ENHANCING PRIVACY ON BACK-END DATA THROUGH NEGATIVE DATABASE CREATION & SHUFFLING TECHNIQUE ON EAV MODEL

[1]Henna Bose, [2]Krishnachalitha K C
[1]Student, [2]Assistant Professor
[1]Computer Science,
[1]St.Joseph's College, Thrissur, India

Abstract : ***Security of data is one of the major issue in today's world. Real world databases have information that needs to be securely stored. Over the years, various encryption schemes have been developed in order to protect the database from various attacks by intruders. But still there are some privacy concerns on database. So proper security measures must be provided in order to keep back-end data more secure. In today's system, majority of data stored in database requires frequent alteration or modification. So we need to have a database which is flexible enough to support addition or alteration of new types of data without changing physical database schema. For that purpose, we use generic database in which data are represented in EAV model. In this paper, we are implementing extra layer of security to generic database using negative database creation and shuffling technique which keeps the Back-End data secure and prevent the third parties to access it.***

IndexTerms - EAV model, Negative database, Shuffling.

I. INTRODUCTION

Design of efficient database is a challenging task. Generic database handles most of the complex problems which helps to perform alterations on data without changing physical database schema of database. Generic database is based on EAV model in which all types of data can be stored in a single table without worrying about type of data to be stored [5]. Its application is mostly seen in hospital management systems in which the structure of table is not strictly defined and undergoes periodic changes and alterations. So for this purpose, we use generic database which handles it with ease. Many of them in today's world is aware only for securing Front-End data, but we need to be aware of securing back end data because most of sensitive data is stored on Back -End. So proper security measures must be implemented on database to keep those data secure .The approach of generating negative databases could solve such a problem. Negative database can be defined as a database that contains huge amount of data consisting of counterfeit data along with the real data [1]. For that purpose, we use generic database along with negative database creation and shuffling technique using hybrid approach to store the data securely on database which is difficult to trace or understand by the third parties. The system uses a framework to implement the concept of negative database along with shuffling technique on generic database to keep data stored in database to be secure. This framework consists of various algorithms which is used to manipulate, process and store data in database with some counterfeit information.

II. EXISTING SYSTEM

In the existing system, the data stored on generic database with negative database creation concept and encryption strategies only. It uses same key value to append negative values during negative database creation. Since data stored on generic database after all operation in same order of the data fields , there is a chance of possibility for the third party to recognize and extract the original data.

III. PROPOSED SYSTEM

The proposed system consists of shuffling technique on negative database using hybrid algorithm. The proposed Hybrid algorithm has two main advantages. The proposed method protects private data with no loss of information which makes usability of data, and also data is reconstructed [2]. It also consists of random generation of key values to append negative values during the creation of negative database which enhances the security. Thus the system creates and enhances the security of database which causes difficulty for the third parties to trace the contents of original data. Thus the concept of negative database along with shuffling technique helps to prevent data theft from malicious users and provide efficient data retrieval for all valid users. The system uses a framework which includes a set of algorithms that manipulate the input data and store it in database. The main objective is the validation of a benign user and rejection of a malicious user for a particular database[4].

IV. SYSTEM ARCHITECTURE

figure1. system architecture

The system architecture is shown in figure 1. It describes how to create a secure database. It uses the generic database for the implementation. The system uses generic database with negative database creation and shuffling technique. The process of shuffling uses a hybrid approach of storing data in EAV model. It considers first encrypting and then converting the sensitive data to negative database and then performs the shuffling technique using hybrid algorithm and then the result is stored in the database. First, the data's stored in database is converted into generic database. Then it performs segregation process which identifies the sensitive and non-sensitive data. The sensitive data is stored in database through various encryption strategies, negative database creation and shuffling approach and the resultant data is stored in EAV model. The non-sensitive data is directly stored on EAV model, as the cost will increase with the increase in sensitive fields of the original data.

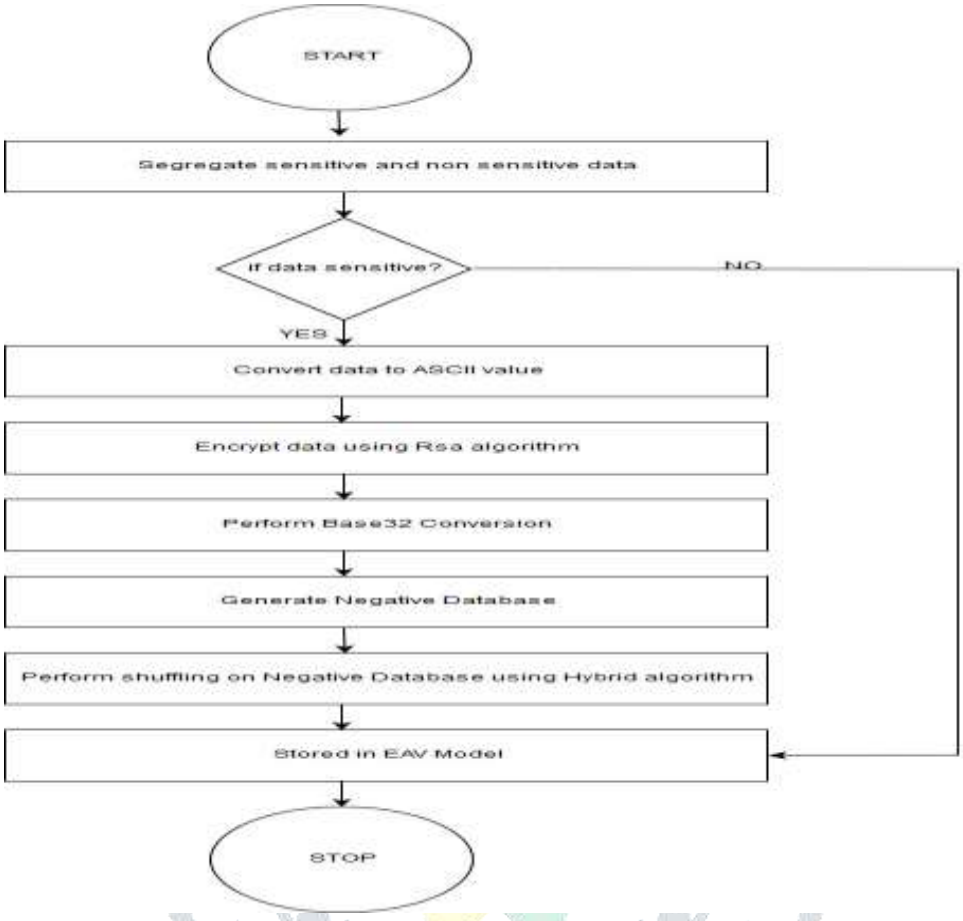

figure 2.flow chart of the system

The figure 2. describes the flowchart of the working of the system. It shows the sequence of operations that includes in the process of storage of data in generic database which enhances security on Back-End data. First, it includes a process of segregating sensitive and non-sensitive data. If data is non-sensitive, it can be directly stored on EAV model. If the data is sensitive, it undergoes various encryption strategies. First, the data is converted into corresponding ASCII value . Then the result is passed through the RSA encryption algorithm which generates a corresponding cipher text which is stored in form of byte array. The byte array values obtained from encryption strategies are processed through Base32 conversion which is used to reduce the length of obtained cipher value. Next the cipher values undergoes through the process of negative database creation and the resultant data values passed through the shuffling technique. The resultant values after shuffling is stored on EAV model which enhances privacy on data stored in database.

V. METHODOLOGY

This paper presents a security technique which uses the concept of negative database and shuffling to enhance level of security in generic databases. We first segregate sensitive and non-sensitive data. The same table is used for storing both sensitive and non-sensitive data. The non-sensitive data is directly stored in EAV model whereas the sensitive data is stored through process of encryption, negative database creation and shuffling technique .

table 1:structure of table for storing the sensitive data

ENTITY	ATTRIBUTE	VALUE1	VALUE2	VALUE3	VALUE4	VALUE5	VALUE6

- **Database Encryption**

In proposed system, first we will convert the sensitive data into ASCII value. Then the resultant value is passed to RSA encryption algorithm which generates the cipher text in form of the byte array. Next performs operation of converting the cipher text into BASE 32 conversion to reduce the length of cipher text and then the resultant data values is passed through the negative database creation process [1].

1. Let input=sensitive data. //sensitive data is given as input.
2. Let as[]=Toascii(input) //convert the sensitive values with corresponding Ascii value.
3. Encrypt[]=RSA_Public_Key(as) //apply RSA encryption algorithm.
4. Let cipher_text=double_hex(Encrypt) //convert resultant ciphertext into base32 to reduce length.

- **Negative Database Creation**

The cipher text is then inserted into the counterfeit data at some specific location. Now, the encrypted sensitive data along with some counterfeit data is stored in the database. If we store the data directly after encryption by RSA algorithm, we always have a risk to lose the public key. If that key is lost or made available to others, then our data can never be secured and will always be on risk [1].

1.Get the cipher_text input from previous section and pass it through break_data.
2.Segregation()
a. Assue a number "n" // the number of value fields in the modified EAV model.
b. Convert the cipher_text to the length that is multiple of "n" by appending zeros in the start.
c. Split the cipher_text in "n" groups, Each group as cipher_val[][] having n rows and (length/n) columns.
3.Negative_database_creation()
a. For each row of cipher_val[][] from (i:=0) to (i<n) is to be stored in a different value field of a modified EAV generic model.
i. Take a variable string Str="".
ii. Take an integer q=0.
iii. For each (q=0) to (q<(length/n))
1.gr =generated_chars(k) //Generate "k" random characters & key values which enhances security.
2.Str =str+gr+cipher_val[i][q] //used for concatenation.
iv. gr =generated_chars(k).
v. str =str+gr.

b. We get "n" encrypted strings with some negative data, where the useful data is only at position at multiple of "k+1" and remaining data is negative.

- **Shuffling Technique**

The shuffling technique uses hybrid algorithm which is applied on data that is created during negative database creation. This hybrid approach keeps back-end data more secure and efficient. First we randomizing the original data. Then we apply generalization on randomized or modified data. This technique protect private data with better accuracy, also it can reconstruct original data and provide data with no information loss, makes usability of data. Hybrid algorithm has two main advantages. The proposed method protects private data with no loss of information which makes usability of data, and also data is reconstructed [2].

Hybrid Algorithm

1) Inputs: Data stored obtained during negative database creation, Transition probability matrix P, Mapping Matrix M with size 1*j between T and P
2) Output: Derived table.
3) Method:
a) Select the attribute from table T.
b) Generate probability matrix P randomly with size j*j.
c) Generate Mapping matrix M randomly
d) Assign each P (P1, P2… Pj) to the column of T (T1, T2… Tj) randomly by M.
e) Rearrange element of T with respect to highest value of P location. If P location is already used, go for next highest location, if value of the P of two or more location is same then choose the left hand side value.
f) Recombine T matrix.
g) Re-substitute in table.
h) Apply K-anonymity on table based on selected attribute.
i) Generalize the value, if numeric takes the range of lowest and highest value.
j) Stop.

VI. IMPLEMENTATION

fig 3. admin login page

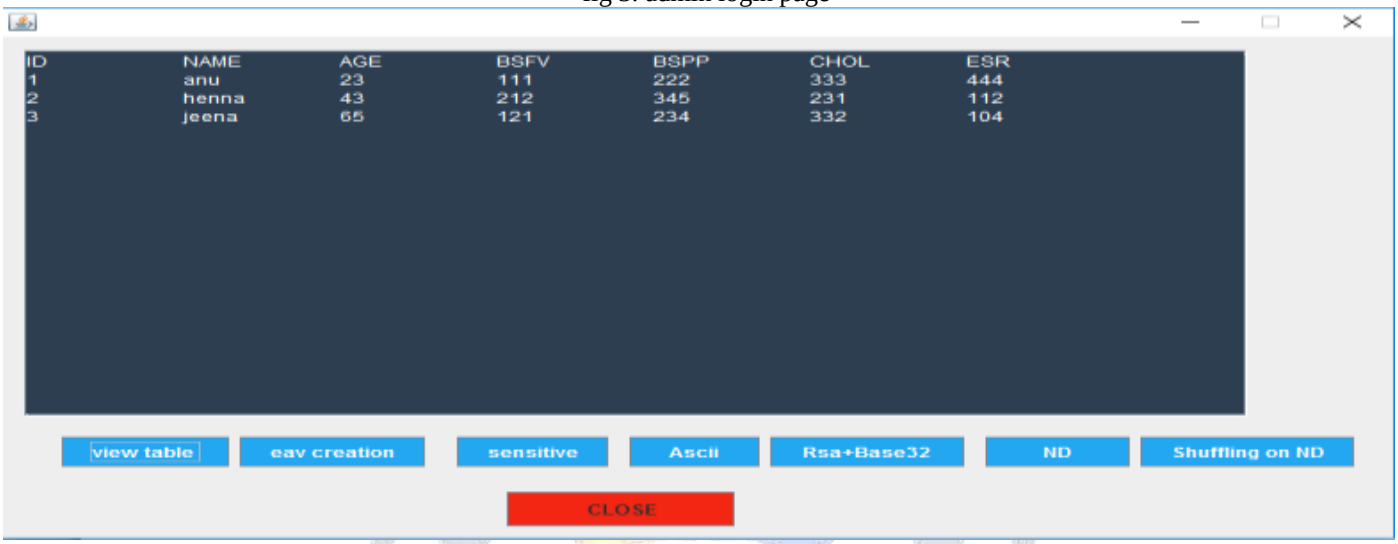

fig 4.admin views the stored data on database.

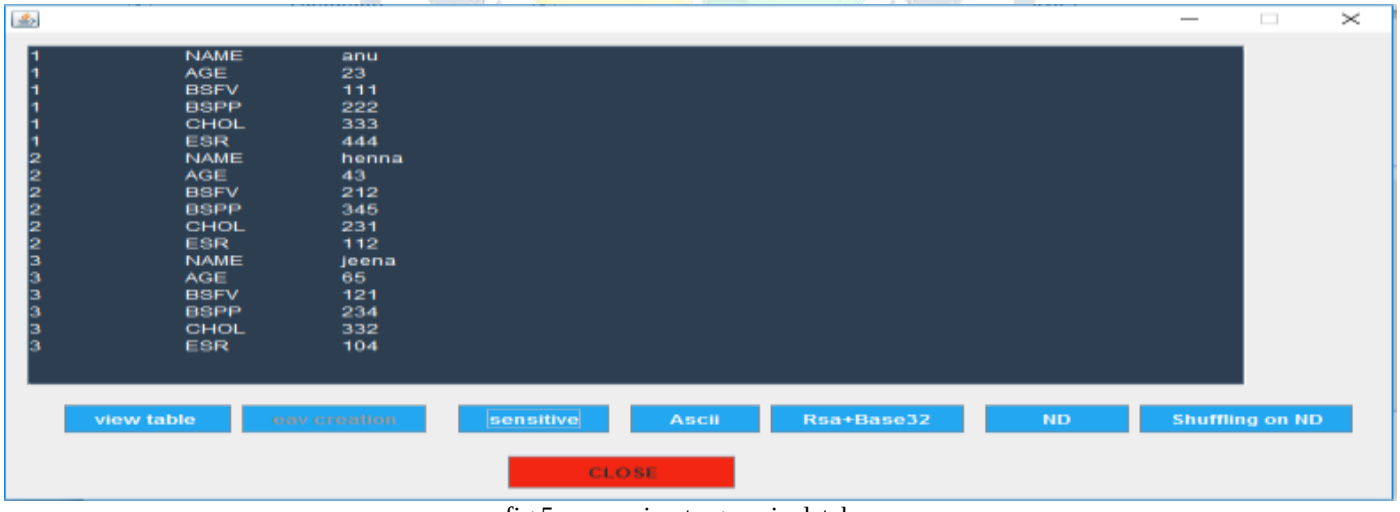

fig 5.conversion to generic database.

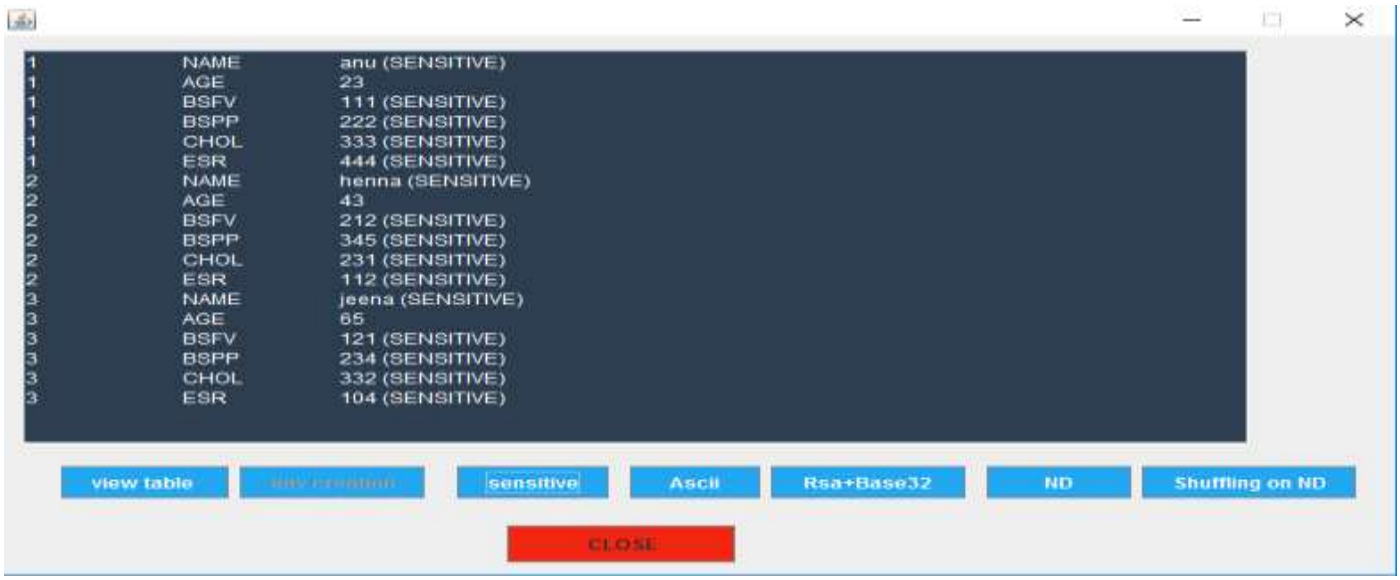

fig 6. segregate sensitive and non-sensitive data.

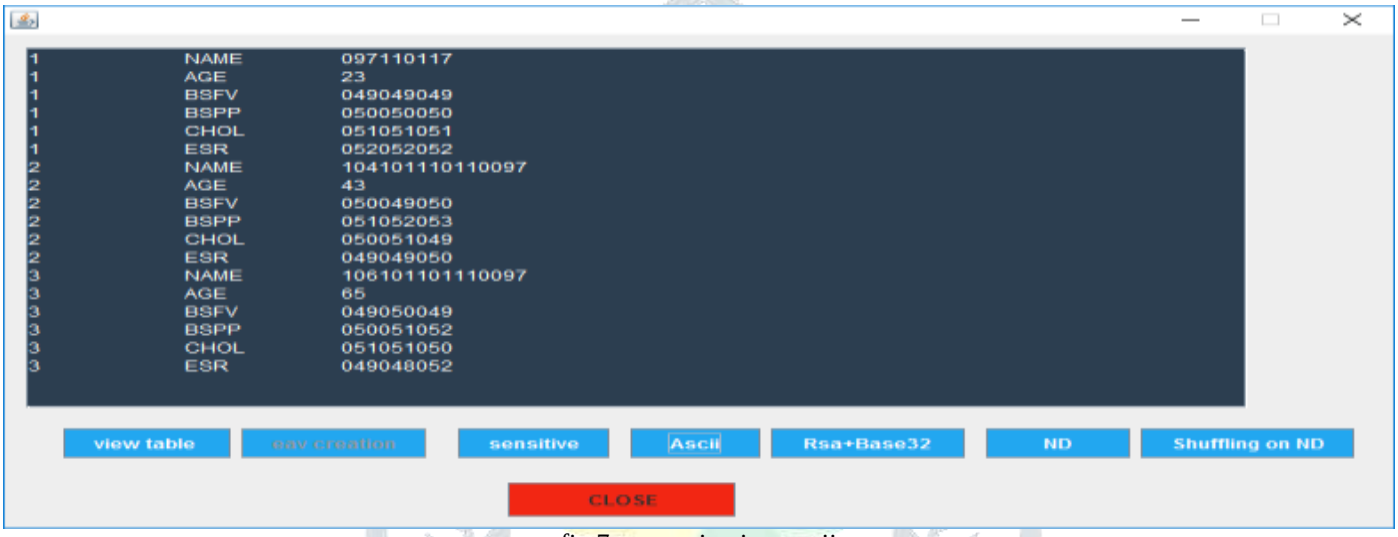

fig 7.conversion into ascii

fig 8. perform rsa algorithm and base32 conversion

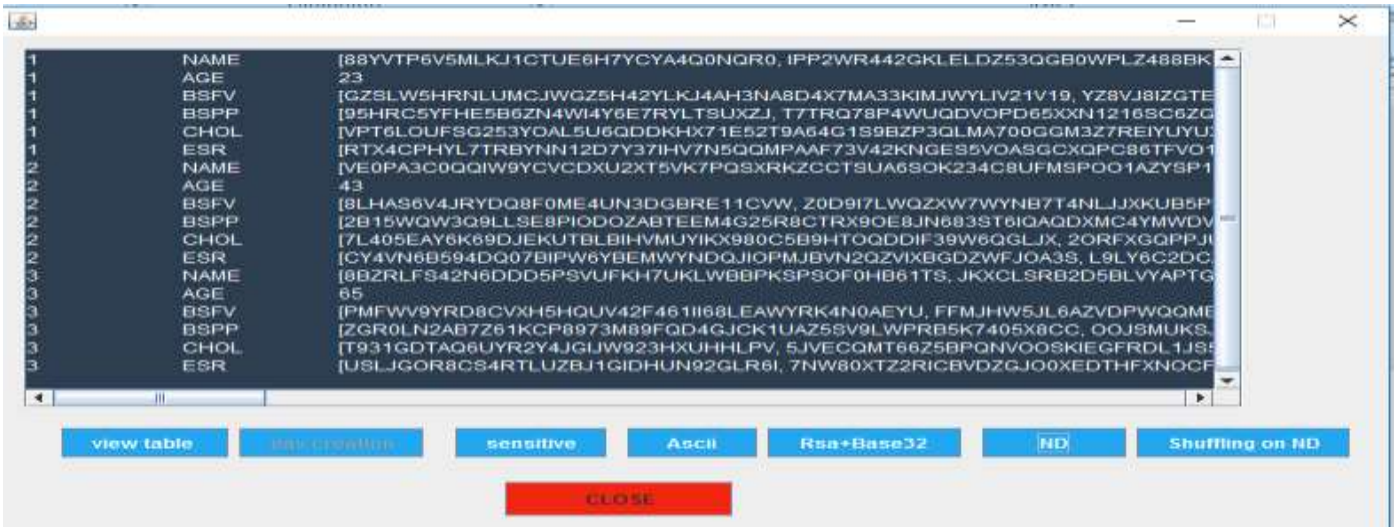

fig 9. creates negative database

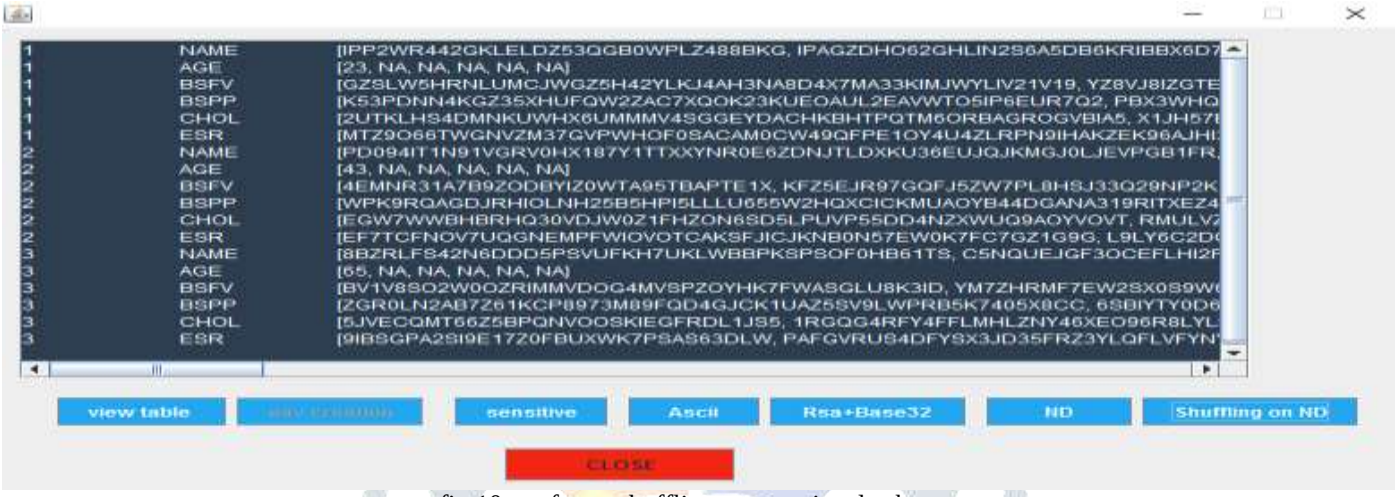

fig 10. performs shuffling on negative database .

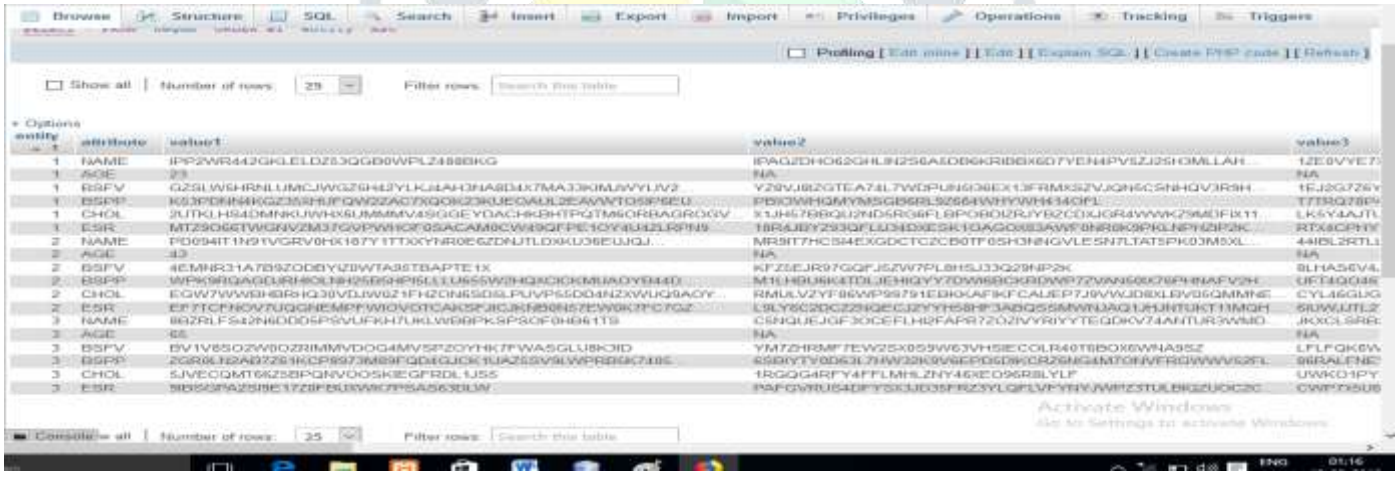

fig 11. stores the shuffled negative data to database.

VII. CONCLUSION

The proposed method implements the shuffling and negative database creation on generic database which enhances the privacy on database. It makes sensitive data more secure. The proposed system can be used to protect confidential data which can be widely used in field of banking, hospitality etc. The proposed Hybrid approach employs randomization and k-anonymity method. By using shuffling technique attacker cannot identify a pattern of data. The proposed method combines k-anonymity with randomization technique thus it is difficult for the attacker to identify homogeneity and background attack.it makes the private data to be stored in database more secure with better accuracy and gives no loss of information which make usability of data ,and makes the data to be reconstructed. In future, the proposed work can be applied on NoSQL database.

VIII. ACKNOWLEDGEMENT

First of all, I am grateful to the Almighty God for establishing me to complete this project. I am especially thankful to my guide, Ms. Krishnachalitha K C , and all other faculty members from the Department of Computer Science and my friends, for giving me their sole co-operation, support and encouragement in the preparation of this report. Finally I express my thanks to Project instructor, colleagues and my dear parents for giving me valuable advice and support throughout my project work.

REFERENCES

[1] Gaurav Dubey, Vikram Khurana, Shelly Sachdeva. "Implementing Security Technique on Generic Database" .

[2] Savita Lohiya, Lata Ragha. "Privacy Preserving in Data Mining Using Hybrid Approach" 2012 Fourth International Conference.

[3] Anup Patel, Niveeta Sharma, Magdalini Eirinaki. "Negative Database for Data Security" .ICC '09 Proceedings of the 2009 International Conference on Computing, Engineering and Information .

[4] Fernando Esponda, Stephanie Forrest and Paul Helman. " Enhancing privacy through Negative representation of data" IEEE,2002 conference.

[5] Iqra Basharat, Farooque Azam, Abdul Wahab Muzaffar. "Database Security and Encryption" .International Journal of Computer Applications. Vol. 47(12), June 2012pp. 975 – 888.