Database Record Duplicate Detection System using Simil Algorithm

Jumoke Soyemi
Department of Computer Science The Federal Polytechnic, Ilaro Nigeria
Jumoke.soyemi@federalpolyilaro.edu.ng

James Adegboye
Department of Computer Science The Federal Polytechnic, Ilaro Nigeria
adegboyeoj@gmail.com

***Abstract*—As more data is populated into the database table, there is the tendency for the table to store duplicate or redundant record which results in the consumption of data spaces in the database and also in the storage device where the database resides. Despite the ever-increasing memory capacities of devices, significant benefits can still be realized by reducing the bytes size needed to represent an object when it is stored or retrieved from the database. This is quite beneficial to mobile devices with limited storage, reference data, e-mail, where sequences of large bytes are repeated and data transmitted over low-bandwidth or congested links. Reducing bytes equates to eliminating unneeded data, and there are numerous techniques for reducing redundancy when objects are stored or sent. This study implemented a database record duplicate detection system using simil algorithm as the reduction technique to achieve efficiency in detecting and reducing the presence of duplicate records in a database and hence provides an automated means of executing database record optimization.**

Keywords-database record; simil algorithm; prediction technique; duplicate detection system

I. INTRODUCTION

The need to detect and remove duplicate records that have to do with the same entity within a dataset is a task that is crucial. Linking data to detect duplicates is good in improving the quality and integrity of data which allow re-uses of existing data sources for future research work [1]. Duplicate detection is different from other Information Retrieval (IR) considering how it defines the similarity check between two or more documents. In many IR document, similarity refers to semantic relevance, which could be syntactically very different but still relevant. In contrast, the appearance of similarity in duplicate detection in early database research is quite conventional, and what it does it to discover syntactically almost-identical documents [2][3][4]. For other tasks that need to detect documents with intermediate level of similarity, there has not been much research done. The major problem that does arise is that, as more data is populated into a database table, there is the tendency for the table to store duplicate or redundant record which results in the consumption of data spaces in the database and also in the storage device the database resides. In reality, entities in the database have two and more representations. This is because records rarely share a common key and they exhibit errors thus making duplicate matching a task that is difficult [5][6]. Errors are introduced as the result of transcription errors, incomplete information, lack of standard formats, or any combination of these factors. In this paper, duplicate record detection system is proposed. The similarity metrics that are commonly used to detect similar field entries are covered with some algorithm used for duplicate detection to find approximately duplicates records in a database.

This study is designed based on Simil algorithm to identify the presence of duplicate records in a MySQL database local server installed on a computer system. It is meant to serve as a means of reducing the quantity of disk space that MySQL database server is hosting on a system and provide an automated means of executing database record optimization. The Database record duplicate detection system is designed using Java Object Oriented Programming Language and MySQL ODBC J-Connector Library for connecting to the database server of MySQL.

The rest of the study is divided into sections, with section 1 introducing the study, section 2 reviews the literature. Section 3 discusses the system design and its implementation and section 4 concludes the study.

II. REVIEW OF LITERATURE

The problem of finding duplicated documents is a research interest in database and web-search communities for quite some years now. The key applications developed related to this area include plagiarism detection in web publishing and redundancy detection in large datasets. The popular duplicate detection techniques are grouped into Fingerprint-based and Full text-based techniques [7].

A. Fingerprint-based Duplicate Detection

A fingerprint of a document is a set of integers, each of which is the hash value for a substring extracted from the document [8]. The term fingerprint refers only to document-level fingerprint while the term integer or hash value refers to hash function output, which is sometimes referred to as a fingerprint. For fast access during the query process, each integer is stored in an index. Also, to measure the similarities between two documents, the numbers of common integers are counted. Algorithms are different in their choices of hash functions, substring size, substring number, and substring selection strategy [9].

- Hash Function. This is used to generate hash values for substrings. Popular hash functions include NIST's SHA1 and Rabin. Other hashing functions are can also perform the same task as long as they are reproducible and with a low rate of hash collision.
- Substring Size. This is defined by the length of each substring obtained from a document. The larger the size of substring, the more the chances of false negatives while the smaller the size, the more the chances of false positive in duplicate detection. Example, SCAM used a very small substring, word, as the unit for fingerprinting. Substrings of 3-5 words are reported as the best in literatures.
- Substring Number. This is the number of substrings extracted from a document to build a fingerprint. Some techniques used a fixed number of substrings for efficiency, example, I-Match, while many others used a variable number of substrings for a more accurate representation of the document. A smaller number of substrings have the risk of ignoring short documents and increasing false negatives.
- Substring Selection Strategy. This is the process of choosing which substrings to hash. Position-based strategy which is a category of substring selection picks substrings on the basis of their offsets in a document, sentence or paragraph. It includes full fingerprinting, non-overlapping fingerprinting, and overlapping fingerprinting. It is commonly used due to the simplicity. Hash-value-based strategy is also popular.

B. Duplicate Detection using Full-Text

Duplicate Duplicate detection using full-text adapts methods initially designed for search engines. An example is vector-space model, which treats a document as bag-of-words, with term weights determined by *term frequency*-inverse document frequency (TF-IDF) values, and similarity determined by cosine similarity. "Traditional cosine-similarity measure focuses on finding a semantic relevant document while near-duplicate detection focuses more on syntactic similarity [6]. The identity measure proposed emphasizes that the gap between rare words' term frequency in two documents should be smaller than that between common words' and their best ranking is giving by a term weighting function biased towards rare terms" [7]. This study by [7] employed model based on statistical translation to find the probability that a sentence in a document is a translation of another sentence in another document.

C. Duplicate Detection Algorithm

There are several numbers of duplicate detection algorithms but this study discusses the few of them that are effective and commonly.

- Jaccard Similarity Algorithm. Most Tanimoto similarity and Tanimoto distance are synonyms with Jaccard similarity and Jaccard distance, but some are mathematically different. A similarity ratio is given over bitmaps, with each bit of a fixed-size array representing the presence or absence of a characteristic in the set modeled [10]. The ratio is defined as the number of common bits, divided by the number of non-zero bits set in either sample. When individual sample is modelled instead of a set of attributes, the value equates to the Jaccard coefficient of the two sets. Tanimoto defines a distance coefficient based on this ratio. This coefficient is not a distance metric but rather chosen to allow the possibility of two sets, which are different from each other, to be similar to a third [11].
- Tanimoto Similarity and Distance Algorithm. Most Tanimoto similarity and Tanimoto distance are synonyms with Jaccard similarity and Jaccard distance, but some are mathematically different. A similarity ratio is given over bitmaps, with each bit of a fixed-size array representing the presence or absence of a characteristic in the set modeled [10]. The ratio is defined as the number of common bits, divided by the number of non-zero bits set in either sample. When individual sample is modelled instead of a set of attributes, the value equates to the Jaccard coefficient of the two sets. Tanimoto defines a distance coefficient based on this ratio. This coefficient is not a distance metric but rather chosen to allow the possibility of two sets, which are different from each other, to be similar to a third [11].
- Euclidean Algorithm (EA). The method is efficient in computation of the greatest common divisor (GCD) of two numbers. EA is a step-by-step method of carrying out a computation based on well-defined rules, and is one of the oldest numerical algorithms that are popularly used. It can be used to reduce fractions to their simplest form, and is a part of many other number-theoretic and cryptographic calculations [12][13].

- Simil Algorithm. This is an algorithm that checks out for similar strings by calculating the similarity between the two strings. Over the recordings of pattern matching algorithm that ever exists, Simil is identified as one or the accurately measured algorithm for checking the identity of string patterns unlike Jaccard and other set driven algorithm which uses set union and intersection rules. Simil algorithm is based on the prefixes and surrogate computation method. Simil operates by calculating similarity that exists between two strings [1]. Typical uses of Simil include; data cleanup and bad data prevention from gaining access into the database.

The Simil algorithm checks out the longest common substring (LCS). It checks the left and right remainders recursively for the LCS until no more. It thus returns the similarity value between 0 and 1, which is achieved by dividing the sum of the lengths of the substrings by the lengths of the strings themselves.

Table 1 is a typical example for two spellings of the word Pennsylvania and Pencilvan. The algorithm finds the LCS lvan, and then repeats with the remaining strings until there are no further common substrings. Simil algorithm is basedon LCS, which performs excellently well [1].

Table 1. How Simil Algorithm works on strings

Word 1	Word 2	Common substring	Length
Pennsylvania	Pencilvaneya	Lvan	8
Pennsy ia	Penci eya	Pen	6
nsy ia	ci ey	A	2
nsy i	ci ey	(none)	0
Subtotal			16
Length of original strings			24
Simil = 16/24			0.67

III. SYSTEM DESIGN

A. Database Record Duplicate Detection Application

The software Database Record Duplicate Detection is an application program designed solely to interact with MySQL database server connecting to all database that may exists in the server database but limited to those that is not secured with a password. The application is designed on a single interface which consists of two tabs to cover the major activity of the designed app. The first tab is designed to operate on the duplicate detection and in the process generate a log in a text file format to store the operation performed by the application user during the duplication. On the other side of the tab is a front end table which shows the record stored in the database table to be optimized. While on the other tab, the application provides a platform to retrieve stored log for reference purpose.

Simil Algorithm takes effect while the button check duplicate is triggered. The algorithm uses the record comparison between the short range of 0.95 and absolute value of 1 on each field that a record has to the other succeeding record's field in the selected database for optimization. The optimization takes effect with the action to delete duplicate files if and only if user requests the optimization action. The architecture in figure 1 shows the summary of how the designed application operates with figure 2 showing the flowchart of the proposed Database Duplicate Detection Software.

The choice of programming language is Object Oriented Programming Language called Java. The development kit JDK and Wamp database server for MySQL database development of the system was installed to store and retrieve data easily. These two Software tools are linked with an object to database connecting tools provided by every java NetBeans IDE known as the MySQL JDBC Driver for linking the Application Interfaces to MySQL Database.

B. Java Program

Java is an object-based programming language used to design both system and application software. The key reason behind the choice of java over all other programming language is that it is capable of executing on any system platform, and it filters out memory that is not used after building and compilation.

C. MySQL Database Server

Database is required for ease of storage, retrieval, and update of data items, generally referred to as a repository for data. There are several choices of databases but MySQL is chosen due to the quality way of data acquisition, its flexibility in querying of the database, and its non-selective connection to all computer object-oriented languages.

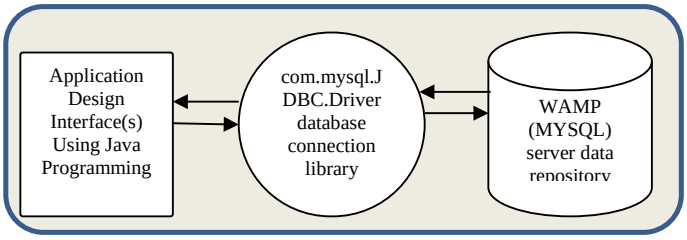


Figure 1: Design Tools Relationship Architecture

IV. SYSTEM IMPLEMENTATION

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. Hardware and Software Requirements

The minimum hardware requirement for deployment includes 512MHz or Higher Intel Premium or AMD Processor, 256Mb Memory (RAM), VGA 800 x 600, 256 color and Hard Disk Storage of 60 GB. Also the software minimum requirement is; 32 or 64bit Windows Operating Systems (OS) or any other OS that support the use of Java Runtime Library, reliable and licensed Antivirus software like Avast, AVG, or any system security shield, MySQL Database Testing (Xampp, Wamp, Lamp, and others) and Java Software Development Kit.

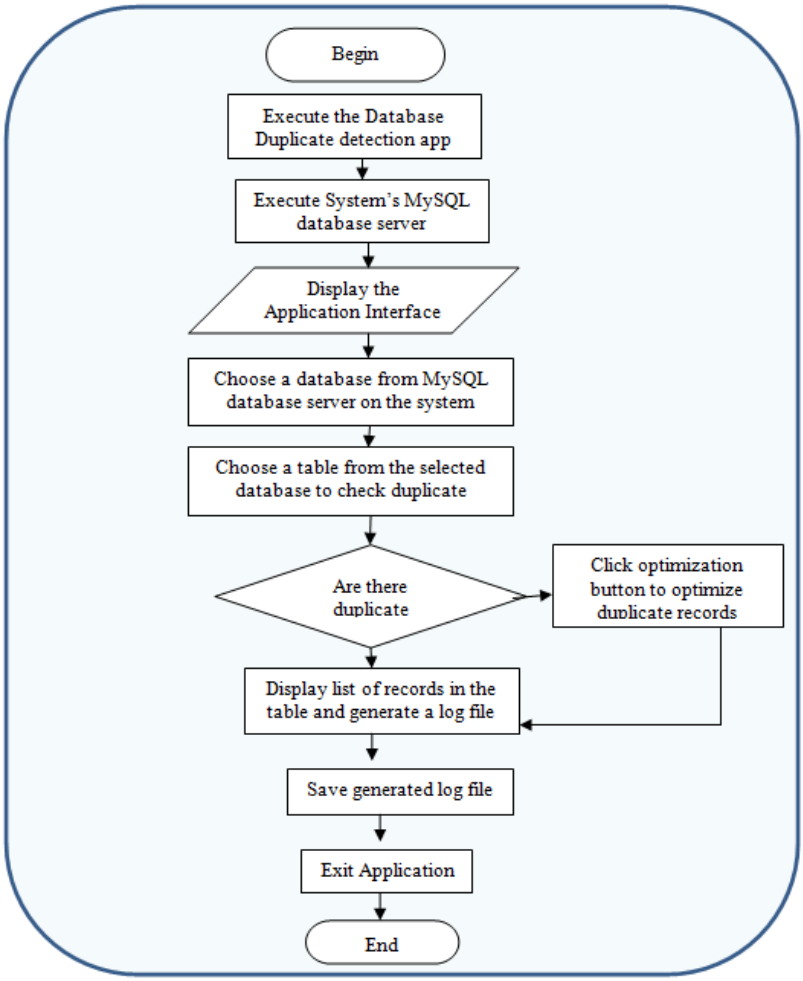
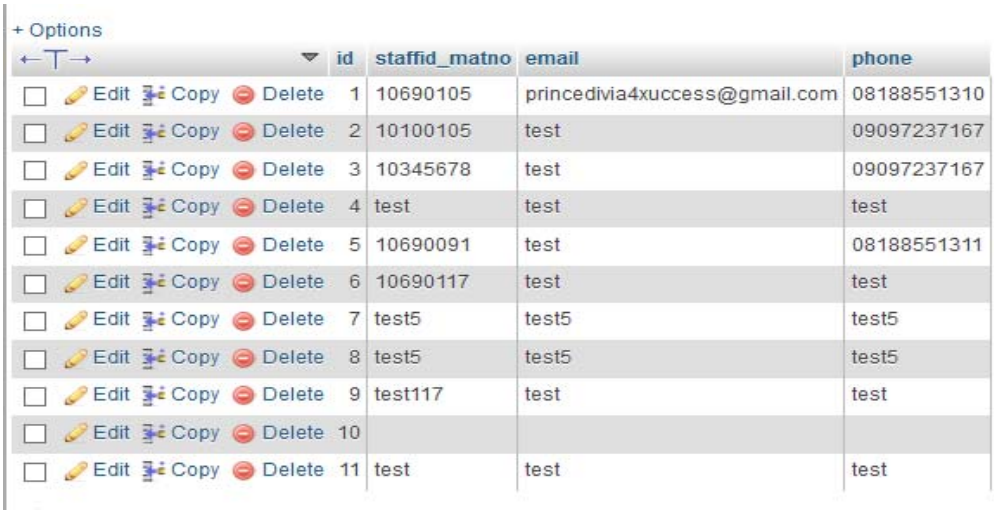


Figure 2: Proposed Database Duplicate detection Software Flowchart

B. Program Description

The application is designed to connect directly to the server and communicate with all existing databases that may reside on the server. Nevertheless, a table for a database called department.signup_tbl was tested on the software and the two screenshot below in figure 3 and figure 4 show the record of the table before and after the application testing.

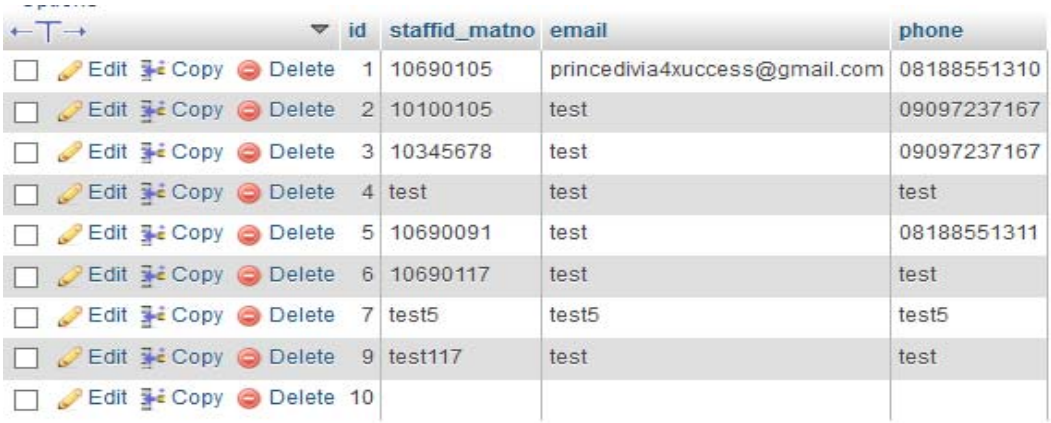
In using a matching simil algorithm, the search for duplicate records in the database table was done by checking the similarity match of each field that a record is composed of in the database using a close range of 0.8, 0.9, or 1.0 similarity of alike records under the similarity space of 0.0 and 1.0. At the end optimization of such duplicated records were achieved.



+ Options

	id	staffid_matno	email	phone
Edit Copy Delete	1	10690105	princedivia4xuccess@gmail.com	08188551310
Edit Copy Delete	2	10100105	test	09097237167
Edit Copy Delete	3	10345678	test	09097237167
Edit Copy Delete	4	test	test	test
Edit Copy Delete	5	10690091	test	08188551311
Edit Copy Delete	6	10690117	test	test
Edit Copy Delete	7	test5	test5	test5
Edit Copy Delete	8	test5	test5	test5
Edit Copy Delete	9	test117	test	test
Edit Copy Delete	10			
Edit Copy Delete	11	test	test	test

Figure 3: department.signup_tbl (record before optimization).



	id	staffid_matno	email	phone
Edit Copy Delete	1	10690105	princedivia4xuccess@gmail.com	08188551310
Edit Copy Delete	2	10100105	test	09097237167
Edit Copy Delete	3	10345678	test	09097237167
Edit Copy Delete	4	test	test	test
Edit Copy Delete	5	10690091	test	08188551311
Edit Copy Delete	6	10690117	test	test
Edit Copy Delete	7	test5	test5	test5
Edit Copy Delete	9	test117	test	test
Edit Copy Delete	10			

Figure 4: department.signup_tbl (record after optimization).

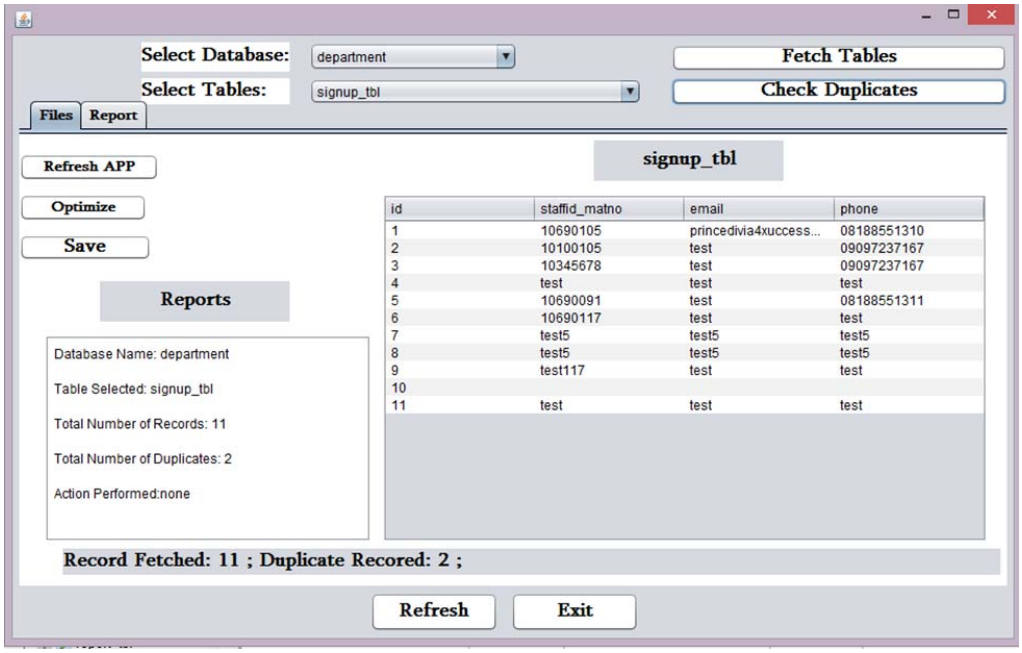


Figure 5: Database Record duplication detection screenshot 1 (Before optimization)

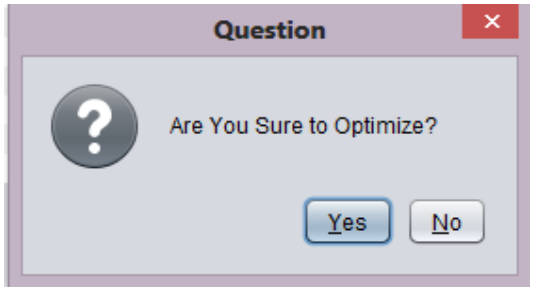


Figure 6: Database Record duplication detection screenshot 2

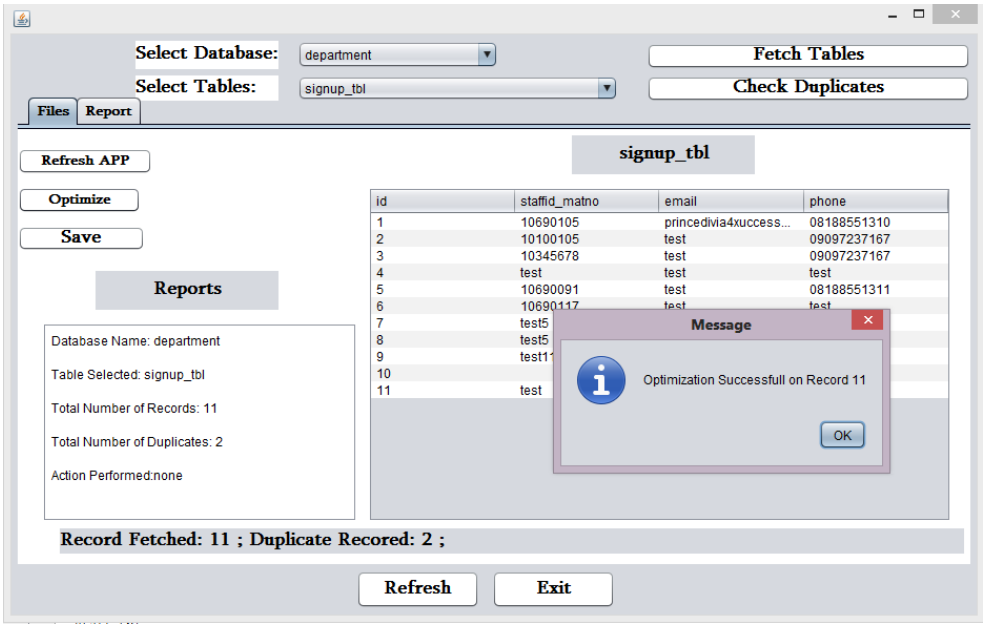


Figure 7: Database Record duplication detection screenshot 3 (After optimization)

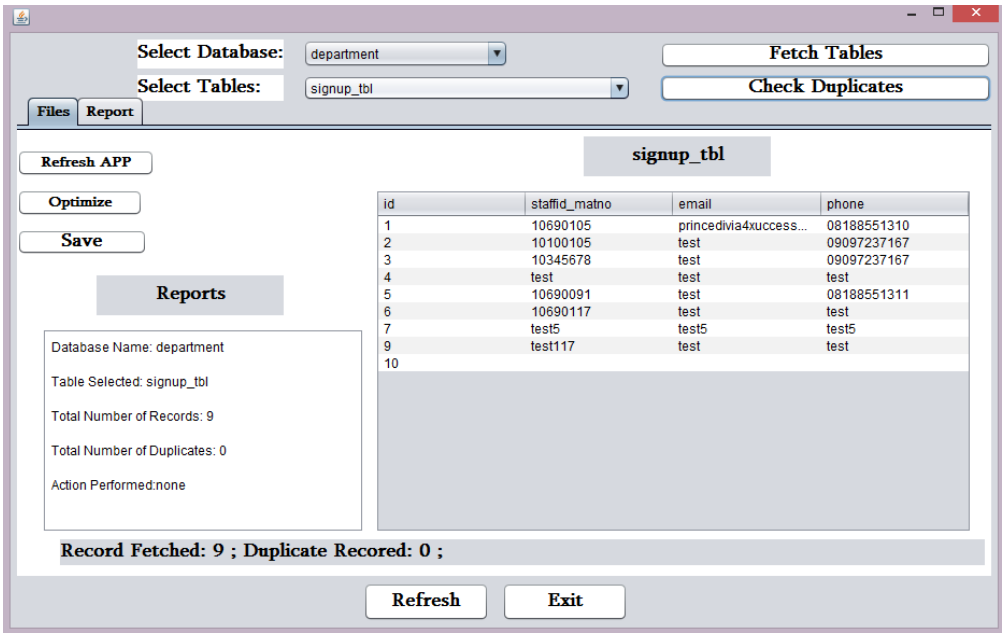


Figure 8: Database Record duplication detection screenshot 4 (After optimization)

V. CONCLUSION

This study implements database record duplicate detection system using simil matching algorithm. The application can achieve effective data reduction by exploiting relationships among similar blocks, rather than only among identical blocks, while improving the performance of computational and memory overheads. Data matching algorithm in general, has gone a long way supporting several areas of need, ranging from redundancy optimization, throughout the level of pattern verification, to the concentrated length of a diagnostic level and others, and it will be useful in the progressive buildup of information technology.

REFERENCES

[1] V.S. Tom, "An algorithm to look for similar strings", 2011. Retrieved 2 Nov, 2017 from: http://www.accessmvp.com/tomvanstiphout/simil.htm
[2] V.S. Verykios, A.K. Elmagarmid and E.N. Houstis, "Automating the approximate record matching process", Inform. Sci., . 2000, vol. pp. 126, 83-98.
[3] M. Weis, F. Naumann, U. Jehle, J. Lufter and H. Schuster, "Industry-scale duplicates detection", In Proc. International Conference on Very Large Databases, Auckland., 2008, vol. 1, pp. 1253-1264
[4] C. Weimin, "New algorithm for ordered tree-to-tree correction problem", Journal of Algorithm, 2001, 40, 135–158.
[5] A.K Elmagarmid, P.G. Ipeirotis and V.S. Verykios, " Duplicate record detection: A survey", IEEE Transaction on Knowledge and Data Engineering, 2006, vol. 19, pp. 1-16.
[6] H. Yang and J. Callan, "Near-Duplicate Detection by Instance-level Constrained Clustering", SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006, pp. 421-428.
[7] C.A. Giles, A.A. Brooks, T. Doszkocs and D. Hummel, "An experiment in computer-assisted duplicate checking", in Proceedings of the ASIS Annual Meeting, 1976, pp. 108.
[8] M.E. Menai, "Detection of Plagiarism in Arabic Documents", .I.J. Informtion Technology and Computer Science, 2012, vol. 10, pp. 80-89.
[9] A. Monge, "Matching algorithms within a duplicate detection system", IEEE Data Engineering Bulletin, 2000, vol. 23, pp. 14-20.
[10] T. Tanimoto, "An Elementary Mathematical theory of Classification and Prediction", Internal IBM Technical Report, 1958.
[11] P. Jaccard, "The distribution of the flora in the alpine zone", New Phytologist, 1912, vol. 11, pp.37–50.
[12] A.W. Goodman and W.M. Zaring, "Euclid's Algorithm and the LeastRemainder Algorithm", The American Mathematical Monthly, 1952, pp. 156-159.
[13] J.R. Goldman and L.H. Kauffman, "Rational Tangles", Advances in Applied Mathematics, 1996, pp.300-332.