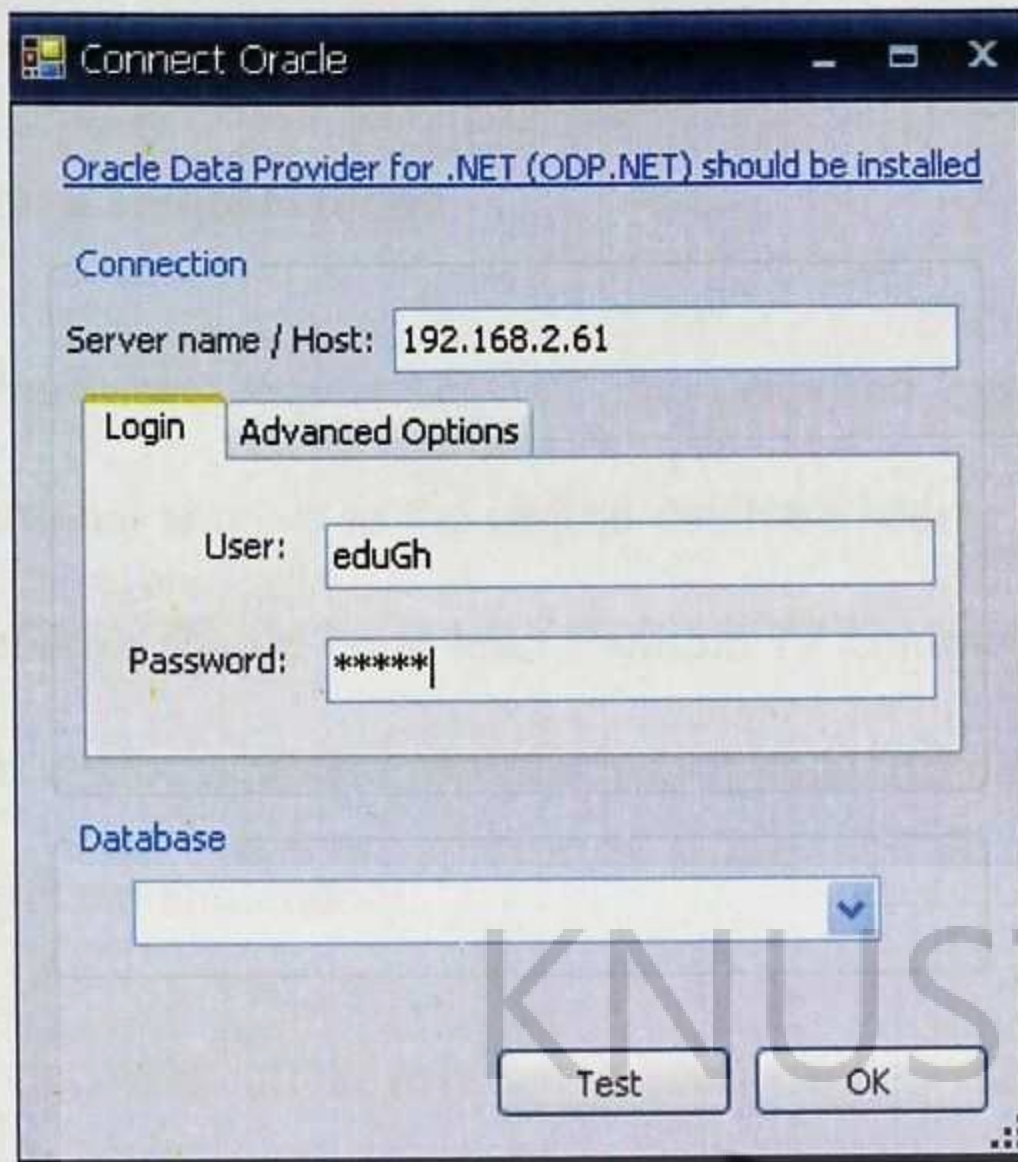


Figure 4.1c: Oracle interface for connecting to data source

To successfully connect to these relational databases (MSSQL, MySQL, and Oracle), verify to ensure that the servers allow remote connection. This will open the connections for the *iCleaner* to connect to the database as illustrated in figure 4.1c.

Another important verification that needs to be done is to add the *iCleaner* software to the Windows firewall allowed programs (better option for security reasons). This will allow the software to communicate through the firewall for protection of data cleaned before moving it to the warehouse.

On the contrary, you can turn off the windows firewall (not better option though in terms of security) to allow the system to successfully connect to the database.

4.6.1.1 Hardware Requirement

The major hardware requirement of the proposed system is a network system connecting the operational sources to the central database source and the cleansing tool. The computers on the network should be at least Pentium IV computers with minimum speed of 1GHz (Dual Core processor or above recommended) and a minimum of 128 MB, 256MB or 512MB of RAM (1GB recommended).

The system can work effectively when connected to the network. This will enable users to connect to servers or computer systems with the databases within their environs. The network topology or type may not necessarily have effect on the system's implementation.

4.6.1.2 Software Requirement

The main software requirement is to have a 32bit or 64 bit Operating System as the system type. The system is designed to be operating system independent in order for it to be implemented on every system in any business organization irrespective of the platform of which their systems are built.

The supported operating systems are as follows: Microsoft Windows XP with Service Pack 2 and Pack 3, MS Windows 7 with Service Pack 1, Windows Server 2003 Service Pack 2, Windows Server 2008, MS Windows Server 2008 R2 SP1, MS Windows Vista Service Pack 1 and MS Windows Server 2003 SP2.

4.6.3 Input Requirements

There are several relational databases such as MySQL, Microsoft SQL Server, Oracle and Access, which have a much more logical structure in the way they store data. For the purposes of the limitation, this research will consider MSSQL and MySQL relational databases and the files to be considered will be text files (.txt and .csv extensions) and Excel (.xls, .xlsx, xlsx, xlsb extensions) as the input to the cleansing system. All other inputs which are not specified could be considered as derived input which may be fed into the system when the need arises.

4.6.4 Processing Requirements

The major processing requirement of this system is the data obtained from the various databases *iCleaner* will be connecting to access its tables. This is to establish whether some of the records have duplications and whether some fields are missing or not. Upon retrievals, the system will clean all duplicates and merge the incomplete data fields to make them complete and useful. The cleansed data will be moved into the data warehouse or a known storage location for usage.

4.7 Design of Proposed System

In order to deal with the drawbacks mentioned above, a proposed conceptual framework to keep the cleansing lineage as well as involving users to validate the cleaned data in DC process before sending it to the data warehouse for quality analysis to improve reporting and analysis of data obtained from the cleansing tool is discussed.

The conceptual framework (*see figure 2.9*), illustrates how data is extracted, cleaned and transformed (applying the proposed algorithm) and loaded from the source computers

into the central database. This data is loaded into the data cleansing tool (*iCleaner*). The major steps that take place in the *iCleaner* are as follows:

- First of all, the data is loaded into a temporary location in order to keep track of databases and the files that have been moved in order to create a cache copy.
- The type of error is now viewed and selected.
- Afterwards, if the error type checked is “duplication,” then the cleansing system will scan through to detect such duplicates by adding all the rows to obtain a unique value of key.
- When the error type checked is “missing,” and then the rows are updated.
- The cleaned data is validated by the user before uploading the data into any location on a secondary storage device or the warehouse, otherwise the data is sent back for further cleansing.

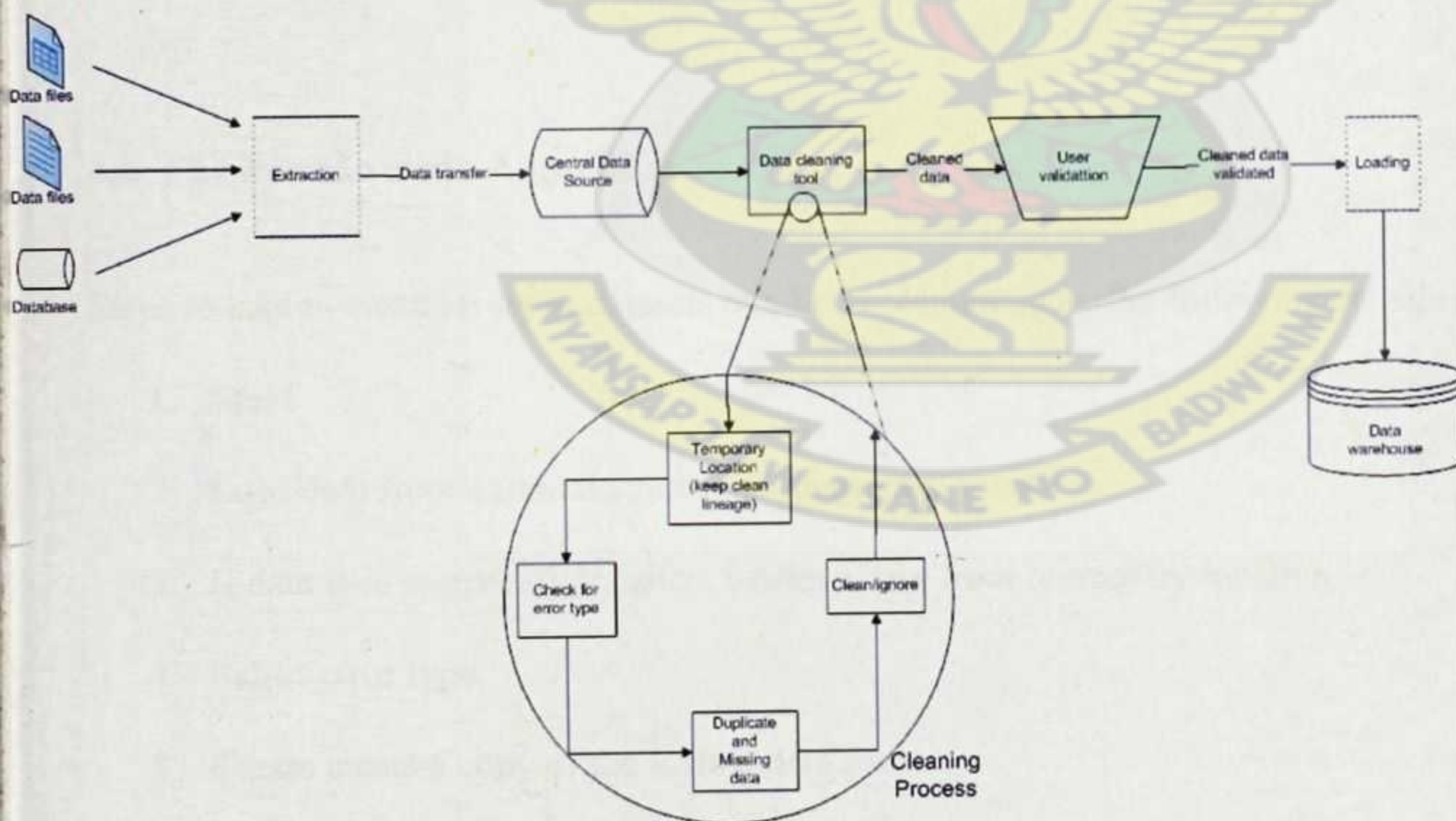


Figure 4.2: diagram for cleansing data records

The above figure 4.2 describes the whole data cleansing process that the current application follows to perform its task. It describes the extraction and loading processes involved in the cleansing data and loading into warehouse. When the data is extracted, it is loaded into a central database before the cleansing tool accesses the data.

4.7.1 Processing Modeling

The required data for processing in this research is data obtained from MSSQL server from *a census database* (the database was obtained from Valley View University (VUU) final year students' project). Specifically, VUU students' records are obtained every semester when both new and old students register for the semester's program.

4.8 The Pseudo-code Algorithm

Steps to take to clean errors in datasets can be implemented by the following pseudo code

1. Start
2. Load data from external sources to temporary location
3. If data is in temporary location, retrieve data from temporary location
4. Select error type
5. Create a cache copy of the loaded data set
6. If error type is duplicated, then go to step (8)
7. Else go to step (9)
8. Determine error in the loaded cache copy of the data set
 - a. Begin loop

While n (rows of loaded data) is not the last row

- i. Add all columns to obtain a single key
 - ii. store key in dictionary
 - iii. implement binary search to compare generated key with the stored key
for the next row
 - iv. if generated key exists in the dictionary, remove the row from cache
copy data and store in a duplicate table (i.e. this key will not be stored
in the dictionary)
 - v. else store unique key in dictionary and continue
 - vi. end if
 - vii. end else
 - viii. End of loop
 - b. display the cleaned data obtained from the operation in (a) above
 - c. display the duplicated rows obtained in (a.iv) above
 - d. Exported data can be moved to a known location. End if
9. Search table for missing data by checking if the key columns in loaded cache copy
data is empty
- a. Checking: if key column is empty, then data is missing in the table.
 - b. Display the missing data records in a missing data table
 - c. Move updated table in (c) above into a location (i.e. save new table).
10. End if
11. End else
12. Stop

4.8.1 Data from the Student table



Figure 4.3: A table of records with labels

In order to work on the data, a portion of the student table within the database is extracted as shown in figure 4.3 above. The “key column” (KC) is any column required for checking a field for data item. “Sum of rows” (SOR) takes into account all the various “key columns” concatenated to form a single string for the purposes of searching the database for erroneous data. “Missing field” is any “key column” that has no data set (i.e. either blank or NULL).

The columns within this table are StudentID, title, lastName, firstName, OtherName, Sex, DOB, POB, P_Address, C_Address and Mature_std, etc. There are several columns within the table. Depending on the item the user may be interested in searching; priority could be placed on any of them as “key column.”

4.8.2 The Algorithm: Method to Detect Errors

This section presents the methods used in the identification of errors in this research. The method of “*Addition of Rows to obtain a single value*” based on key columns to detect errors (identification and cleansing) which are duplicated and the method of Identification of “*Null values and Blanks*” to detect missing data (that is, a column that is empty) in key columns. Each of the two methods presented here searches through the data set by implementing the binary search methods. This search method is used in order to obtain optimal performance and response time.

4.8.2.1 Method of Addition (or Concatenation) of rows

In the algorithm, in order to detect duplicates in any data set, all the key columns are concatenated (or added) together and converted into string. It is imperative to note that there is a consistency among columns within the various tables in the database. In the method of

detecting duplicates, each row that has concatenated or added generates a *single value (this is a string)*.

Table - iSchool.Students		Summary						
	StudentID	Title	LastName	FirstName	OtherName	Sex	DOB	POB
▶	100/00006	Miss	Adjetey	Sarah	Adjorkor	Female	6/30/1981 12:0...	Tema
	100/00007	Miss	Adu	Ivy	NULL	Female	2/27/1969 12:0...	Tema
	100/00008	Miss	Afrifa	Linda	NULL	Female	7/16/1978 12:0...	Kumasi
	100/00009	Miss	Agbeko	Mary	NULL	Female	11/22/1977 12:...	Accra
	100/00010	Mr.	Agbo	Joshua	Elikplim	Male	7/1/1941 12:00:...	Anloga
	100/00011	Miss	Agyare	Yvonne	Adjepong	Female	6/5/1981 12:00:...	Koforidua
	100/00012	Mr.	Agyei	Paa Kwesi	P.	Male	5/31/1981 12:0...	Accra
	100/00013	Mr.	Agyei-Baah	John	NULL	Male	1/1/1980 12:00:...	Akim Swedru
	100/00014	Mr.	Agyemang	Augustine	NULL	Male	12/1/1980 12:0...	Wenchi
	100/00015	Mr.	Agyemang	Jacob	NULL	Male	7/22/1976 12:0...	Sekyeduwase
	100/00016	Miss	Ainoo	Patience	NULL	Female	2/23/1965 12:0...	Tarkwa
	100/00017	Mr.	Akafia	Charles	NULL	Male	7/7/1962 12:00:...	Accra
	100/00018	Mr.	Akanluke	James	NULL	Male	2/20/1970 12:0...	Sefwi Wiawso
	100/00019	Mr.	Akawobsa	Dennis	Elisha	Male	6/9/1980 12:00:...	Sandema

Figure 4.4: table of records

From the figure 4.4 above, concatenating the first row (i.e. adding each value in the column; here *value* represents any data type found within the column), the following string will be generated as shown in the table 4.1 below.

Table 4.1: Concatenated rows

Row number	Single value (string) obtained
1	100/00006MissAdjeteySarahAdjorkorFemale6/30/198112:0...Tema
2	100/00007MissAduIvyNULLFemale2/27/196912:0...Tema
3	100/00008MissAfrifaLindaNULLFemale7/16/197812:0...Kumasi
4	100/00009MissAgbekoMaryNULLFemale11/22/197712:...Accra
5	100/000010MrAgboJoshuaElikplimMale7/1/194112:00...Angola

From the above table 4.1, after the string has been generated, each of these strings is stored in a single value dictionary containing single values/strings, and they are compared to find for the same strings (i.e. duplicates). During the comparison of all the strings, each single value or string that does not recur in the list is considered as unique and it is stored in a hash table (unique key dictionary).

For the purposes of this research, a unique key is defined as the single value or string that has no recurrent (repeated) value in the single key dictionary. When the first string is compared to the other string and a match is found then that string value is considered duplicated and this string value is stored in non-unique key dictionary.

It should be noted that every key acts as a pointer to the record within the rows fetched from the main table. This means that the actual data is presented for reporting purposes and not data stored in the key. Further prove on how the algorithm detects duplicates is discussed in section 4.8.2.1 above.

4.8.2.2 Identification by key columns

In searching for missing data, the research proposes a technique where in each row a column(s) is checked. In this case, two pointers are distinct – blank or null column; this is because data is presented in columns in tables. In this regard, the missing data is identified in a row when one of the column data is not available or present.

Though this has been established, the algorithm only identifies some particular columns of relevance (since not every column in a table in a database is required); for example, in figure xx above the column with “*OtherName*” is not always required since it is not all students who have other names as shown in figure 4.5. Considering columns such as *FirstName*, *LastName* (as shown in figure 4.5), *DOB* and *POB* (as shown in figure 4.4), all

students possess these attributes; hence, any or all such fields that are required fields are referred to as *key columns* in this research because they are required fields in the database.

LastName	FirstName	OtherName
Adjekey	Sarah	Adjorkor
Adu	Ivy	NULL
Afrifa	Linda	NULL
Agbeko	Mary	NULL
Agbo	Joshua	Elikplim
Agyare	Yvonne	Adjepong
Agyei	Paa Kwesi	P.
Agyei-Baah	John	NULL

Figure 4.5: column identification

4.8.3 Applying Data Cleansing

The goal of this section is to demonstrate and prove that these methods proposed can be successfully used to identify duplicates and missing data. The implementations are designed and tested on both smaller data sets and larger data sets. The technique that will serve as advantage is the use of dictionaries to store the various values for the searching processes and also for the detecting of duplicates and missing data. To illustrate this we define the following variables.

Definition of variables:

Let error instance be denoted E

Let the data set be represented as D , where the last data row is D_n

Let each row instance be denoted as R , where $R = \{d_1, d_2, d_3 \dots d_i, \dots d_n\}$

Where d_i is the i^{th} data value

Let each record be denoted as F , where $F = \{f_1, f_2, f_3 \dots f_i, \dots, f_n\}$

where f_i is the i^{th} field and $f_i \subseteq D$

Let each table item be represented as T , where

$$T = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_j \end{bmatrix}$$

and the value of $d_i \subseteq T$

Summation of columns, C where C is *dup* or *uq*

Representation of error *dup* for duplicated rows, *uq* for unique rows and *mis* for missing data

Let user validation be represented as UV_c

Let S_d represents all duplicated rows

Let S_u represents unique rows

To determine that a row is unique or has duplicated values, calculate for S_u and S_d , where S_u and S_d are the variables to store the differences between the two columns, C_k and C_{k+1} .

In this work, two rows are equal (i.e. when their single value or string value obtained are the same) when the system performs the concatenation operation (i.e. the addition of the key columns within that particular row in the table) and turns out to be the same.

For example, using the figure 4.6 below and using the method of addition of rows explained in section 4.8.1.1, the string values obtained in the table 4.2 below show that the row numbers (7 and 12) have the same string value (i.e. 100/00018Mr.AkanlukeJamesNULLMale2/20/197012:0...Wenchi), hence the two rows can be said to be equal. This situation results in duplicates. It is an essential step to detect the approximately the number of duplicated records for cleansing. By detecting these duplicated records, we can determine whether the two records are the same by comparing their values as shown in figure 4.6 below.

Table - iSchool.Students		Summary						
	StudentID	Title	LastName	FirstName	OtherName	Sex	DOB	POB
	100/00012	Mr.	Agyei	Paa Kwesi	P.	Male	5/31/1981 12:0...	Accra
	100/00013	Mr.	Agyei-Baah	John	NULL	Male	1/1/1980 12:00:...	Akim Swedru
	100/00014	Mr.	Agyemang	Augustine	NULL	Male	12/1/1980 12:0...	Wenchi
	100/00015	Mr.	Agyemang	Jacob	NULL	Male	7/22/1976 12:0...	Sekyeduwase
	100/00016	Miss	Ainoo	Patience	NULL	Female	2/23/1965 12:0...	Tarkwa
	100/00017	Mr.	Akafia	Charles	NULL	Male	7/7/1962 12:00:...	Accra
	100/00018	Mr.	Akanluke	James	NULL	Male	2/20/1970 12:0...	Sefwi Wiawso
	100/00019	Mr.	Akawobsa	Dennis	Elisha	Male	6/9/1980 12:00:...	Sandema
	100/00020	Mr.	Amaniampong	Kofi	NULL	Male	12/19/1980 12:...	Accra
	100/00021	Mr.	Amankwah	Daniel	K. Fordwor	Male	10/11/1980 12:...	Accra
	100/00022	Mr.	Quaye	Nicholas	Amarteyfio	Male	1/2/1970 12:00:...	Takoradi
	100/00018	Mr.	Akanluke	James	NULL	Male	2/20/1970 12:0...	Sefwi Wiawso
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 4.6: Comparing two fields

Table 4.2: Equality among strings

Row number	String values
1	100/00012Mr.AgyeiPaaKwesiP.Male5/31/198112:0...Accra
2	100/00013Mr.Agyei-BaahJohnNULLMale1/1/198012:00:...AkimSwedru
3	100/00014Mr.AgyemangAugustineNULLMale12/1/198012:0...Wenchi
....	
7	100/00018Mr.AkanlukeJamesNULLMale2/20/197012:0...SefwiWiawso
...	
11	100/00022Mr.QuayeNicholasAmarteyfioMale1/2/197012:00...Takoradi
12	100/00018Mr.AkanlukeJamesNULLMale2/20/197012:0...SefwiWiawso

The following deductions hold:

Assuming that $k = 0$, and $i = 0$; where k and i represents the starting row in a given table.

This means that

$$S_d = C_k - C_{k+1} = 0$$

This implies with respect to duplicates (*dup*) the resulting value after the computation is equal to zero.

$$dup = \sum_{k=1}^n (C_k - C_{k+1}) = 0 ; \text{ for the detection of duplicates}$$

On the other hand, a second deduction with respect to unique (*uq*) value could be computed using the expression below. In this regard, when the value is less than 0, then we can infer that that row is unique.

$$S_u = C_i - C_{i+1} < 0$$

$$uq = \sum_{i=1}^n (C_i - C_{i+1}) < 0; \text{ for the detection of unique keys}$$

4.8.4 Procedure for Detection Errors

In order to detect the number of missing fields in a particular data set, the following procedure was experimented. A key column in a row is checked for null values or spaces based on the procedure below. The main approach is to select the key columns to search for the missing data. Some of the parameters include d_a and F_i , where d_a and F_i represents row and field respectively.

Once these key columns are compared and they are not equal to null, then there exist no empty field. On the other hand, empty fields exist.

For a = 0 to D_n

{ For i = 1 to n

{ If $d_a F_i = \text{NULL}$

{ return true

println (" Missing data found in row")

//end of if statement checking empty table

else

{ println ("No missing data found in row") }

}

}//end of checking for missing field

To determine that some rows have duplicates and others have unique values, the basic operation conducted is described below. The approach used is clearly defined with variables that represent the beginning line for checking for duplicates. Here, all the terms in the expressions below are defined. For a row to be identified as unique, the summation of the key columns within that row should not be equal to the summation of the key columns in the n number of rows.

On the other hand, a duplicate is any row which key column(s) summed up has its equivalence in the n number of rows.

$k = 0$; $key = ""$; $m = 0$; $i = 0$;

- where **key** is the unique identifier
- where **m** is the variable to hold the summation of all fields in a specified row
- where **i** is the value of each field
- where R_n is the last row within the data set

This implies that if the **key** stored in the dictionary contains the value of **m**, then a duplicate has been found. This also implies that the key is added to the dictionary.

while ($k < R_n$)

{

$$m = R_k F_i + R_k F_{i+1} + \dots + R_k F_n$$

The representation of key column addition is shown below.

$$\sum_{k=0}^n \sum_{i=1}^n R_k F_{i-1}$$

```
if ( key.contains (m))
```

```
{      println ("Yes, duplicate found \n" + "Clean error instance  $E$ ");      }
```

```
else
```

```
{      println ("No, duplicate not found \n" + ignore error instance  $E$ ");
```

```
      key = key + m;      //counter
```

```
    }//end of else statement
```

```
k= k+ 1;
```

```
//end of loop
```

4.8.5 Code Snippet (Application of the algorithm): Duplicate data

```
//create a list of column names by adding them to a
```

```
//List<string> generic collection to feed into the method. For Example,
```

```
//List<string> keyColumns = new List<string>();
```

```
//keyColumns.Add("ColumnA");
```

```
//keyColumns.Add("ColumnB");
```

```
public DataTable RemoveDuplicates(DataTable table, List<string> keyColumns)
```



```
{///Creation of dictionary => uniquenessDictionary
```

```
//NB: the length of the dictionary is equal to total number of rows in loaded //data
```

```
Dictionary<string, string> uniquenessDict = new Dictionary<string,  
string>(table.Rows.Count);
```

```
StringBuilder sb = null; //create string builder for generating the key
```

```
int rowIndex = 0; //first row index of the loaded data //data table to store //duplicate  
data have same structure
```

```
DataTable dt = table.Clone(); DataRow row;
```

```
//cache copies of the rows in the loaded data set are determined here
```

```
DataRowCollection rows = table.Rows;
```

```
//beginning of loop and using the binary search mechanism
```

```
while (rowIndex <= rows.Count - 1)
```

```
{
```

```
row = rows[rowIndex]; //store each row
```

```
sb = new StringBuilder(); //the key builder
```

```
foreach (string colname in keyColumns)
```

```
{ try {
```

```
sb.Append(((string)row[colname])); //creating a key for every cache //copy of data
```

```
} catch { }
```



```

//compare key in sb object to keys already stored in dictionary

//binary search dictionary for generated key

if (uniquenessDict.ContainsKey(sb.ToString()))//comparing unique keys

//generated to detect duplicates

{

try { dt.Rows.Add(row.ItemArray); }//insert to duplicate table

catch { }

rows.Remove(row);//removing or deleting the duplicated row from //cache copy

}

else

{

uniquenessDict.Add(sb.ToString(), string.Empty);//add to dictionary

rowIndex++;//increment count to next row }

}

//end of loop

project.MissingData = dt;//storage location for all duplicates and missing //data.

return table;//display the table containing the duplicated data

}}}

```


4.8.6 Code Snippet (Application of the algorithm): Missing data

```
/// <param name="table">The Table to be used in cleansing</param>
```

```
/// <param name="keyColumns">The Columns that are to be searched</param>
```

```
/// <returns>DataTable</returns>
```

```
public DataTable FindMissing(DataTable table, List<string> keyColumns)//search table  
and compare columns
```

```
{
```

```
int rowIndex = 0;
```

```
DataTable dt = table.Clone();
```

```
DataRow row;
```

```
DataRowCollection rows = table.Rows;
```

```
while (rowIndex <= rows.Count - 1)//loop begins
```

```
{
```

```
row = rows[rowIndex];
```

```
foreach (string colname in keyColumns)
```

```
{
```

```
try
```



```

{      //missing row identification, when field name is null or //contains
      space

      if (DBNull.Value == row[colname] || (string)row[colname] == "")

      {

          try { dt.Rows.Add(row.ItemArray); }

          //add all key columns //containing missing data

          catch { }

          rows.Remove(row); //remove missing data

      }

      catch { }

      rowIndex++;

  }

  project.MissingData = dt;

  return table; }}} framework

```

4.8.7 Operation before exporting

Having completed the detection of missing data records, the detected missing data records should be processed (if required). For a group of missing data records retrieved, two methods are applied: regard one record true in the missing records in order to delete or edit the records in the database. In this situation, the following measures can be taken:

4.8.7.1 Editing records

Editing records can apply to the removed missing data moved out of the main table in the database. When this is performed, the record could then be added to the clean data before exporting the data.

4.8.7.2 Adding records

The missing data record once removed from the main table could be added back if the user is not too sure of its existence and perform deletion in a later date. To add to the main table again, the user should right click edge of field and click on "Add Record."

4.9 Determining the Algorithm Complexity

From the literature review conducted, it turned out that the authors did not state the various indicators/parameters they measured the complexity of their algorithms with. In this research, the algorithm developed is measured based on its performance, running time, hardware, space and evolution.

4.9.1 Performance and complexity of algorithm

The process of detecting and eliminating approximately duplicate records and missing data from operational sources into a data warehouse by data cleansing techniques is a very complicated one. To identify and remove duplicate records and missing data of different sources, the **column addition using binary search tree to recognize duplicates and pair-wise strategy for comparing various columns** is proposed.

The ~~system is expected to~~ generate records from the appropriate tables in the databases 100% of the time. The speed of the proposed system will depend on the hardware requirement rather than the systems characteristics.

4.9.2 The running time of the algorithm (Time Complexity)

In this thesis the algorithm proposed will be used to solve a particular task which makes use of a set of basic operations in the determination of the unique key for detecting duplicates and identifying rows with missing fields. The work done by the algorithm is calculated based on the estimation of the major operations. Initialization of some of the variables in the algorithm was not considered since the running time of such primitive operations is usually constant (i.e. $O(1)$).

To able to understand how the performance of this algorithm reacts to changes in input size, this research; basic goal is to provide an insight to its readers by determining the running time. The Big-O notation is a way of measuring the order of magnitude of a mathematical expression and the expression $O(n)$ means on the Order of n . The above code snippet contains a lot of basic operations which will be repeated until all the rows have been searched. The operations it contains are shown in table 4.3 and table 4.4 below. In the tables 4.3 and 4.4, the various statements and their interpretations are listed and used for the determination of the running time.

Table 4.3: Running Time for detecting duplicates

Statement	Interpretation
int rowIndex = 0;	This statement executes only once. It is a primitive operation. The running time is $O(1)$
DataRow row =0;	This statement executes only once. It is a primitive operation. The running time is $O(1)$
DataRowCollection rows = table.Rows	This statement executes only once. It is a primitive operation. The running time is $O(1)$
rowIndex < rows.count-1;	This operation will execute n times
row = rows[rowIndex];	This operation will execute n times
Sb = new StringBuilder ()	This statement executes only once. It is a primitive operation. The running time is $O(1)$
for (string colname in key columns)	This operation will execute $n*n$ times
sb.Append((string)row[columns]))	This operation will execute n times
if(uniqueDict.containskey (sb.ToString()))	This operation will execute n times
rowIndex++	This operation will execute n times
project.MissingData = dt	This statement executes only once. It is a primitive operation. The running time is $O(1)$

To determine the running time from the number of operations specified in the table above we derive the following:

$$\begin{aligned}
 T_1(n) &= 1 + 1 + 1 + n + n + 1 + n * n + n + n + 1 \\
 &= 5 + 4n + n^2 \\
 &= O(5 + 4n + n^2)
 \end{aligned}$$

Since n approaches infinity the value 5 becomes insignificant (i.e. 5 nearing zero (0))

$$= O(4n + n^2) = O(n^2)$$

From the above calculation, we can simply say that the algorithm exhibits a quadratic time or order n^2 .

Table 4.4: Running time for detecting missing data

Statement	Interpretation
<code>int rowIndex = 0;</code>	This statement executes only once. It is a primitive operation. The running time is $O(1)$
<code>DataTable dt = table.Clone();</code>	This statement executes only once. It is a primitive operation. The running time is $O(1)$
<code>DataRowCollection rows = table.Rows;</code>	This statement executes only once. It is a primitive operation. The running time is $O(1)$
<code>rowIndex <= rows.Count - 1</code>	This operation will execute n times
<code>row = rows[rowIndex];</code>	This operation will execute n times
<code>foreach (string colname in keyColumns)</code>	This operation will execute $n*n$ times
<code>if(DBNull.Value==row[colname] (string)row[colname] == "")</code>	This operation will execute N times
<code>project.MissingData = dt;</code>	This statement executes only once. It is a primitive operation. The running time is $O(1)$
<code>rowIndex++;</code>	This operation will execute n times

$$T_2(n) = 1 + 1 + 1 + n + n + n * n + 1 + n$$
$$= 4 + 3n + n^2$$
$$= O(4 + 3n + n^2)$$

Since n approaches infinity the value 4 becomes insignificant (i.e. 4 nearing zero (0))

$$= O(3n + n^2)$$

Since n^2 approaches infinity the value $3n$ becomes insignificant nearing zero (0).

$$= O(n^2)$$

The result of $T_1(n)$ and $T_2(n)$ shows that the running time of the algorithm is a quadratic function. This can be further explained that the running time depends much on the input size of the data set that will be loaded for the cleansing tool to perform its cleansing process. The size of the input from the deduction above will be n^2 depending on the system

requirements but will always be quadratic in n . The algorithm's running time greatly depends on the size of the data loaded into the cache copy.

4.9.3 Hardware Complexity due to the proposed framework

The proposed system can run with the minimum hardware requirement but comparatively performs better on hardware with higher specifications. Though the proposed framework seems a little complex, it does not require any complex hardware to run or execute efficiently.

4.9.4 Space Complexity

Space complexity deals with how much memory the algorithm uses when in operation. From the time complexity analysis, it could be inferred that the better the time complexity of an algorithm is, the faster the algorithm will carry out its work when in full operation.

The algorithm's *space complexity* is essentially the number of memory cells which the algorithm needs. After running the system several times with different input sizes, it was realized that the system uses a reasonable portion of memory. When the input size is large it still uses a reasonable amount of memory and processing capacity for the cleansing operation to complete.

4.9.5 System Evolution

In the future, this system will be easily updated to allow for other errors (such as incomplete data and wrong formatted data) and cleaned when scripts of such errors are clearly defined and added to its knowledge base.

4.9.6 Modeling Flowchart representation for the cleansing system

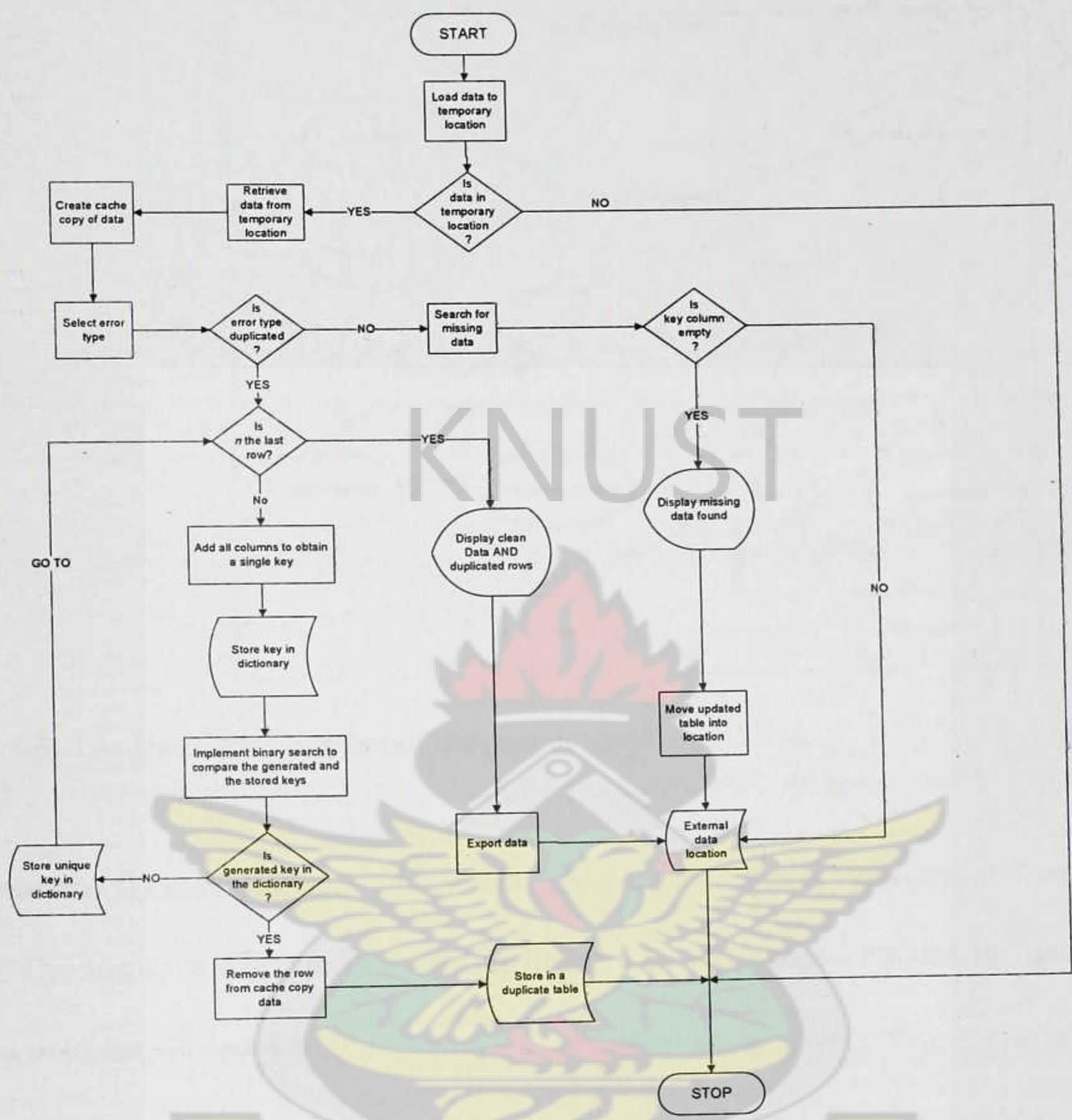


Figure 4.7: The Flowchart

The figure 4.7 represents the pseudo-code diagrammatically. The system flowchart depicts the flow of operations in the data cleansing model. The diagram starts by accepting the type of error and ends with the user validation and then terminates. The system flowchart in this thesis serves as the roadmap for the data cleansing tool because it is showing how all the major components interact and fit together.

4.9.7 Modelling Use Case diagram for the integrated cleansing system

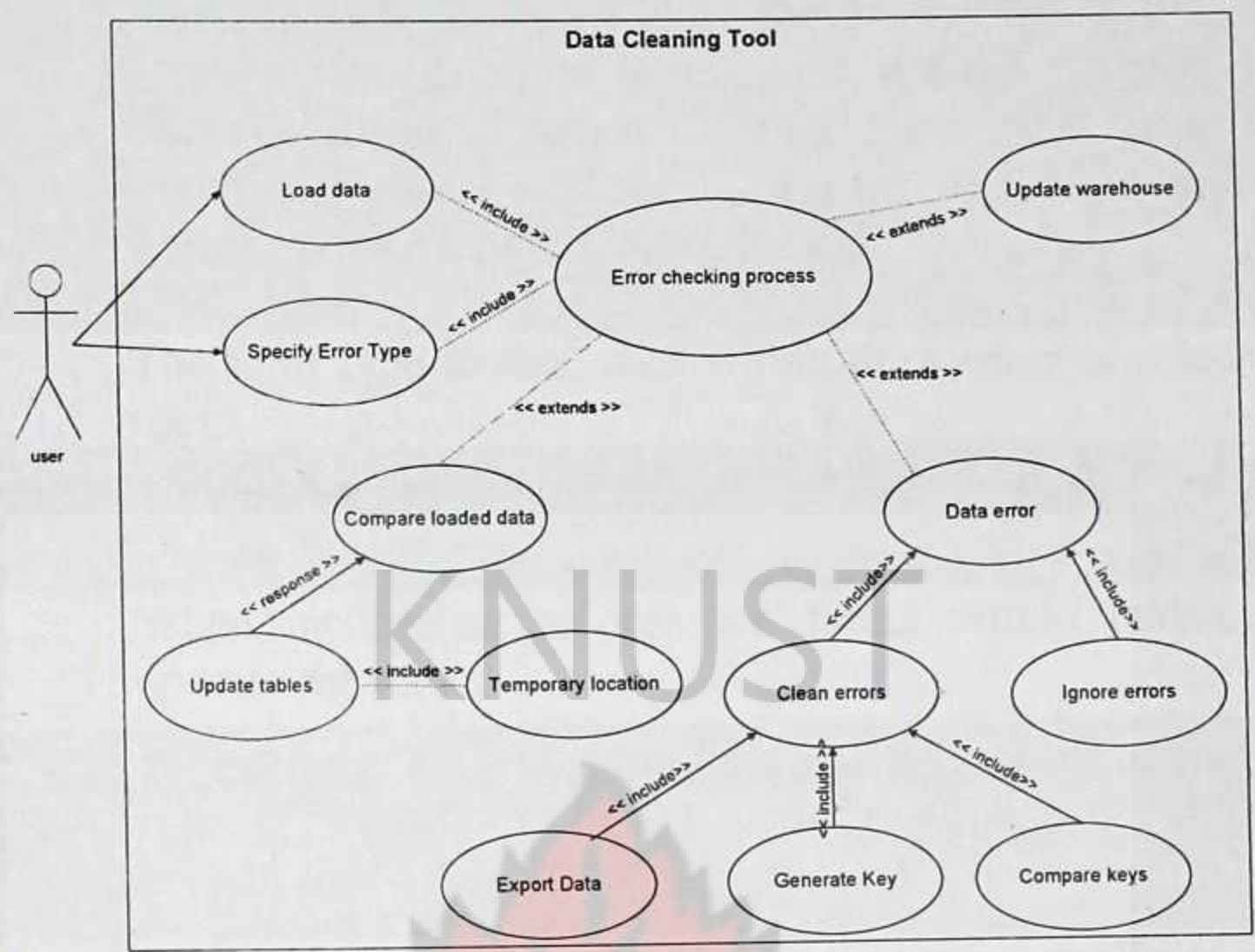


Figure 4.8: The Use Case Diagram of the pseudo code

There are *two (2)* main actors in this system. They are the System Administrator and the user (staff). The use case diagram is shown in figure 4.8. The various entities in figure 4.8 are defined in tables 4.5 and 4.6.

Definition of the actor

- **User:** Anyone person assigned to perform the cleansing task.

Table 4.5: Definition of the Use Case

Use Case	Definition
Load data	The user or staff can load data either from external (operational sources) or from the resident computer. This data load can be either a text, excel or a relational database. The system will display all the databases within the loaded data from the specified location and display all the tables. Locations where files are stored will also be displayed.
Specify error type	The error type to deal with within the system is selected by the user.
Compare loaded data	The loaded data is compared field by field until all the fields have been checked.
Update table	When incomplete records are found within tables, the user updates the table.
Temporary location	When the data is extracted from the operational sources, a cache copy of it is stored in a temporary location for the cleansing process to act upon it.
Data Error	Two major errors are checked by the user – duplicate or missing data.

Table 4.6: Definition of the Use Case Continued

Use Case	Definition
Clean errors	When the cleansing system detects errors (duplicates) in the selected tables it will display them for an appropriate action to be taken before the data is loaded into the data warehouse. This can only be done when the data has been populated and the duplication algorithm has been duly performed.
Generate key	A unique key obtained as a result of the addition of rows within a field.
Compare keys	When the unique key is generated, it is compared to other ones already stored in a dictionary for uniqueness.
Export data	Cleaned data is exported by saving it to a known location in .txt or .csv format for easy importation.
Ignore errors	This action is performed by the user of the system if to his knowledge the data needs re-cleansing and then reverses the action to the cleansing system.
Validate action	The user of the system can validate the clean action before saving it into a known location.

4.9.8 System model diagram for the integrated cleansing system

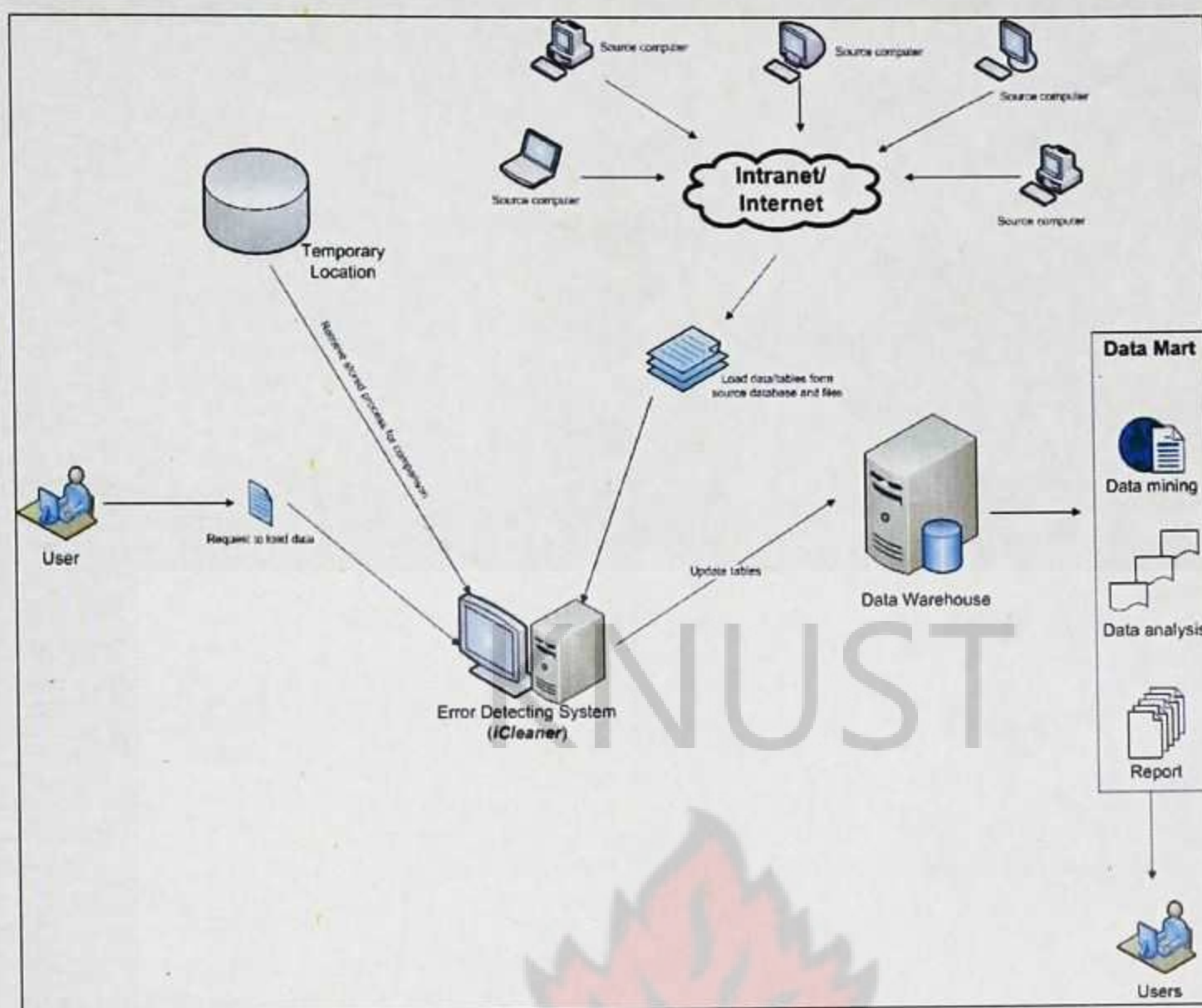


Figure 4.9: The System Model

Figure 4.9 shows the basic interaction of the proposed system when the algorithm is duly implemented systematically. Data extraction is from variety of sources and after cleansing loading takes place. This model is useful since it describes the interaction among the components of the system.

4.9.9 Data Flow diagram (Level 1) for the integrated cleansing system

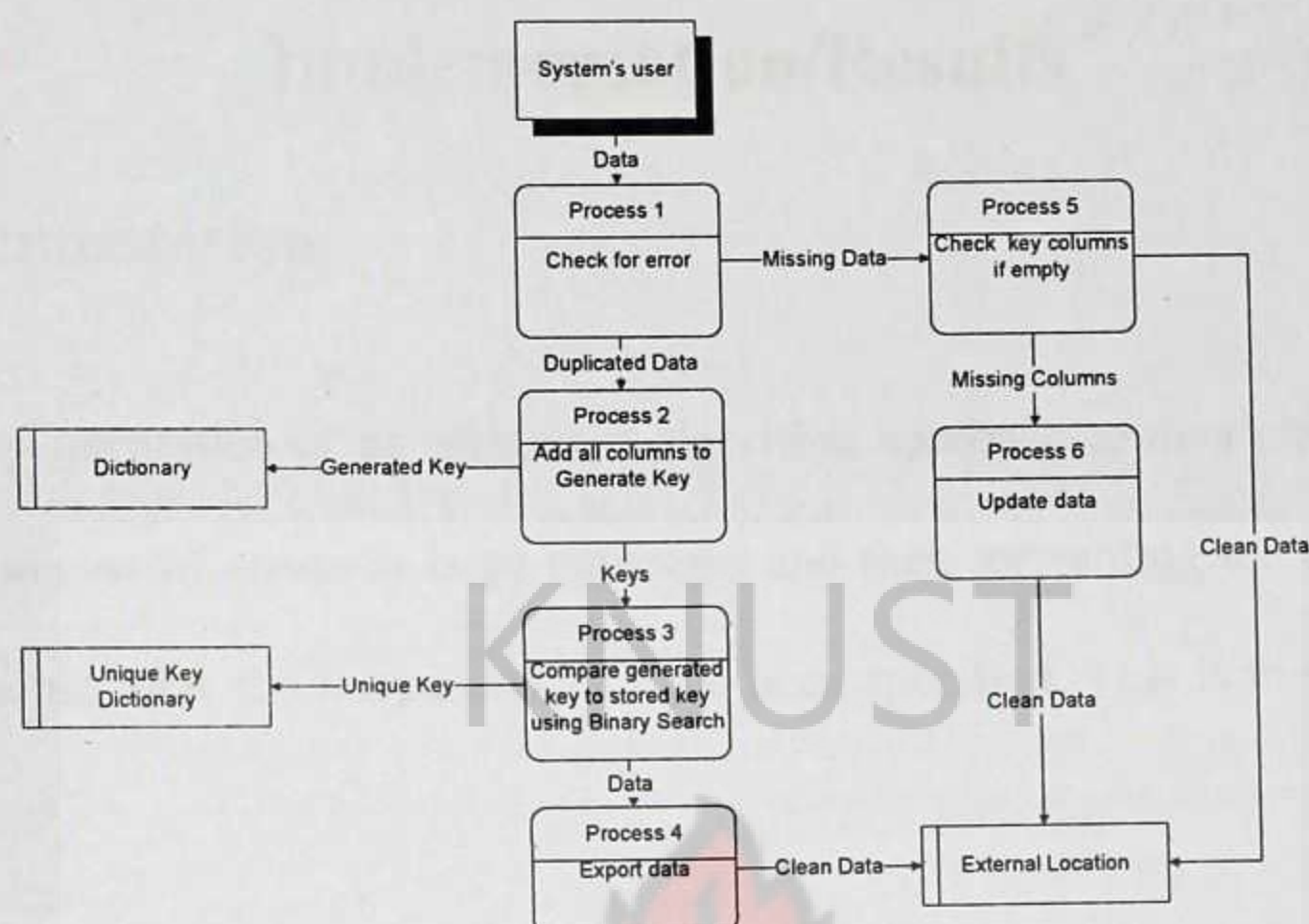


Figure 4.10: Level 1 Data Flow Diagram

4.9.10 Context Level diagram for the integrated cleansing system

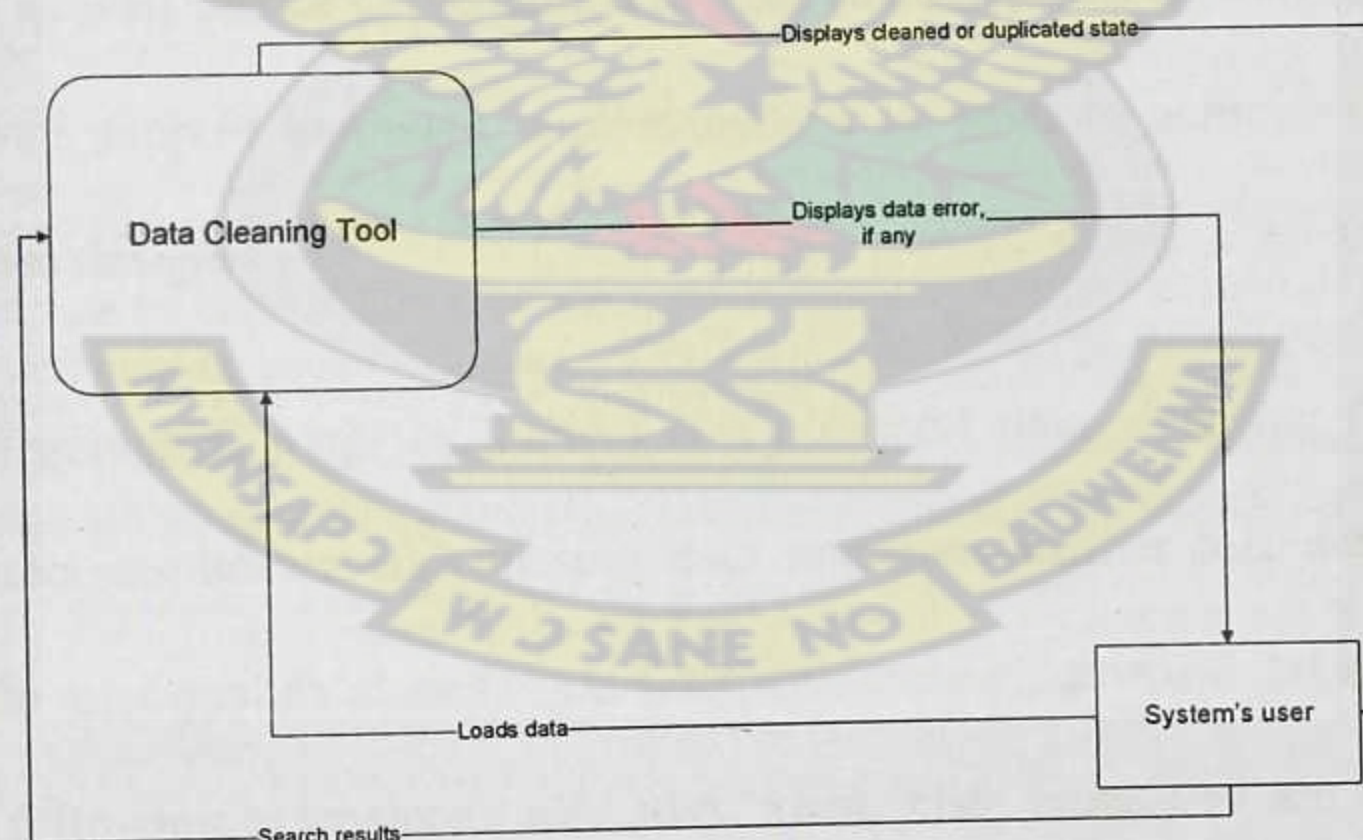


Figure 4.11: The Context Diagram

Figure 4.10 and figure 4.11 illustrate the data flow diagram (level 1) and the context level diagram respectively. The data flow describes the various processes involved in the cleansing process. The keys generated are all stored in respective dictionaries. On the other hand, the context level diagram depicts the interaction the user has with the system by sending or uploading the appropriate data and the responds the system provides to those request made.

Chapter five

Implementation/Results

5.0 Implementation

The implementation of an integrated algorithm approach to data cleansing enhances the cleansing process of errors in large databases and then forwarding the cleaned data into the data warehouse after the business rules have been specified. This is the hallmark of this chapter.

The implementation of the integrated algorithm for data cleansing is achieved through the coding of the pseudo-code algorithm inferred from the analysis in the previous chapter. The programming language being used is C# (C sharp). C# is used because it is very easy to implement and as well can be implemented in distributed systems or web based applications. This will as well support hardware implementation in order to achieve the system's data cleansing process design.

The *Integrated Cleaner (iCleaner)* is a powerful data cleansing tool developed to clean and correct duplicates and missing data and standardizes this data from MSSQL, MySQL, Oracle relational databases, Text files with these extensions: .txt and .csv and Excel files with the following extensions: .xls, .xlsx, xlsx, xlsb. Once you are able to obtain the necessary credentials (server name/instance, user name, and password) for the relational databases, (MSSQL, MySQL and Oracle) you will be able to access all the databases and the various tables within the databases can be imported for onward cleansing.

5.1 System Environment

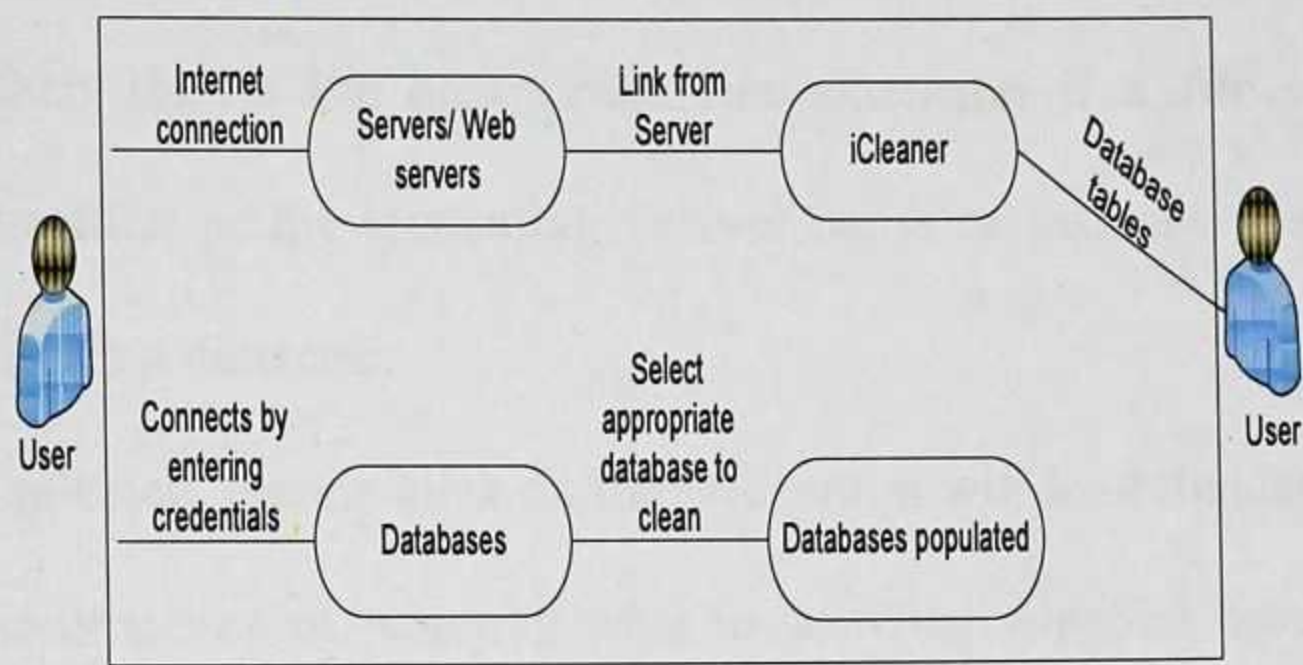


Figure 5.1: Use Case for user accessibility

The user will operate *iCleaner* either on the same server the database resides or will need to connect the server as illustrated in figure 5.1 and with the right credentials be able to access the databases on the server or web server.

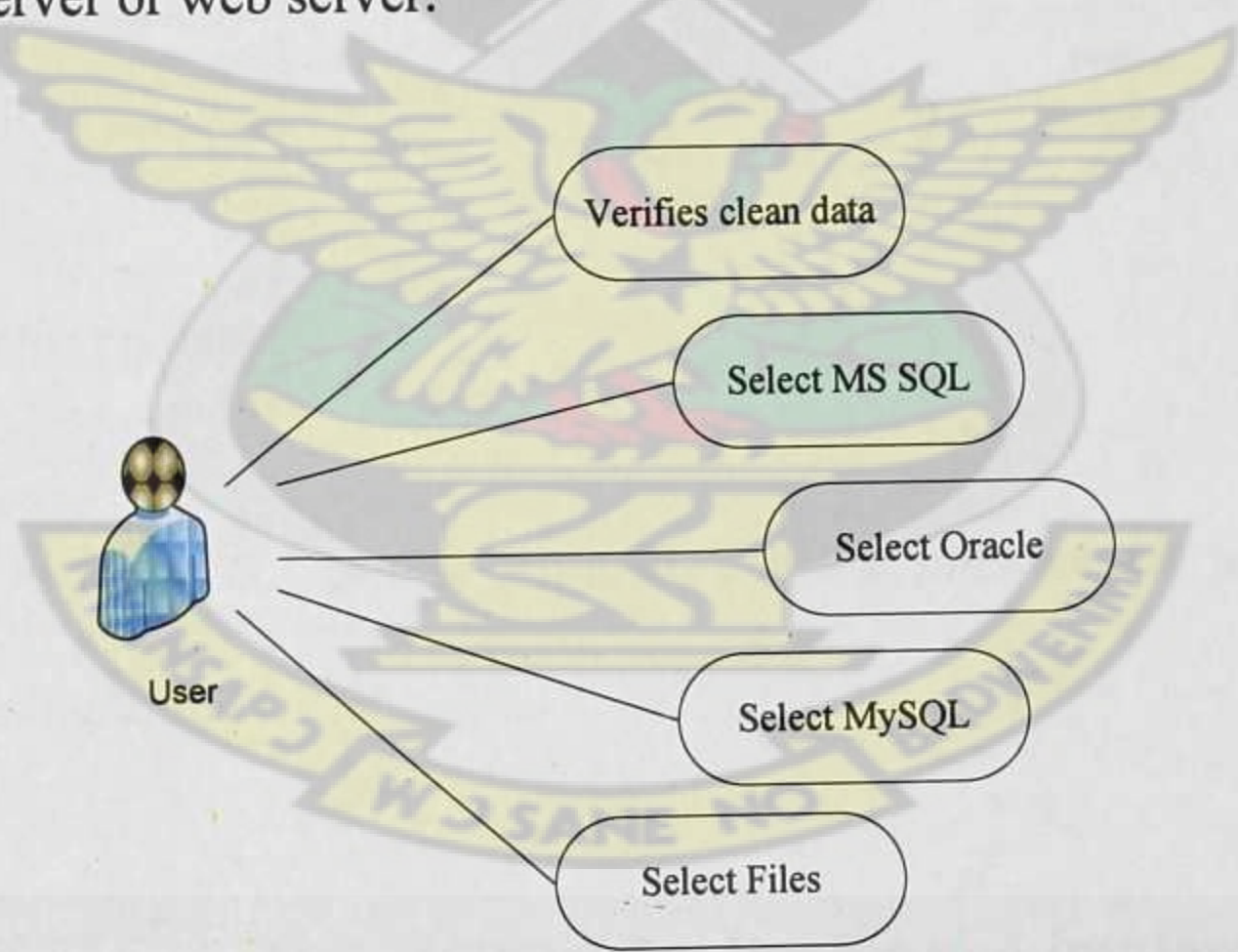


Figure 5.2: Sets of Use Cases for user connectivity

The user chooses to select a file (Excel or text) or any of the relational databases (MSSQL, MySQL, and Oracle) by clicking the product logos on the main page of the system as shown in figure 5.2. The following processes are then followed: figure 5.2 illustrates the processes of selecting files or databases to verifying the cleaned data.

- i. The *iCleaner* returns a report by displaying a dialog box to the user to select either the suitable file and input the suitable credentials for the relational databases.
- ii. The user then selects the appropriate file extension if a file is selected from a particular location or the credentials (server name or instance name, user name and password) if it's a database.
- iii. If a file is selected, then, a click on the OK button will load the data into the interface to begin the next step of selecting what to do. Otherwise the database to work on is selected and upon click the databases are displayed for the next step of selecting what to do (decide on type of error) to begin.
- iv. The user verifies the cleaned data and pushes it into the data warehouse or back to the cleansing process.



Figure 5.3: User operates on data

Before the use cases in figure 5.3 can be initiated, the user must access the main *iCleaner* page of the data cleansing tool.

5.2 Graphical User Interfaces (GUIs) for the System implementation

The following graphical user interfaces (GUIs) demonstrate the various procedures the user must implement to clean data from operational sources and sections of the Integrated cleansing tool. Brief descriptions are given in order to clarify how these interfaces are operated.

The Integrated data cleansing system, allows the user to begin by selecting the type of data and determine the source so as to load the data for the data cleansing process. As part of the limitations, the system will consider the cleansing of two major types of errors; namely, duplicates and missing data.

5.2.1 The Main Interface

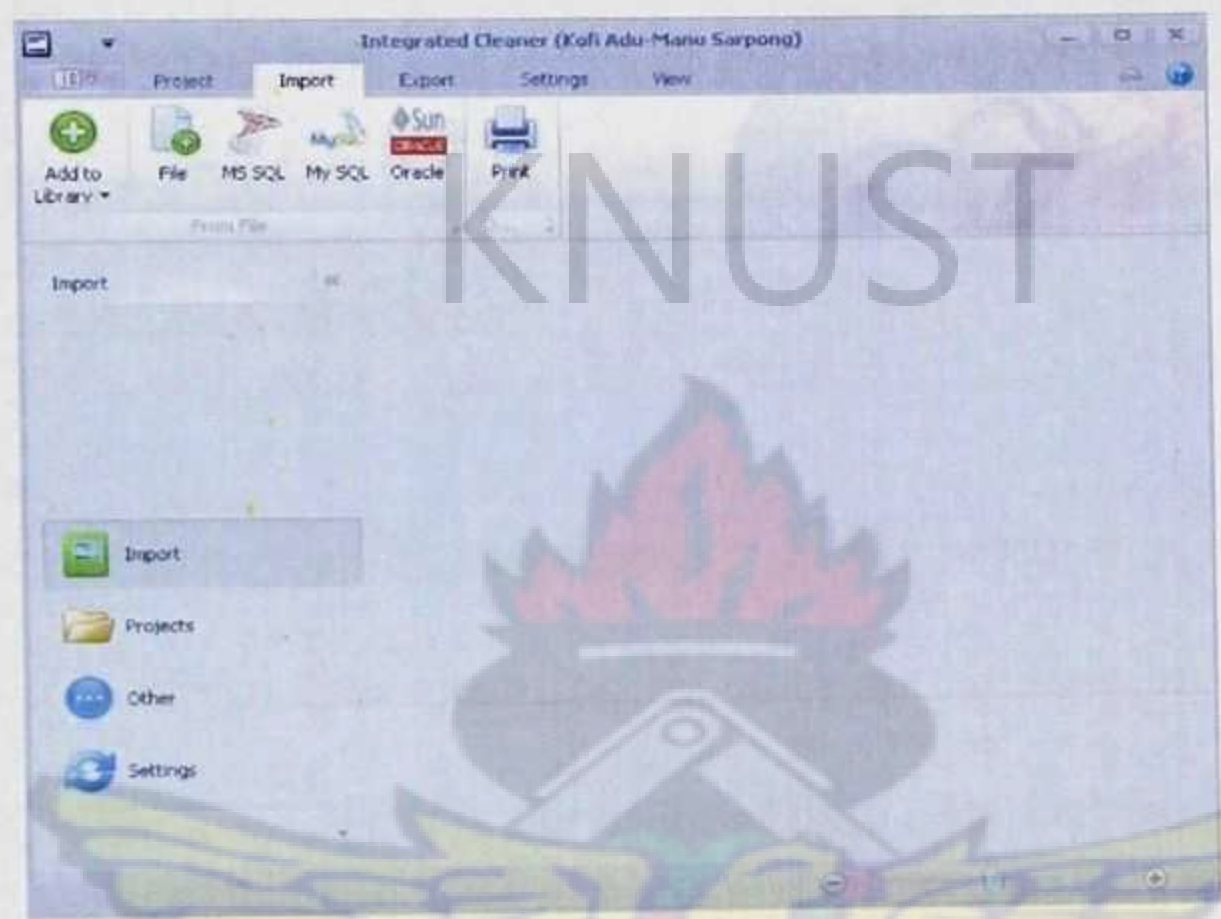


Figure 5.4: Main interface for iCleaner

Figure 5.4 shows the main interface of the cleansing tool that effectively implements the proposed algorithm. Users of the system have the option of selecting either a file or any of the relational databases.

5.2.2 Populating data from MSSQL using Select Database Dialog box

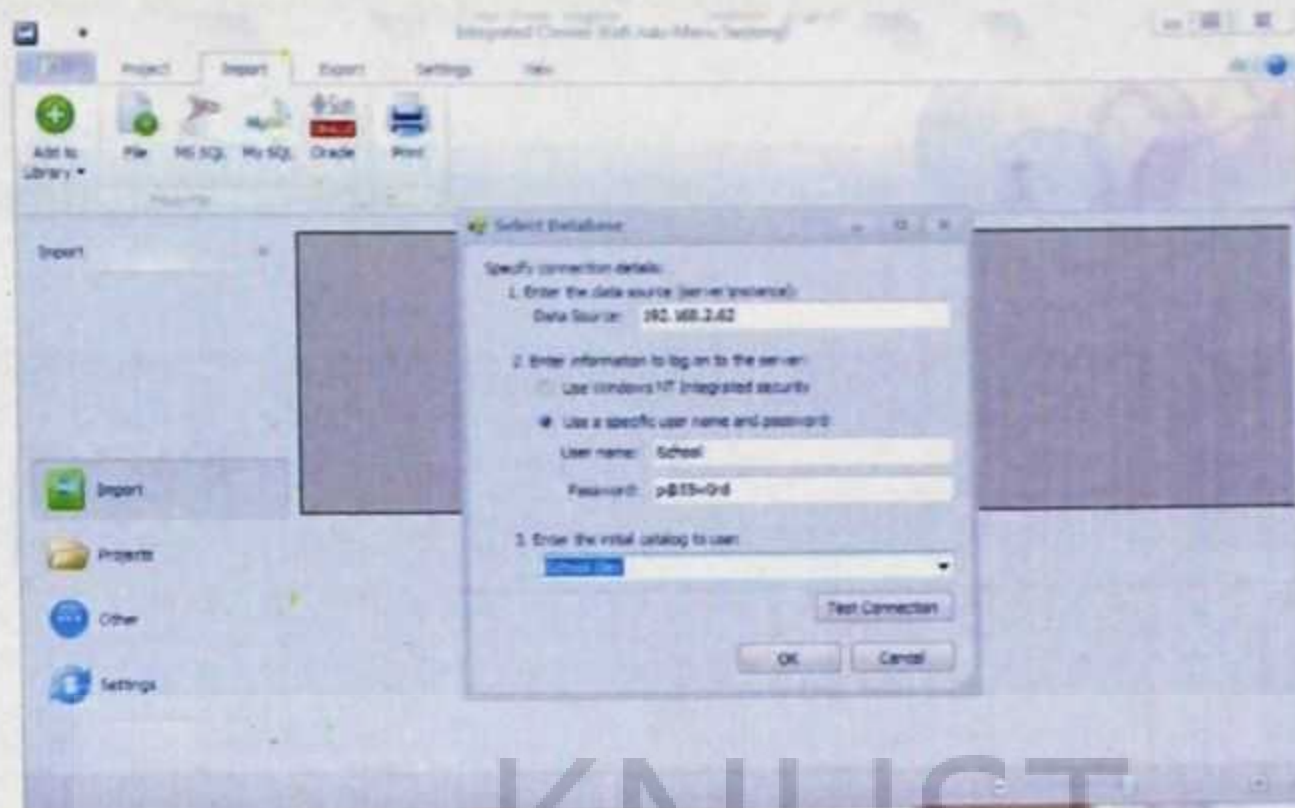


Figure 5.5: MSSQL connection interface

To populate data from MSSQL use figures 5.5. The screenshot illustrates the process a user goes through to perform data cleansing by entering database credentials. This data is obtained from MSSQL database. This is the original data obtained from the database. When the user clicks on the import button, a cache copy of the data is loaded into a temporary location for the cleansing process to start.

5.2.3 Populating data from MySQL using *MySQL Data* Dialog box

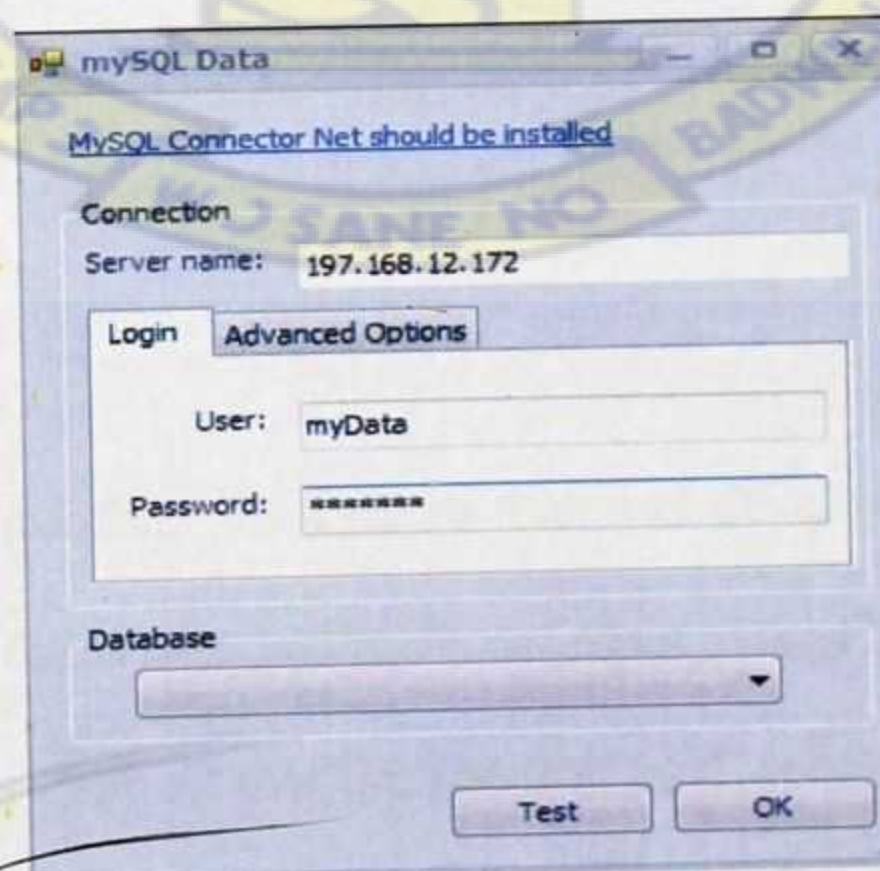


Figure 5.6: MySQL connection interface

Figure 5.6 shows the implementation interface for extracting data from MySQL database. The port number should be verified at the Advanced Options.

5.2.4 Data import interface

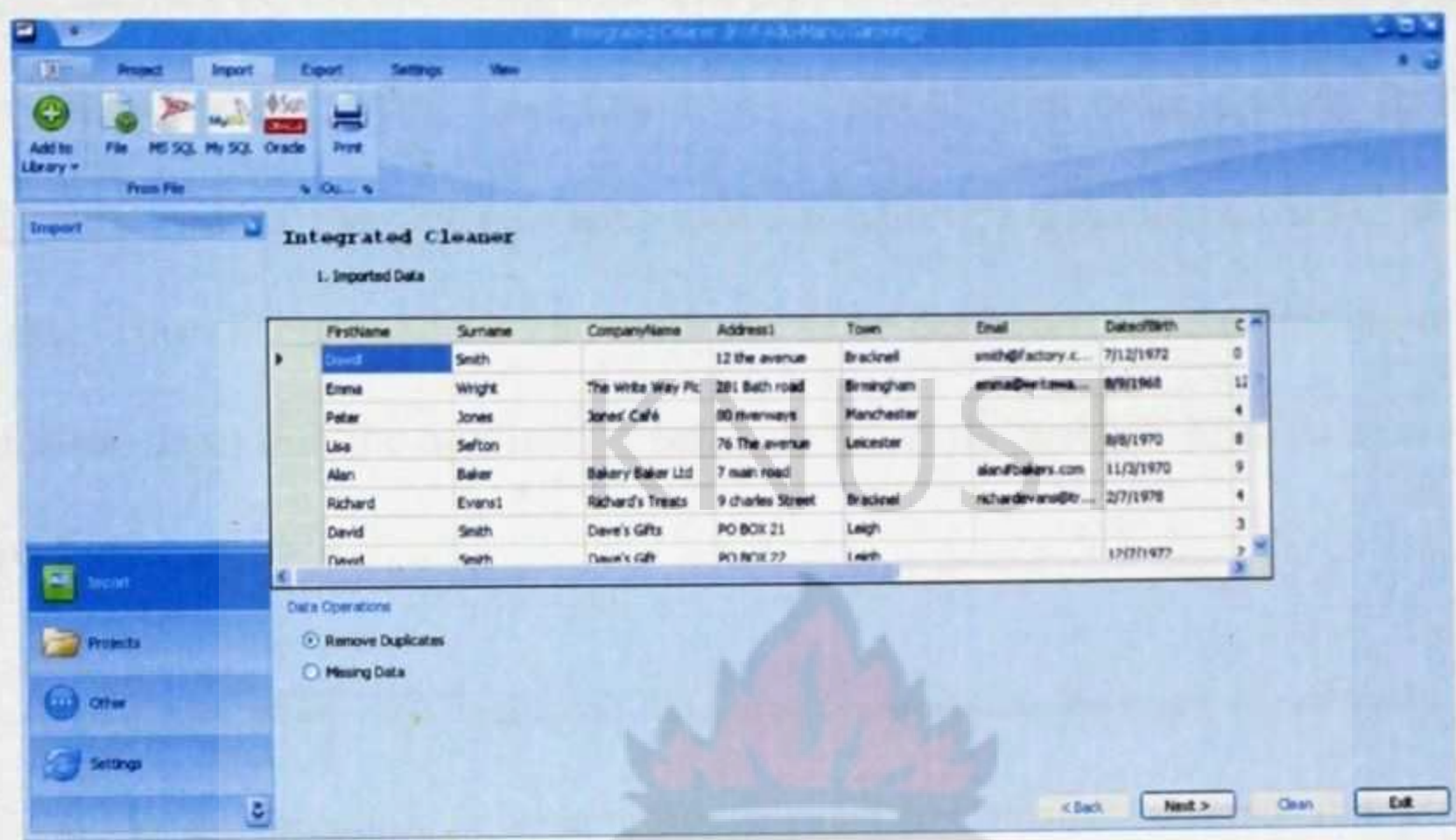


Figure 5.7: Main interface for cleansing errors

The figure 5.7 above illustrates the main interface for beginning the cleansing process. The imported data is cache copy of the data loaded from the operational source. The data operation to be performed is selected and next button pressed for the process to continue until finally cleaned.

5.3 Testing

In implementing the algorithm, the system that was developed has been tested with different file formats. Each of these files had different input size. The differences in the time taken for the cleansing process showed that the running time deduced above is quadratic. It was realised that the smaller the input size, the smaller the time it takes the system to remove either missing data or duplicates.

5.3.1 Expected performance for detecting duplicates

In order for the tool developed to meet its expected performance for detecting and cleansing duplicates, the cleansing tool will take advantage of the advances in processor and memory development in the near future. As a result, with more Central Processing Unit (CPU) on the system, the data cleansing tool can achieve a variable amount of work done in a faster rate. This characteristic is known as the scale-up factor, i.e. the amount of speed up or gained throughput that the application achieves when the number of CPU's on the system is increased.

The application also loads the data and creates a cache copy in memory and uses this cache copy in its operation. It is therefore relevant to consider memory size especially when the data to be loaded is huge. It will perform better when a considerable memory size is freed for such a purpose.

The application was run with several data sets (database files and text files) sizes and the times recorded at the entire given test runs. The table 5.1 and 5.2 below shows the data sizes, the average loading and cleansing times. A graphical representation of the data is shown in figure 5.8 and figure 5.9 respectively. It can be realized that the size of the data to be cleansed has determined the cleansing time. From the table 5.1, the minimum data size used in the testing is 10KB and the maximum data size is 2.45GB, with average cleansing time 0.1 seconds and 45 seconds respectively.

Table 5.1: Test with two CPUs

Data Size	Average Loading time (seconds)	Average Cleansing time (seconds)
10KB	0.1	0.1
100KB	0.1	0.3
200KB	0.2	0.5
10MB	0.8	7
100MB	6	10
255MB	9.8	13
498MB	12	17
798MB	15.6	23
1GB	18	27
1.45GB	22	32
2GB	27	38.7
2.45GB	30	45

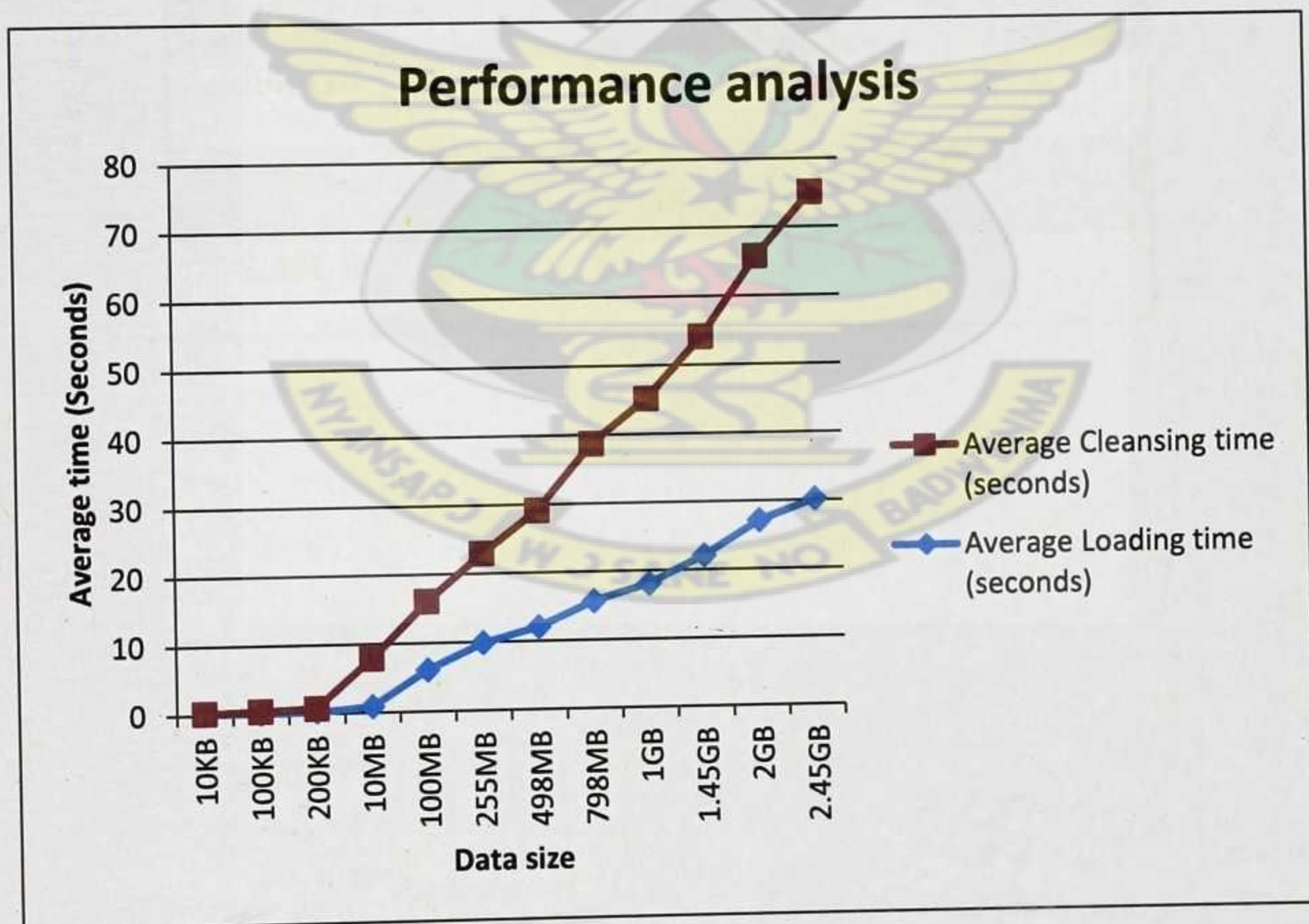


Figure 5.8: Graph of duplicate test with two CPUs

Table 5.2: duplicate test five CPUs

Data Size	Average Loading time (seconds)	Average Cleansing time (seconds)
10KB	0.03	0.03
100KB	0.03	0.1
200KB	0.06	0.17
10MB	0.26	2.33
100MB	2	3.33
255MB	3.27	4.33
498MB	4	5.67
798MB	5.2	7.67
1GB	6	9
1.45GB	7.33	10.67
2GB	9	12.9
2.45GB	10	15

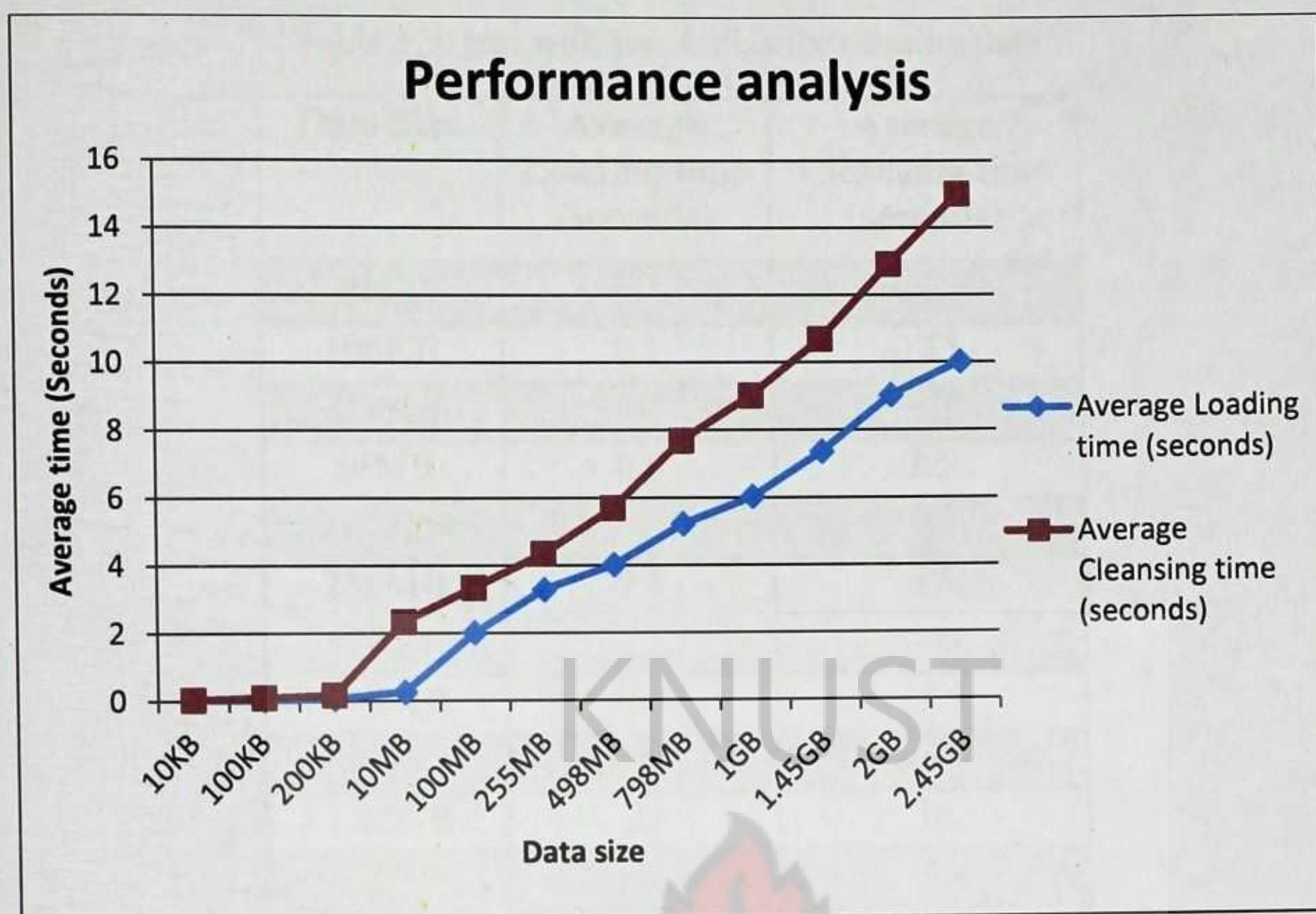


Figure 5.9: Graph of duplicate test with five CPUs

5.3.2 Expected Performance for Missing data

With respect to missing data set, it depends on the “key column” one will be using as the search criteria during the process. Here, either all the columns in the table could be used or one or more could be used as the basis for searching for records with missing fields. It could be realized from the table 5.3 and table 5.4 that the same loading rate applies to the records that were loaded for testing for missing fields in data sets. The table 5.3 shows that the cleansing time was two times less than the cleansing time for identification of duplicates since in that instance; all columns are factored into the searching criteria. Figure 5.10 and Figure 5.11 shows the graph which represents a quadratic function.

Table 5.3: test with two CPUs for missing data

Data Size	Average Loading time (seconds)	Average Cleansing time (seconds)
10KB	0.1	0.05
100KB	0.1	0.15
200KB	0.2	0.25
10MB	0.8	3.5
100MB	6	5
255MB	9.8	6.5
498MB	12	8.5
798MB	15.6	11.5
1GB	18	13.5
1.45GB	22	16
2GB	27	19.4
2.45GB	30	22.5

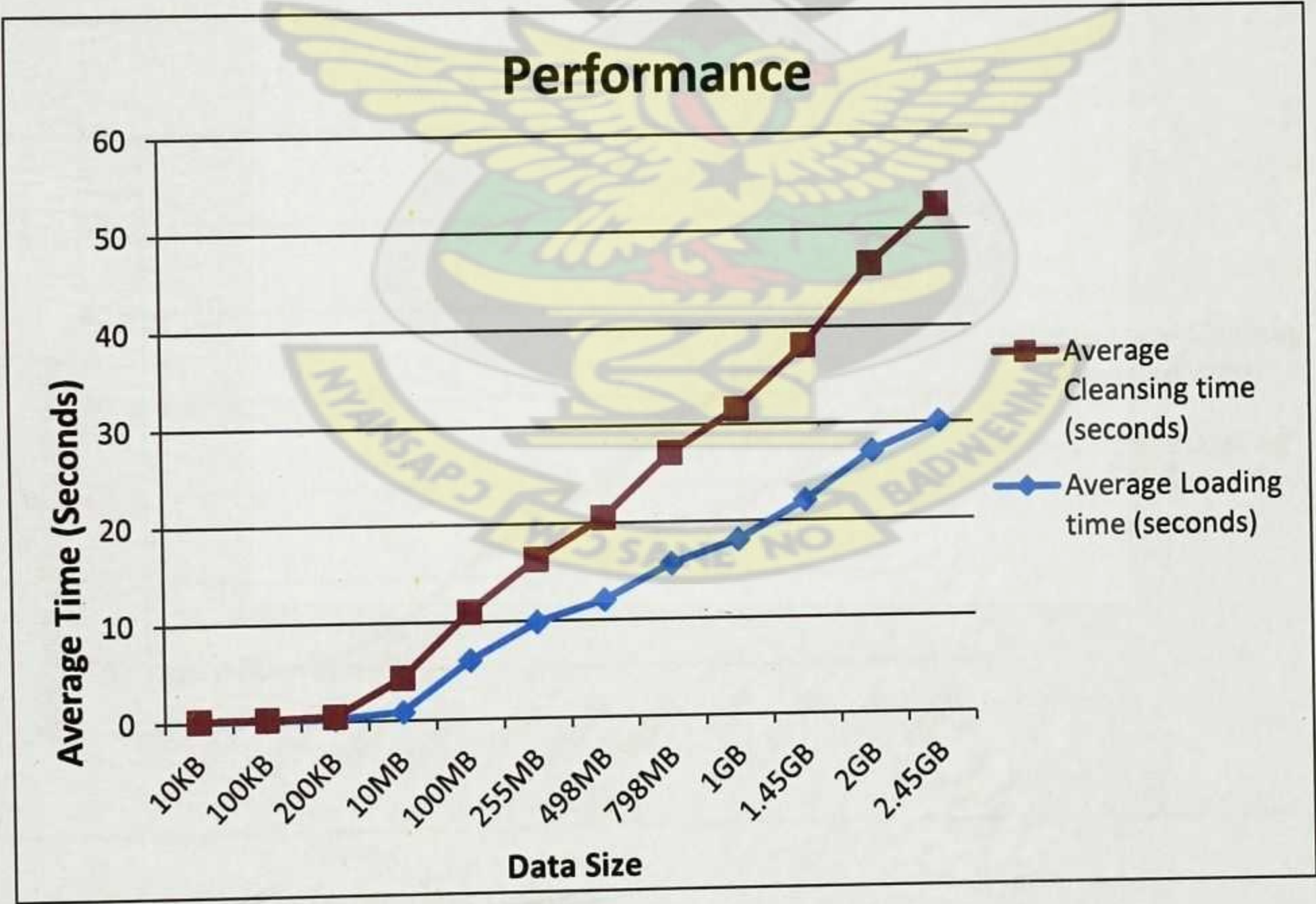


Figure 5.10: graph of missing data fields

Table 5.4: missing data test with five CPUs

Data Size	Average Loading time (seconds)	Average Cleansing time (seconds)
10KB	0.1	0.02
100KB	0.1	0.05
200KB	0.2	0.09
10MB	0.8	1.17
100MB	6	1.17
255MB	9.8	2.17
498MB	12	2.84
798MB	15.6	3.84
1GB	18	4.5
1.45GB	22	5.3
2GB	27	6.45
2.45GB	30	7.5

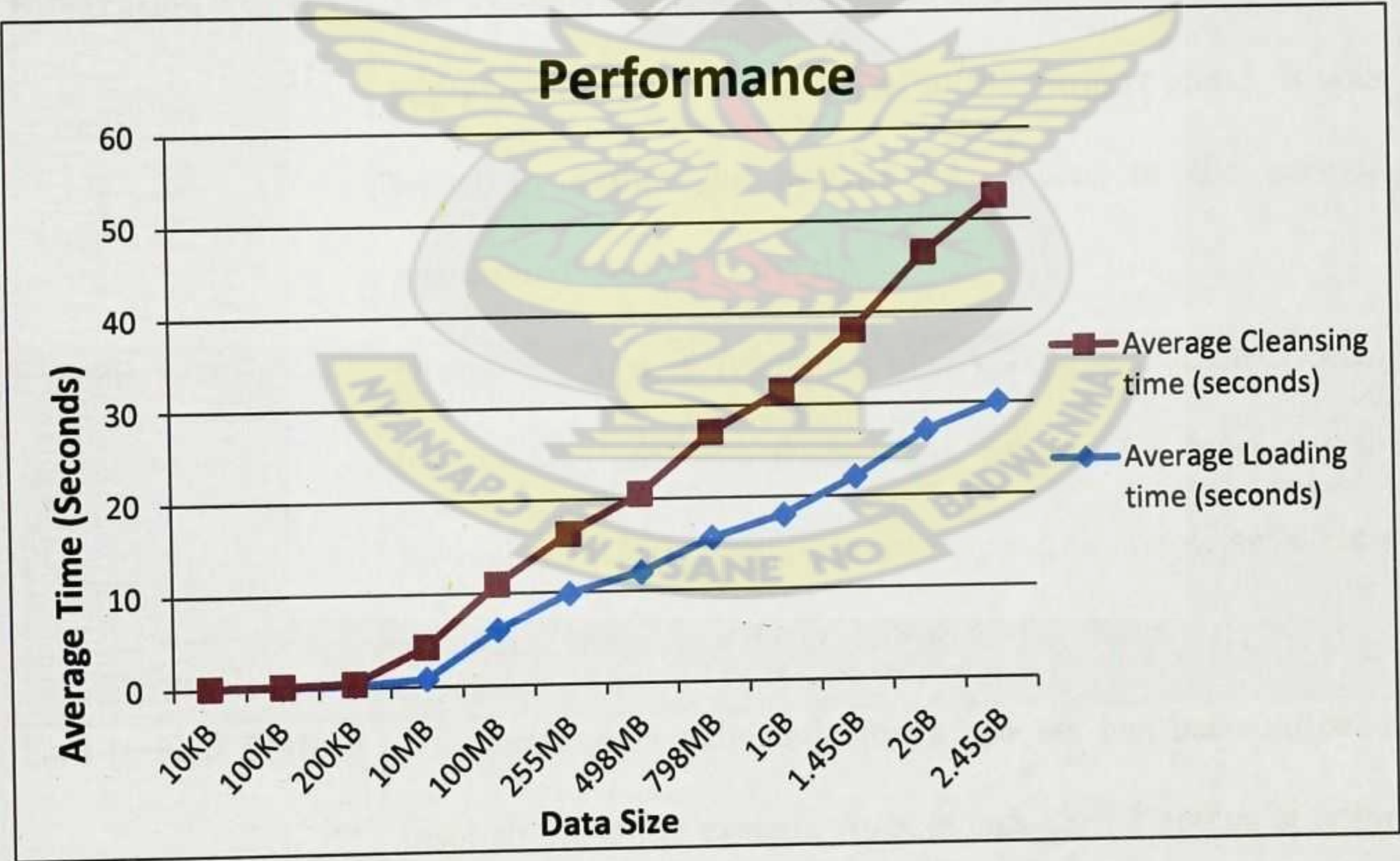


Figure 5.11: graph of missing data fields

5.3.3 Test Phases

The algorithm was tested based on the separate phases of testing which are designated on the timeline within the overall System Test phase. The following is a list of the test phases included in the overall System Test timeframe. A brief description of the test performed is described in the tables 5.5 and 5.6. The phases of testing in System Integration Testing include the integration, system testing, user testing, security and end-to-end testing.

5.3.4 Testing Descriptions

Table 5.5: System Integration Testing

Test Type	Focus
Integration Testing	The system was tested to find errors in complete functions and processes within and between the various units. It was tested to ensure the system was linked to the various operational sources correctly.
System Testing	During the system testing, it was validated that the system functionality performs as specified. In order for the system to be deployed effectively its functional requirements have been defined to show how the system should perform.
End-to-End Testing	An operation is validated after a process has been initiated through the entire system. At both entry point and exist point the system was tested.

Table 5.6: System Integration Testing (Continuation)

Test Type	Focus
Security Testing	Eliminates security accessibility errors.
User Testing	Users have been incorporated to test the system in order to validate the functional requirements.

After testing the system, a comparative analysis between the four systems reviewed were compared to the system developed. The table 5.7 below shows the comparison between the systems.

Table 5.7: Comparison between reviewed systems and the proposed system

Parameter	Porters wheel	AJAX	IntelliClean	ARKTOS	iCleaner
Interactivity	It is very interactive; hence easy to use	Complex interface and hence not friendly to non-technical persons.	Interactive with end user. However, requires little input from end-users	Highly interactive; it has graphical interfaces for loading and executing validations on loaded files.	Highly interactive and has graphical user interface for loading, cleansing, validating and exporting data files, making it easy to use.
Data format/structure	Text	Text	Text	Text	Text and Multimedia databases
Human dependency	High human dependency for exceptional errors.	High human dependency. Example; evaluation and validation of errors is fully dependent on human expert.	Very minimal because of the expert module embedded in the system.	Although the system has complex modules for dealing with duplicates, however, there is a high dependency on human experts for error correction.	There is simple method adapted for error detection. This is very little human intervention because the system does the evaluation of errors and the export is done by human expert.
Maintenance	Not considered	Not considered	Not considered	Not considered	Considered
Detailed cleansing process	Not considered	Not considered	Not considered	Not considered	Considered

5.4 Deployment

The proposed algorithm after testing it successfully needs to be deployed. Normally, deployment diagrams are used for describing the hardware components and show where software components are deployed. The reasons for deploying this data cleansing tool are to visualize hardware topology of the system and describe the hardware components used to deploy the software components.

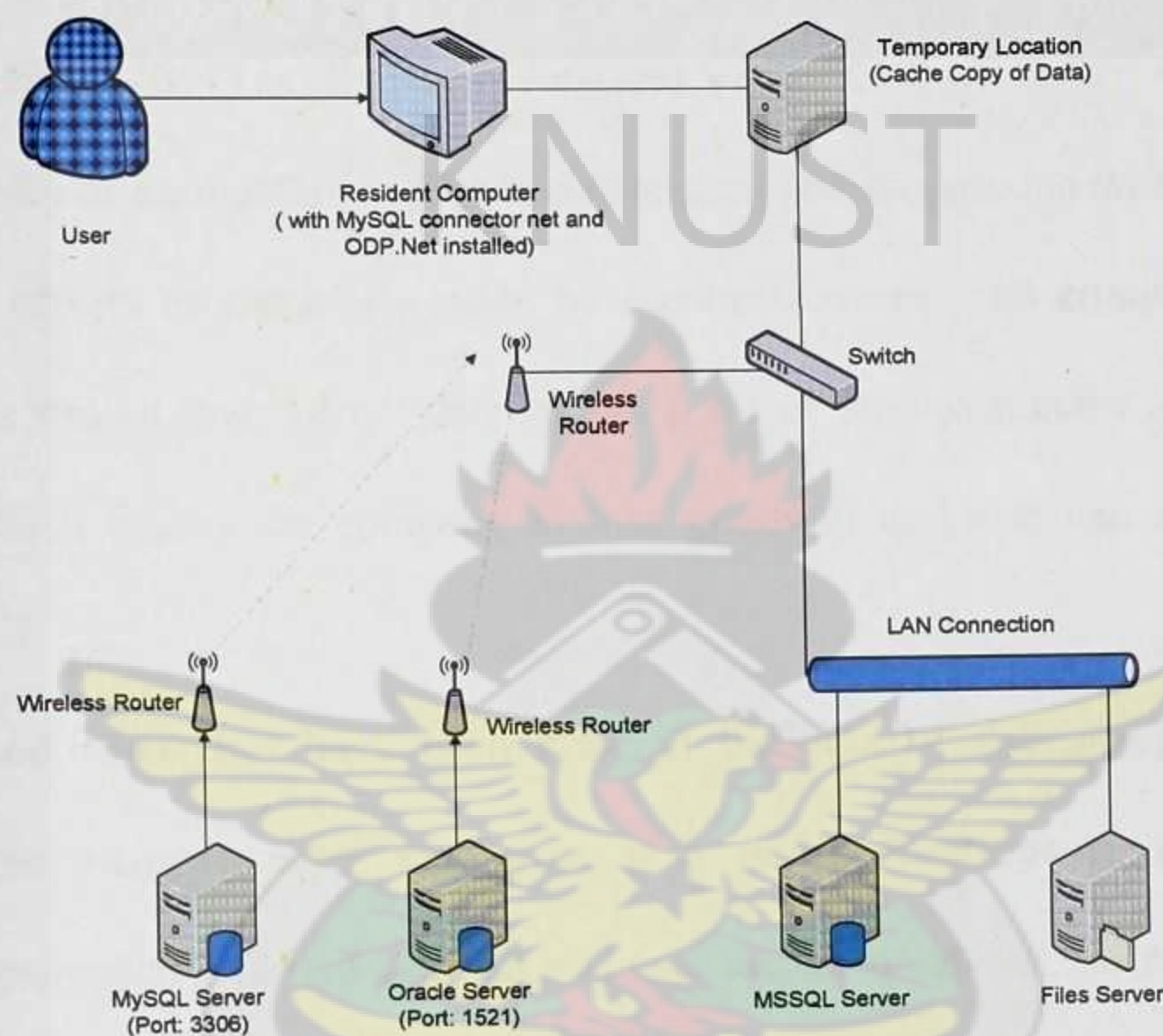


Figure 5.12: Deployment Diagram

Figure 5.12 describes the physical components (hardware) and their distribution and association. The components the system will need for its management include: Network connection (wireless or cable), a computer system (a monitor and a system unit where a cache copy of the data will be kept) and a server (MySQL Server, Oracle Server, MSSQL Server or a File Server). The components or devices can be accessed only when the network connection is established. This gives access to the cleansing system to able one access the server. On the contrary, when the cleansing system is resident on the same server, then, the connection is possibly not needed.

Chapter six

6.0 Summary and Conclusion

Poor quality data costs businesses vast amounts of money every year. Defective data leads poor business decisions, and inferior customer relationship management. Data are the core business asset that needs to be managed if an organization is to generate a return from it (Hamad and Jihad, 2011) as cited by Porwal and Vora (2013).

The idea of duplication in files and databases and the missing datasets and other errors has adverse effects on decisions made by business owners and companies on the whole. Research has shown that “dirty” data causes a lot of inconsistencies in data reporting and analyzing which causes the company to lose goodwill and also can affect their customer satisfaction.

Several researches have been done in the area of data cleansing in the past to overcome the inconsistencies that arise as a result of erroneous data. Most of these researchers presented different methods and approaches in solving this problem.

The main goal for this research paper was to provide a new approach to efficiently perform data cleansing in its entirety. The goals for each of the phases of the algorithm are to meet the desired target of removing erroneous values from databases extracted from different sources. In this research, two major errors were identified and as a result a complete algorithm was produced to solve these errors to enhance the quality of data. The algorithm presented and explored in this research relates to error detection and data cleansing generating a key and storing in a dictionary. The above algorithm work as an important tool in the area of data warehouses to obtain quality data.

In this research, the algorithm was restricted to identification of two (2) specific errors (duplicate and missing data). One major error which was a challenge to be resolved in this research was incomplete data. It was realized that there could be duplication of data if

incomplete data was not taken care of. Therefore, we recommend that it should be considered in future research. The research has sought to promote among other things the incorporation of the conceptual framework into a variety of methods that are used to identify a particular type of error in data sets.

6.1 Contributions to the research

This research presented a new perspective in terms of approach, to address the problem of erroneous data, taking into account every field within the databases loaded to undergo the cleansing process.

In contrast to the works of Rahm and Do (2000), Vassiliadie et al.,(2001), Bradji and Boufaida (2011), Elmasri and Navathe (2004), a new framework has been proposed to cater for the lapses that was presented in their works. The study took into consideration by replacing the huge and complex calculations used by other algorithms and adapted one with lesser calculations since the study took advantage of dictionary strategy which was extremely useful. The difference in computational time between the existing works and the proposed work largely depends on the amount of input data size loaded into the system for the cleansing operation to be performed and the system requirements (i.e. processor speed and memory size). A comparative analysis was done to show how the proposed system was measured with the reviewed systems by using the same parameters (refer to table 5.7).

In the removal of duplicates, specific fields were not chosen to check for duplication but rather all which makes this algorithm very robust. This overcomes the problem of using one field such as "name" in the work of Arora et al. (2009) as cited by Porwal and Vora (2013).

Another milestone reached in this research work is the proposal of a novel framework which showed clearly that all the frameworks reviewed had not considered one of the most important aspects of the data cleansing process – maintenance. An attempt was made in the

development of current system to maintain the clean data. The automatic maintenance schedule feature has been fully developed and tested. The system checks the state of the file after a specified time period by comparing the clean data at that time to its original state properties when it was cleaned and exported to the central repository.

Cleansing is important but without maintenance it becomes fruitless effort. This research also pointed out weakness and strengths in the existing frameworks which aided the research to conceptualise a novel framework to fill out the gaps that remained in the data cleansing process in a data warehouse.

6.2 Recommendation and Future work

This research proposed an algorithm that was implemented into developing a data cleansing tool which is recommended to support the data cleansing process. In the study, the algorithm was implemented in the development of a system that extracts data from a source file or database (internal or external) and creates a cache copy in memory to enhance the speed with which the cleansing tool works.

Future work can involve a further research into the maintenance aspect of the framework since little work on that was considered in this research work. Also, the introduction of expert module in the framework should be considered in future research to do away with the human intervention completely from the data cleansing process. Here, it is expected that, the expert module would periodically clean any inconsistencies within the cleansed data and have expert judgement concerning any other error the cleansing tool is unable to fix during its operation.

Further research ~~trend~~ should be tailored to include the investigation into a technique to support various scripts on error instances that could be written and embedded into the algorithm to support the cleansing process.

Incomplete data in some instances needs to be tracked since it can give rise to duplicate data. It is therefore recommended that much attention be given at the data entry level and if possible a new algorithm be proposed in future research to cater for incomplete data.

To enable effective reporting of business analysts in Ghana and some parts of the Africa, it is recommended that the data cleansing tool be extended for use in most organizations in Ghana that deal with huge data obtained from different operational sources (e.g. databases) by these organizations.



REFERENCES

- Louardi BRADJI, Mahmoud BOUFAIDA. (2011). Open User Involvement in Data Cleaning for Data Warehouse Quality. *International Journal of Digital Information and Wireless Communications (IJDWC)* 1(2), pp. 573.
- Rahm, E., Do, H.H. (2000). Data Cleaning: Problems and Current Approaches. *IEEE Data Engineering Bull.* Vol 23 No. 4, pp. 3-13
- Chapman, A. (2005). Principles and Methods of Data Cleaning – Primary Species and Species-Occurrence Data, Version 1.0. Report for the Global Biodiversity Information Facility, Copenhagen , pp. 1.
- Jonathan I. Maletic, Andrian Marcus. (2000). Data Cleansing: Beyond Integrity Analysis, pp. 8.
- Kimball, R., "Dealing with Dirty Data. (September 1996). "DBMS, vol. 9, no. 10, pp. 55.
- Wang R, English J. (1995). A Framework for Analysis of Data Quality Research. *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4 , pp. 623-639.
- Ramez Elmasri, Shamkant B. Navathe. (2004). *Fundamentals of Database Systems*. Boston: Pearson Education, Inc, pp. 912 - 913.
- Heiko Muller, Johann-Christoph Freytag. (2004). Problems, Methods, and Challenges in Comprehensive Data Cleansing, pp. 21.
- Guide, O. D. (2002). Oracle Corporation. Retrieved April 29, 2012, from <http://docs.oracle.com>:
http://docs.oracle.com/cd/B10501_01/server.920/a96520/concept.htm
- Mallach, E. G. (2000). *Decision Support and Data Warehouse Systems*. Lowell: David Kendrick Brake. pp. 14.
- Vassiliads, P. (2009): A Survey of Extract-Transform-Load Technology. In *International Journal of Data Warehousing & Mining*, vol.5, no. 3, pp. 1-27.
- Nemani, R.R., Konda, R. (2011). A Framework for Data Quality in Data Warehousing. *UNISCON 2009, LNBIP 20*, Springer Heidelberg, pp. 292-297
- R.Kavitha Kumar, DR. RM.Chadracharan. (2011). Attribute Correction-Data Cleaning Using Association Rule And Clustering Methods. *International Journal of Data Mining & Knowledge Management Process (IJDMP)* Vol.1, No.2, March , 22, pp 23.
- Matyia, D. (2007). Applications of data mining algorithms to analysis of medical data. Master Thesis, ~~Software~~ Engineering, Thesis no: MSE-2007. Blekinge Institute of Technology, pp. 41.
- Lin, J.H., Haug, P.J. (2008). Exploiting missing clinical data in Bayesian network modelling for predicting medical problems. In *Journal of Biomedical Informatics*, vol. 41, no. 1, pp. 1-14

- Hernandez, M. A. and Stolfo, J. S. (1998). "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem," *Journal of Data Mining and Knowledge Discovery*, vol. 2, pp. 9-37.
- S. Chaudhuri and U. Dayal. (1997). An overview of data warehousing and OLAP technology. In *SIGMOD Record*. pp. 65-74.
- Fox, C., Levitin, A., and Redman, T., (1994). "The notion of Data and Its Quality Dimensions," *Information Processing and Management*, vol. 30, no. 1, pp. 9-19.
- Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., & Lee, D. (2003). A Taxonomy of dirty data, *Data Mining and Knowledge Discovery*; 7(1): pp 81-99.
- Jonathan I. Maletic, Andrian Marcus. (LLC 2010). *Data Cleansing: A Prelude to Knowledge Discovery, Data Mining and Knowledge Discovery*, pp. 20.
- Herbert, K.G., Wang, J.T.L. (2007). Biological data cleaning: a case study. In *International Journal of Information Quality*, vol. 1, number. 1, pp. 60-82.
- F. Naumann, *Quality-Driven Query Answering for Integrated Information Systems*, Lecture Notes in Computer Science, LNCS 2261, Springer, 2002. pp. 34.
- Ali, K.; Warraich, M.A.; (June 2010), "A framework to implement data cleaning in enterprise data warehouse for robust data quality," *Information and Emerging Technologies (ICIET)*, 2010 International Conference on, vol., no., pp.1-6.pp. 2.
- Herbert, K.G., Wang, J.T.L. (2007). Biological data cleaning: a case study. In *Int. J. of Information Quality*, vol. 1, number. 1, pp. 60-82
- Vijayshankar Raman and Joseph M. (2001). *Hellerstein. Potter's Wheel: An Interactive Data Cleaning System*. Proceedings of the 27th VLDB Conference, Roma, Italy, pp. 27.
- H. Galhardas, D. Florescu, D. Shasha, E. Simon, C.-A. Saita. (2001). Declarative data cleaning: Language, model, and algorithms. Proceedings of the 27th VLDB Conference, Roma, Italy, pp. 32-34.
- Millicom Ghana Ltd (2010). *Data Warehouse Documentation*, pp. 7-8
- P. Vassiliadis, Z.a Vagena, S. Skiadopoulos, N. Karayannidis, T. Sellis. (2001). ARKTOS: towards the modeling, design, control and execution of ETL Processes . *Information Systems*, Vol.26 , pp. 537-556.
- H. Galhards, D. Florescu, D. Shasha, E. Simon. (May 2000). AJAX: An extensible data cleaning tool. Proceedings of the ACM SIGMOD on Management of data, Dallas, TX USA, pp. 21-22.
- tech-faq. (2012). Retrieved May 30, 2012, from [www.tech-faq.com](http://www.tech-faq.com/data-cleansing.html): <http://www.tech-faq.com/data-cleansing.html>
- Bruade, E. J. (2001). *Software Engineering: An Object-Oriented Perspective*. Phoenix- USA: John Wily & Sons, Inc. pp. 30-35.

- Reaves, M. J. (2003). Software Requirements Specification (Web Accessible Alumni Database). Jacksonville. pp. 3-9.
- Monge, A. E. (2000). Matching Algorithms within a Duplicate Detection System. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, pp. 18-19.
- Mohan, S. D and Willshire M. J. (n.d). Databryte: A proposed data warehouse cleaning framework. University of Phonix and Colorado Technical University, pp. 248-250
- Deku JerryYao,Mohammad Sarrab and Hamza Aldabbas (2012).Three Tier level Data Warehouse Architecture for Ghanaian Petroleum Industry. International Journal of Database Management Systems (IJDMS) Vol.4, No.5, pp 1
- Bank of Ghana. (2007). Development of banks and non-banks financial institutions. Annual Report and Accounts, pp. 43
- Diako. T, Jean-Marc Petit, Sylvie S. (n.d). Edditing rules: discovery and application to data cleansing,. pp. 1-20
- Bhushan and Parikshit (2010). System Analysis and Design Flexibility in Approach based on the product definition, Maharashtra Institute of Technology Pune, India, pp. 45-50.
- Dongre Kuldeep (2004). Data cleansing strategies, pp. 10.
- Dr. Mortadha M. Hamad, Alaa Abdulkar Jihad (2011). An Enhanced Technique to Clean Data in the Data Warehouse. IEEE, pp 1
- Rajiv Arora,PayalPahwa and ShubhaBansal (2009). Alliance Rules for Data Warehouse Cleansing. IEEE Press,pp. 743-747
- Sonal Porwal and Deepali Vora (2013). A Comparative Analysis of Data Cleaning Approaches to Dirty Data. Internal Journal of Computer Applications (0975 – 8887), Volume 62-No.17, pp. 1-5