

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/321704020>

# Proposed Algorithm to Query on Negative Database with Positive Database

**Article** in International Journal of Computer Science and Information Security, · October 2017

CITATIONS

0

READS

33

**4 authors**, including:



**Sayed Abdelgaber**

Helwan University

**16** PUBLICATIONS **5** CITATIONS

[SEE PROFILE](#)



**Hanan Fahmy**

Helwan University

**2** PUBLICATIONS **0** CITATIONS

[SEE PROFILE](#)

**Some of the authors of this publication are also working on these related projects:**



A Roadmap to Implement EHR Nationwide in Egypt [View project](#)



Big Data Hadoop in Banking Sector [View project](#)

# Proposed Algorithm to Query on Negative Database with Positive Database

Sayed AbdelGaber<sup>1</sup>, Azza Ahmed Nail<sup>2</sup>, Hanan Fahmy<sup>1</sup>, Mohamed Ibrahim<sup>2</sup>

<sup>1</sup>Faculty of Computer Sciences and Information, Helwan University, Helwan – Egypt

<sup>1</sup>Faculty of Computer Sciences and Information, Fayoum University, Fayoum – Egypt

**Abstract-**The negative database (NDB) is a representation of all elements not found in a positive database (DB) to secure data from malicious users. Therefore, it is important to study the concept of negative databases and their algorithms, such as prefix, randomize, On-line, hybrid-Prefix, and real value algorithm, also interested in retrieving the positive data as a result of the query on the negative data. This research proposed a hybrid algorithm for converting from NDB to DB. The main idea of the proposed algorithm is to convert the positive query directed to the NDB and to obtain the result of the query in a secure way as if this query were on DB.

## I. INTRODUCTION

Protecting information stored in a database from inspection by unauthorized parties has been a concern since the start of this domain. Nowadays the user uses different types of encryption to protect the database, such as cryptographic protection of databases, multi-party computation schemes, the use of one-way functions, and dynamic accumulators [4]. However, as Esponda(2004) mentioned the encryption and decryption are not suitable for some applications [1]. Esponda proposed the NDB, which inspired by the biological immune system as a novel security algorithm [2][3][5].

Using the concept of NDB to secure the data is an optimal solution. However, when querying data, the user should use negative representation for the key search data [8]. Also, the results of query will be in negative representation [12]. Consequently, the user will need to convert these results to positive representation. This is break the security of database. This research work on this problem to find a solution to let the user work with the positive data query with the NDB and getting the result also with the positive data.

The big problem is the query on NDB, data security needed to maintain its confidentiality and satisfy the user requests from the original database.

The research organized into some sections, which are: Introduction, Query on Negative Database, the proposed Algorithm to query on NDB, Implementation to prove the correctness of the Proposed Algorithm and Conclusions.

## II. QUERY ON NEGATIVE DATABASE

The previous works extended that work by defining a set of relational operators for negative representations. For each relational operator, the corresponding negative operator is defined such that the result of the negative operator applied to a negative representation is equivalent to the positive version applied to the positive representation. Algorithms for each relational operator are described and compared to its positive counterpart. This work enhances the practicality of negative databases and expands their range of application [7].

All previous works apply the query to the positive data by using relational algebra and extract the positive result. Then, apply the same step to the negative database and extract the result of negative query. Matching the positive results with the negative results and make sure that the negative result

does not contain any element of positive result. There is no algorithm to convert the result of the query on NDB into a similar query result on DB. So far all that has been mentioned in the previous research is only the conversion of the query on NDB instead of DB [10].

### III. THE PROPOSED ALGORITHM TO QUERY ON NDB

The main idea of the proposed algorithm is to obtain the results of the positive query directed to NDB in a positive representation in a secure way. Fig .1 represents the conceptual using procedures of the proposed algorithm.

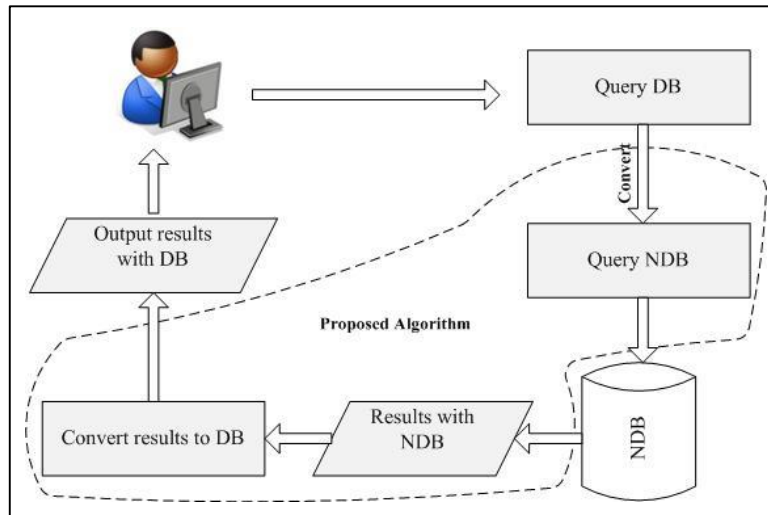


Figure1. The Conceptual using Procedures of the proposed Algorithm

#### A. The Proposed Algorithm Steps

Step1:

Get query from the authenticated user on NDB with positive data representation [9][11].

Step2:

Convert data query from positive to negative representation using Relational Algebra operations such as Select( $\sigma$ ), Projection( $\pi$ ), Intersection ( $\cap$ ) and Union( $\cup$ ) .(F.Esponda et al.,2007; F.Esponda,2008) Showed the detailed steps to convert the queries structure from positive representation to negative representation using relation algebra[7] [10].

Step3:

Retrieve all records satisfy the requested query from NDB, with negative representation.

Step4:

Calculate minimum value (Min) of each record with substitute all \* with 0. This will give the expected smallest values in binary representation as example:

The (Min) value of  $11^{**} = 1100$

Step5:

Calculate maximum value (Max) of each record with substitute all \* with 1. This will give the expected largest values in binary representation as example:

The (Max) value of  $11^{**} = 1111$

Step6:

Convert Minimum and Maximum values from the binary to decimal representation as example:

- 1) The (Min) value of  $11^{**} = (1100)_2 = (12)_{10}$

2) The (Max) value of  $11^{**} = (1111)_2 = (15)_{10}$

Step7:

Calculate MergeSort(Min, Max) and put them in negative list (NL) depend on step6.

MergeSort(Min, Max): is a Function has two arguments is MIN, MAX. The job of function, firstly adding the values of MIN, MAX to NL and secondly sorting MIN, MAX from smallest to the largest values.

Step8:

Find the hidden elements in NL, where these elements will represent the positive values which required and put in positive list (PL).

Step9:

Convert element in (PL) from decimal to string format to get query results with positive data representation.

Fig.2 shows the sequence of steps of the proposed algorithm.

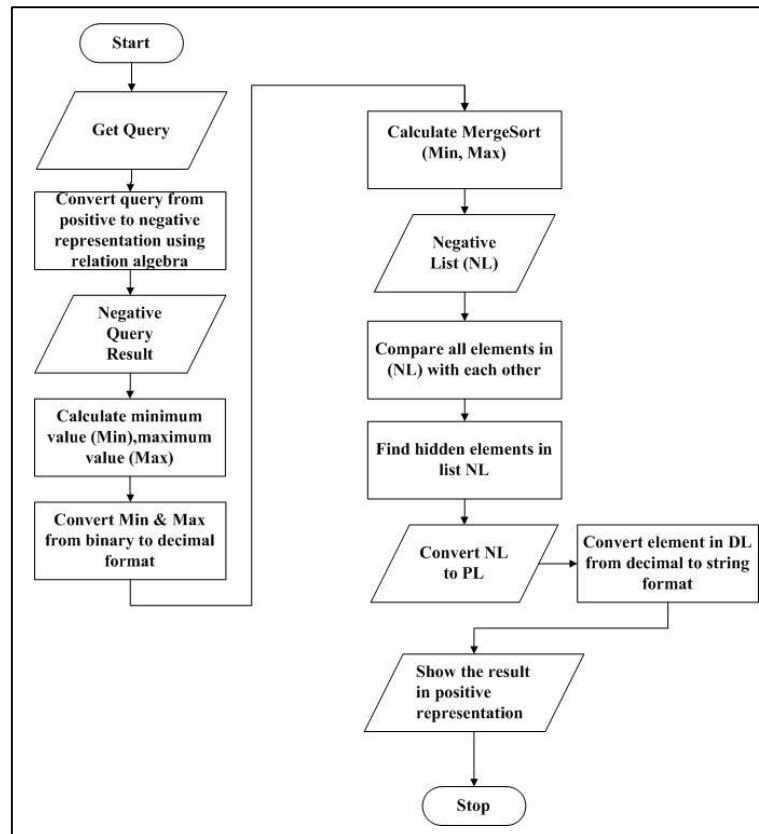


Figure2.The Proposed Algorithm Flowchart

#### IV. IMPLEMENTATION TO PROVE THE CORRECTNESS OF THE PROPOSED ALGORITHM

In this section, an example will be proposed to positive DB. Convert the positive DB to NDB and process the same queries using the proposed algorithm to get results in negative representation then convert the results to positive representation. The last step is comparing the results from the first attempt on positive DB with the results of the second attempt on negative NDB; if they are equal this will prove the correctness of the proposed algorithm.

1) Assume the follow positive DB as DB1

TABLE I  
POSITIVE DATABASE (DB1) OF STUDENT\_NAME1

Student_Name1	Course_code
Ali	CS1
Aza	IS2
Omr	CS3

- 2) Convert Student\_Name1 column in DB1 from string to binary representation is shown in the following table

TABLE II  
BINARY REPRESENTATION OF STUDENT\_NAME1

Student_Name1
0100 0001 0110 1100 0110 1001
0100 0001 0111 1010 0110 0001
0100 1111 0110 1101 0111 0010

- 3) Convert Student\_Name1 in DB1 from binary to decimal representation to calculate the complement of data is shown in the following table

TABLE III  
DECIMAL FORMAT OF STUDENT\_NAME1

Student_Name1
4287593
4291169
5205362

- 4) The complement of Student\_Name1 is shown in the following table

TABLE IV  
THE COMPLEMENT OF STUDENT\_NAME1

Student_Name1 (0-16777215)	
Start	End
0	4287592
4287594	4291168
4291170	5205361
5205363	16777215

The Steps of the Complement Calculations:

- 1) Calculate the complement (U- Student\_Name1) of pervious table to get the negative database.
- 2) Number of all combinations (U) =  $2^n$  where n is number of bits in record.
- 3) Number of all combinations in Student\_Name1 =  $2^{24} = 16777215$  record.
- 4) The complement of data (U- Student\_Name1) =  $16777215 - 3 = 16777212$  record.
- 5) The all of following steps to generate negative representation with using table (3)

Step1:

- 1) The start of the first set of records for negative representation Start=0
- 2) The end of the first set of records for negative representation End=the First value record in Student\_Name1-1=4287593-1=4287592

Step2:

- 1) The start of the second set of records for negative representation Start= the First value record in Student\_Name1 +1= 4287593+1=4287594
- 2) The end of the second set of records for Student\_Name1End=the Second value record in Student\_Name1-1=4291169-1=4291168

Step3:

- 1) The start of the third set of records for negative representation Start= Second value record in Student\_Name1+1=4291169+1=4291170

- 2) The end of the third set of records for negative representation End= Third value record in Student\_Name1-1=5205362-1=5205361

Step4:

- 1) The start of the fourth set of records for negative representation Start= the third value record in Student\_Name1+1= 5205362+1=5205363
- 2) The end of the fourth set of records for negative representation End=the fourth value record in Student\_Name1-1=5925230-1=5925229

Step5:

- 1) The start of the last set of records for negative representation Start= fourth value record in Student\_Name1+1= 5925230+1=5925231
  - 2) The end of the last set of records for negative representation End= 16777215
- 5) The negative representation of Student\_Name1 is shown in the following table using table V

TABLE V  
THE NEGATIVE DATABASE OF STUDENT NAME1

Studentname1 (NDB1)
00****
0100 0000 ****
0100 0001 00****
0100 0001 010****
0100 0001 0110 0****
0100 0001 0110 10****
0100 0001 0110 1100 00****
0100 0001 0110 1100 010****
0100 0001 0110 1100 0110 0****
0100 0001 0110 1100 0110 1000
0100 0001 0110 1100 0110 101*
0100 0001 0110 1100 0110 11**
0100 0001 0110 1100 0111 ****
0100 0001 0110 1100 1****
0100 0001 0110 1101 ****
0100 0001 0110 111* ****
0100 0001 0111 0****
0100 0001 0111 100****
0100 0001 0111 1010 00****
0100 0001 0111 1010 0101 ****
0100 0001 0111 1010 0110 0000
0100 0001 0111 1010 0110 001*
0100 0001 0111 1010 0110 01**
0100 0001 0111 1010 0110 1****
0100 0001 0111 1010 0111 ****
0100 0001 0111 1010 1****
0100 0001 0111 1011 ****
0100 0001 0111 11** ****
0100 0001 1****
0100 0010 ****
0100 0011 ****
0100 01****
0100 10****
0100 110****
0100 1110 ****
0100 1111 00****
0100 1111 0100 ****
0100 1111 0101 ****
0100 1111 0110 0****
0100 1111 0110 10****
0100 1111 0110 1100 ****
0100 1111 0110 1101 00****
0100 1111 0110 1101 010****
0100 1111 0110 1101 0110 ****
0100 1111 0110 1101 0111 000*
0100 1111 0110 1101 0111 0011
0100 1111 0110 1101 0111 01**

0100 1111 0110 1101 0111 1***
0100 1111 0110 1101 1*** ****
0100 1111 0110 111* **** ****
0100 1111 0111 **** **** ****
0100 1111 1*** **** **** ****
0101 **** **** **** **** ****
011* **** **** **** **** ****
10* **** **** **** **** ****
11* **** **** **** **** ****

1) Assume the following table as DB2

TABLE VI  
POSITIVE DATABASE (DB2) OF STUDENT\_NAME2

Student_Name2	Student_Grade
Ali	A
Omr	C

2) Convert Student\_Name2 attribute in DB2 from string to binary representation

TABLE VII  
BINARY FORMAT OF STUDENT NAME2

Student_Name2
01000001 01101100 01101001
01001111 011011010 1110010

3) Convert Student\_Name2 in DB2 from binary to decimal representation to calculate the complement of data is shown in the following table

TABLE VII  
DECIMAL FORMAT OF STUDENT\_NAME2

Student_Name2
4287593
5205362

4) The complement of Student\_Name2 is shown in the following table

TABLE VIII  
THE COMPLEMENT OF STUDENT\_NAME2

Student_Name2 (0-16777215)	
Start	End
0	4287592
4287594	5205361
5205363	16777215

The Steps of the Complement Calculations:

- 1) Calculate the complement (U- Student\_Name1) of pervious table to get the negative database.
- 2) Number of all combinations (U) = $2^n$  where n is number of bits in record.
- 3) Number of all combinations in Student\_Name1= $2^{24}$  =16777215 record.
- 4) The complement of data (U- Student\_Name1) = 16777215-3=16777212 record.
- 5) The all of following steps to generate negative representation with using table (3)

Step1:

- 1) The start of the first set of records for negative representation Start=0
- 2) The end of the first set of records for negative representation End=the First value record in Student\_Name1-1=4287593-1=4287592

Step2:

- 1) The start of the second set of records for negative representation Start= the First value record in Student\_Name1 +1= 4287593+1=4287594

- 2) The end of the second set of records for Student\_Name1 End=the Second value record in Student\_Name1-1=5205362-1=5205361

Step5:

- 1) The start of the last set of records for negative representation Start= fourth value record in Student\_Name1+1= 5205362+1=5205363
- 2) The end of the last set of records for negative representation End= 16777215
- 5) The negative representation of Student\_Name2 is shown in the following table using table IX

TABLE IX  
THE NEGATIVE DATABASE OF STUDENT\_NAME2

Student_Name2(NDB2)
00****
0100 0000 ****
0100 0001 00****
0100 0001 010****
0100 0001 0110 0****
0100 0001 0110 10****
0100 0001 0110 1100 00****
0100 0001 0110 1100 010****
0100 0001 0110 1100 0110 0****
0100 0001 0110 1100 0110 1000
0100 0001 0110 1100 0110 101*
0100 0001 0110 1100 0110 11**
0100 0001 0110 1100 0111 ****
0100 0001 0110 1100 1****
0100 0001 0110 1101 ****
0100 0001 0110 111* ****
0100 0001 0111 0****
0100 0001 0111 100****
0100 0001 0111 1010 00****
0100 0001 0111 1010 0101 ****
0100 0001 0111 1010 0110 00****
0100 0001 0111 1010 0110 01**
0100 0001 0111 1010 0110 1****
0100 0001 0111 1010 0111 ****
0100 0001 0111 1010 1****
0100 0001 0111 1011 ****
0100 0001 0111 11****
0100 0001 1****
0100 0010 ****
0100 0011 ****
0100 01****
0100 10****
0100 110****
0100 1110 ****
0100 1111 00****
0100 1111 0100 ****
0100 1111 0101 ****
0100 1111 0110 0****
0100 1111 0110 10****
0100 1111 0110 1100 ****
0100 1111 0110 1101 00****
0100 1111 0110 1101 010****
0100 1111 0110 1101 0110 ****
0100 1111 0110 1101 0111 000*
0100 1111 0110 1101 0111 0011
0100 1111 0110 1101 0111 01**
0100 1111 0110 1101 0111 1****
0100 1111 0110 1101 1****
0100 1111 0110 111* ****
0100 1111 0111 ****
0100 1111 1****
0101 ****
011* ****
10****
11****

- 1) Positive Select



```
SELECT Student_Name1
From DB1
Where Student_Name1='Aza'
```

TABLE X  
THE POSITIVE DATABASE OF STUDENT\_NAME1 WITH 24- BITS STRINGS. AND THE RESULT OF APPLYING  
SELECT OPERATION

Student_Name1	Result Query in Positive
0100 0001 0110 1100 0110 1001	0100 0001 0111 1010 0110 0001
0100 0001 0111 1010 0110 0001	<b>L.H.S</b>
0100 1111 0110 1101 0111 0010	

## 2) Negative Select

TABLE XI  
THE RESULT OF NEGATIVE AND POSITIVE SELECT REPRESENTATION

Result of Negative Select	Min	Max	MergeSort(Min,Max)	Result in Positive Representation
0100 0001 0111 0*** **** *	4290560	4291071	(4290560,4291071, 4291072,	$(4291169)_{10}=(0100\ 0001\ 0111\ 1010\ 0110\ 0001)_2$ <b>R.H.S</b>
0100 0001 0111 100* **** *	4291072	4291135	4291135, 4291136, 4291167,	
0100 0001 0111 1010 00** ****	4291136	4291167	4291168, 4291170, 4291171,	
0100 0001 0111 1010 0110 0000	4291168	-	4291172, 4291175, 4291176,	
0100 0001 0111 1010 0110 001*	4291170	4291171	4291183, 4291184, 4291199,	
0100 0001 0111 1010 0110 01**	4291172	4291175	4291200, 4291327, 4291328,	
0100 0001 0111 1010 0110 1***	4291176	4291183	4291583, 4291584, 4292607)	
0100 0001 0111 1010 0111 ****	4291184	4291199		
0100 0001 0111 1010 1*** ****	4291200	4291327		
0100 0001 0111 1011 **** ****	4291328	4291583		
0100 0001 0111 11** **** ****	4291584	4292607		

Comparing between L.H.S and R.H.S, if they are equal .This prove the correctness of the proposed algorithm where processing select operation.

## 3) Positive Projection

```
SELECT Student_Name1
From DB1
Where Student_Name1='A'
```

TABLE XII  
THE POSITIVE DATABASE OF STUDENT\_NAME1 WITH 24- BITS STRINGS. AND THE RESULT OF APPLYING  
PROJECTION OPERATION

Student_Name1	Result Query in Positive
0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 1001
0100 0001 0111 1010 0110 0001	0100 0001 0111 1010 0110 0001
0100 1111 0110 1101 0111 0010	<b>L.H.S</b>

## 4) Negative Projection

TABLE XIII  
REVERSING THE RESULT OF DATA FROM NEGATIVE TO POSITIVE PROJECTION

Result of Negative Projection	Min	Max	MergeSort(Min,Max)	Result in Positive Representation
0100 0001 00** **** **** *	4259840	4276223	(4259840,4276223	$(4287593)_{10}=(0100\ 0001\ 0110\ 1100\ 0110\ 1001)_2$ $(4291169)_{10}=(0100\ 0001\ 0111\ 1010\ 0110\ 0001)_2$ <b>R.H.S</b>
0100 0001 010* **** **** *	4276224	4284415	,4276224,4284415	
0100 0001 0110 0*** **** *	4284416	4286463	,4284416,4286463	
0100 0001 0110 10** **** *	4286464	4287487	,4286464,4287487	
0100 0001 0110 1100 00** ****	4287488	4287551	,4287584,4287591	
0100 0001 0110 1100 010* ****	4287552	4287583	,4287592,4287594	
0100 0001 0110 1100 0110 0***	4287584	4287591	,4287595,4287596	
0100 0001 0110 1100 0110 1000	4287592	-	,4287599,4287600	
0100 0001 0110 1100 0110 101*	4287594	4287595	,4287615,4287616	
0100 0001 0110 1100 0110 11**	4287596	4287599	,4287743,4287744	
0100 0001 0110 1100 0111 ****	4287600	4287615	,4287999,4288000	
0100 0001 0110 1100 1*** ****	4287616	4287743	,4288511,4288512	
0100 0001 0110 1101 **** ****	4287744	4287999	,4290559,4290560	
0100 0001 0110 111* **** ****	4288000	4288511	,4291071,4291072	
0100 0001 0111 0*** **** *	4288512	4290559	,4291135,4291136	
0100 0001 0111 100* **** ****	4290560	4291071	,4291167,4291168	
0100 0001 0111 1010 00** ****	4291072	4291135	,4291170,4291171	
0100 0001 0111 1010 0101 ****	4291136	4291167	,4291172,4291175	

0100 0001 0111 1010 0110 0000	4291168	-	,4291176,4291183
0100 0001 0111 1010 0110 001*	4291170	4291171	,4291184,4291199
0100 0001 0111 1010 0110 01**	4291172	4291175	,4291200,4291327
0100 0001 0111 1010 0110 1***	4291176	4291183	,4291328,4291583
0100 0001 0111 1010 0111 ****	4291184	4291199	,4291584,4292607
0100 0001 0111 1010 1*** *****	4291200	4291327	,4292608,4325375)
0100 0001 0111 1011 ****	4291328	4291583	
0100 0001 0111 11** ****	4291584	4292607	
0100 0001 1*** ****	4292608	4325375	

Comparing between L.H.S and R.H.S, if they are equal .This prove the correctness of the proposed algorithm where processing Projection operation.

5) Positive Intersection

```
SELECT Student_Name1 from DB1
INTERSECT
SELECT Student_Name2 from DB2
```

TABLE XIV  
THE POSITIVE DATABASE OF STUDENT\_NAME1 WITH 24- BITS STRINGS. AND THE RESULT OF APPLYING INTERSECTION OPERATION

Student_Name1	Student_Name2	Result Query in Positive
0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 1001
0100 0001 0111 1010 0110 0001	0100 1111 01101101 0111 0010	0100 1111 0110 1101 0111 0010
0100 1111 0110 1101 0111 0010		<b>L.H.S</b>

6) Negative Intersection

TABLE XV  
THE RESULT OF NEGATIVE AND POSITIVE INTERSECTION REPRESENTATION

Result of Negative Intersection	Min	Max	MergeSort(Min,Max)	Result in Positive Representation
00** ****	0	4194303	(0,4194303,4194304	(4287593) <sub>10</sub> =(100 0001 0110 1100
0100 0000 ****	4194304	4259839	,4259839,4259840	0110 1001) <sub>2</sub>
0100 0001 00** ****	4259840	4276223	,4276223,4276224	(5205362) <sub>10</sub> =(100 1111 0110 1101
0100 0001 010* ****	4276224	4284415	,4284415	0111 0010) <sub>2</sub>
0100 0001 0110 0*** ****	4284416	4286463	,4284416,4286463	<b>R.H.S</b>
0100 0001 0110 10** ****	4286464	4287487	,4286464,4287487	
0100 0001 0110 1100 00** ****	4287488	4287551	,4287488,4287551	
0100 0001 0110 1100 010* ****	4287552	4287583	,4287552,4287583	
0100 0001 0110 1100 0110 0***	4287584	4287591	,4287584,4287591	
0100 0001 0110 1100 0110 1000	4287592	-	,4287592,4287594	
0100 0001 0110 1100 0110 101*	4287594	4287595	,4287595,4287596	
0100 0001 0110 1100 0110 11**	4287596	4287599	,4287599,4287600	
0100 0001 0110 1100 0111 ****	4287600	4287615	,4287615,4287616	
0100 0001 0110 1100 1*** ****	4287616	4287743	,4287743,4287744	
0100 0001 0110 1101 ****	4287744	4287999	,4287999,4288000	
0100 0001 0110 111* ****	4288000	4288511	,4288511,4288512	
0100 0001 0111 0*** ****	4288512	4290559	,4290559,4290560	
0100 0001 0111 100* ****	4290560	4291071	,4291071,4291072	
0100 0001 0111 1010 00** ****	4291072	4291135	,4291135,4291136	
0100 0001 0111 1010 0101 ****	4291136	4291167	,4291167,4291168	
0100 0001 0111 1010 0110 00**	4291168	4291171	,4291171,4291172	
0100 0001 0111 1010 0110 01**	4291172	4291175	,4291175,4291176	
0100 0001 0111 1010 0110 1***	4291176	4291183	,4291183,4291184	
0100 0001 0111 1010 0111 ****	4291184	4291199	,4291199,4291200	
0100 0001 0111 1010 1*** ****	4291200	4291327	,4291327,4291328	
0100 0001 0111 1011 ****	4291328	4291583	,4291583,4291584	
0100 0001 0111 11** ****	4291584	4292607	,4292607,4292608	
0100 0001 1*** ****	4292608	4325375	,4292608,4325375	
0100 0010 ****	4325376	4390911	,4325376,4390911	
0100 0011 ****	4390912	4456447	,4390912,4456447	
0100 01** ****	4456447	4718591	,4456447,4718590	
0100 10** ****	4718592	4980735	,4718592,4980735	
0100 110* ****	4980736	5111807	,4980736,5111807	
0100 1110 ****	5111808	5177343	,5111808,5177343	
0100 1111 00** ****	5177344	5193727	,5177344,5193727	
0100 1111 0100 ****	5193728	5197823	,5193728,5197823	
0100 1111 0101 ****	5197824	5201919	,5197824,5201919	
		5203967	,5201920,5203967	
		5204991	,5203968,5204991	

0100 1111 0110 0*** **** *	5201920	5203967	,5204992,5205247
0100 1111 0110 10** **** *	5203968	5204991	,5205248,5205311
0100 1111 0110 1100 **** *	5204992	5205247	,5205312,5205343
0100 1111 0110 1101 00** ****	5205248	5205311	,5205344,5205359
0100 1111 0110 1101 010* ****	5205312	5205343	,5205360,5205361
0100 1111 0110 1101 0110 ****	5205344	5205359	,5205363,5205364
0100 1111 0110 1101 0111 000*	5205360	5205361	,5205367,5205368
0100 1111 0110 1101 0111 0011	-	5205363	,5205375,5205376
0100 1111 0110 1101 0111 01**	5205364	5205367	,5205503,5205504
0100 1111 0110 1101 0111 1***	5205368	5205375	,5206015,5206017
0100 1111 0110 1101 1*** ****	5205376	5205503	,5210110,5210113
0100 1111 0110 111* **** *	5205504	5206015	,5242878
0100 1111 0111 **** **** *	5206016	5210111	,5242880,6291455
0100 1111 1*** **** **** *	5210112	5242879	,6291456,8388607
0101 **** **** **** **** *	5242880	6291455	,8388608,12582911
011* **** **** **** **** *	6291456	8388607	,12582912,16777215)
10** **** **** **** **** *	8388608	12582911	
11** **** **** **** **** *	12582912	16777215	

Comparing between L.H.S and R.H.S, if they are equal .This prove the correctness of the proposed algorithm where processing Intersection operation.

7) Positive Union

```
SELECT Student_Name1 from DB1
UNION
SELECT Student_Name2 from DB2
```

TABLE XVI  
THE POSITIVE DATABASE OF STUDENT \_NAME1 WITH 24- BITS STRINGS. AND THE RESULT OF APPLYING UNION OPERATION

Student_Name1	Student_Name2	Result Query in Positive
0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 1001
0100 0001 0111 1010 0110 0001	0100 1111 0110 1101 0111 0010	0100 0001 0111 1010 0110 0001
0100 1111 0110 1101 0111 0010		0100 1111 0110 1101 0111 0010
		<b>L.H.S</b>

8) Negative Union

TABLE XVII  
THE RESULT OF DATA NEGATIVE AND POSITIVE UNIONREPRESENTATION

Result of Negative Union	Min	Max	MergeSort(Min ,Max)	Result in Positive Representation
00** **** **** **** **** *	0	4194303	(0,4194303	(4287593) <sub>10</sub> =(0100 0001 0110 1100 0110 1001) <sub>2</sub>
0100 0000 **** **** **** **** *	4194304	4259839	,4194304,4259839	(4291169) <sub>10</sub> =(0100 0001 0111 1010 0110 0001) <sub>2</sub>
0100 0001 00** **** **** **** *	4259840	4276223	,4259840,4276223	(5205362) <sub>10</sub> =(0100 1111 0110 1101 0111 0010) <sub>2</sub>
0100 0001 010* **** **** **** *	4276224	4284415	,4276224,4284415	
0100 0001 0110 0*** **** **** *	4284416	4286463	,4284416,4286463	
0100 0001 0110 10** **** **** *	4286464	4287487	,4286464,4287487	
0100 0001 0110 1100 00** **** *	4287488	4287551	,4287488,4287551	
0100 0001 0110 1100 010* **** *	4287552	4287583	,4287552,4287583	
0100 0001 0110 1100 0110 0***	4287584	4287591	,4287584,4287591	
0100 0001 0110 1100 0110 1000	4287592	-	,4287592,4287594	
0100 0001 0110 1100 0110 101*	4287594	4287595	,4287595,4287596	
0100 0001 0110 1100 0110 11**	4287596	4287599	,4287599,4287600	
0100 0001 0110 1100 0111 ****	4287600	4287615	,4287615,4287616	
0100 0001 0110 1100 1*** ****	4287616	4287743	,4287743,4287744	
0100 0001 0110 1101 **** ****	4287744	4287999	,4287999,4288000	
0100 0001 0110 111* **** ****	4288000	4288511	,4288511,4288512	
0100 0001 0111 0*** **** **** *	4288512	4290559	,4290559,4290560	
0100 0001 0111 100* **** **** *	4290560	4291071	,4291071,4291072	
0100 0001 0111 1010 00** **** *	4291072	4291135	,4291135,4291136	
0100 0001 0111 1010 0101 ****	4291136	4291167	,4291167,4291168	
0100 0001 0111 1010 0110 0000	4291168	-	,4291172,4291175	
0100 0001 0111 1010 0110 001*	4291170	4291171	,4291170,4291171	
0100 0001 0111 1010 0110 01**	4291172	4291175	,4291175,4291177	
0100 0001 0111 1010 0110 1***	4291176	4291183	,4291200,4291327	
0100 0001 0111 1010 0111 ****	4291184	4291199	,4291328,4291583	
0100 0001 0111 1010 1*** ****	4291200	4291327	,4291584,4292607	
0100 0001 0111 1011 **** ****	4291328	4291583	,4292608,4325375	
0100 0001 0111 11** **** ****	4291584	4292607	,4325376,4390911	<b>R.H.S</b>

0100 0001 1*** **	4292608	4325375	,4390912,4456447
0100 0010 **** **	4325376	4390911	,4456447,4718590
0100 0011 **** **	4390912	4456447	,4718592,4980735
0100 01** **** **	4456447	4718591	,4980736,5111807
0100 10** **** **	4718592	4980735	,5111808,5177343
0100 110* **** **	4980736	5111807	,5177344,5193727
0100 1110 **** **	5111808	5177343	,5193729,5197822
0100 1111 00** **** **	5177344	5193727	,5197824,5201919
0100 1111 0100 **** **	5193728	5197823	,5201920,5203967
0100 1111 0101 **** **	5197824	5201919	,5203968,5204991
0100 1111 0110 0*** **	5201920	5203967	,5204992,5205247
0100 1111 0110 10** **** **	5203968	5204991	,5205248,5205311
0100 1111 0110 1100 **** **	5204992	5205247	,5205312,5205343
0100 1111 0110 1101 00** ****	5205248	5205311	,5205344,5205359
0100 1111 0110 1101 010* ****	5205312	5205343	,5205360,5205361
0100 1111 0110 1101 0110 ****	5205344	5205359	,5205363,5205364
0100 1111 0110 1101 0111 000*	5205360	5205361	,5205367,5205368
0100 1111 0110 1101 0111 0011	-	5205363	,5205375,5205376
0100 1111 0110 1101 0111 01**	5205364	5205367	,5205503,5205504
0100 1111 0110 1101 0111 1***	5205368	5205375	,5206015,5206017
0100 1111 0110 1101 1*** **	5205376	5205503	,5210110,5210113
0100 1111 0110 111* **** **	5205504	5206015	,5242878,5242880
0100 1111 0111 **** **	5206016	5210111	,6291455,6291456
0100 1111 1*** **** **	5210112	5242879	,8388607,8388608
0101 **** **	5242880	6291455	,12582911,12582912
011* **** **	6291456	8388607	,16777215)
10** **** **	8388608	12582911	
11** **** **	12582912	16777215	

Comparing between L.H.S and R.H.S, they are equal .This prove the correctness of the proposed algorithm where processing Union operation.

9) Positive Join

```
SELECT Student_Name1, Student_Grade
FROM DB1 JOIN DB2
ON DB1.Student_Name1 = DB2.Student_Name2
```

TABLE XVIII  
THE POSITIVE DATABASE OF STUDENT\_NAME1 WITH 24- BITS STRINGS. AND THE RESULT OF APPLYING JOIN OPERATION

Student_Name1	Student_Name2	Result Query in Positive
0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 10010100 0001
0100 0001 0111 1010 0110 0001	0100 1111 0110 1101 0111 0010	0100 1111 0110 1101 0111 0010 01000011
0100 1111 0110 1101 0111 0010		<b>L.H.S</b>

TABLE XIX  
THE COMPLEMENT OF STUDENT\_NAME2 AND STUDENT\_GRADE

Student_Name2 and Student_Grade (0-4,294,967,295)	
Start	End
0	1,097,623,872
1,097,623,874	1,332,572,738
1,332,572,740	4,294,967,295

TABLE XX  
THE NEGATIVE DATABASE OF STUDENT\_NAME2 AND STUDENT\_GRADE

Student_Name2 and Student_Grade(NDB3)
00** **** **
0100 0000 **** **
0100 0001 00** **** **
0100 0001 010* **** **
0100 0001 0110 0*** **** **
0100 0001 0110 10** **** **
0100 0001 0110 1100 00** **** **
0100 0001 0110 1100 010* **** **
0100 0001 0110 1100 0110 0***** **

0100 0001 0110 1100 0110 1000**** ****
0100 0001 0110 1100 0110 1001 00** ****
<b>0100 0001 0110 1100 0110 1001 0100 0000</b>
0100 0001 0110 1100 0110 1001 0100 001*
0100 0001 0110 1100 0110 1001 0100 01**
0100 0001 0110 1100 0110 1001 0100 1***
0100 0001 0110 1100 0110 1001 0101 ****
0100 0001 0110 1100 0110 1001 011* ****
0100 0001 0110 1100 0110 1001 1*** ****
0100 0001 0110 1100 0110 101**** * **
0100 0001 0110 1100 0110 11***** **
0100 0001 0110 1100 0111 ***** **
0100 0001 0110 1100 1*** ***** **
0100 0001 0110 1101 **** ***** **
0100 0001 0110 111 ***** ***** **
0100 0001 0111 **** ***** ***** **
0100 0001 1*** ***** ***** ***** **
0100 0010 **** ***** ***** ***** **
0100 0011 **** ***** ***** ***** **
0100 0100 **** ***** ***** ***** **
0100 0101 **** ***** ***** ***** **
0100 0110 **** ***** ***** ***** **
0100 0111 **** ***** ***** ***** **
0100 10***** ***** ***** ***** **
0100 110* **** ***** ***** ***** **
0100 1110 **** ***** ***** ***** **
0100 1111 000* ***** ***** ***** **
0100 1111 001* ***** ***** ***** **
0100 1111 010* ***** ***** ***** **
0100 1111 0110 0***** ***** ***** **
0100 1111 0110 10***** ***** ***** **
0100 1111 0110 1100***** ***** ***** **
0100 1111 0110 1101 00** ***** ***** **
0100 1111 0110 1101 010* **** ***** **
0100 1111 0110 1101 0110 **** ***** **
0100 1111 0110 1101 0111 0000 **** *****
0100 1111 0110 1101 0111 0001 **** *****
0100 1111 0110 1101 0111 0010 00** ****
0100 1111 0110 1101 0111 0010 0100 000*
<b>0100 1111 0110 1101 0111 0010 0100 0010</b>
0100 1111 0110 1101 0111 0010 0100 01**
0100 1111 0110 1101 0111 0010 0100 1***
0100 1111 0110 1101 0111 0010 0101 ****
0100 1111 0110 1101 0111 0010 011* ****
0100 1111 0110 1101 0111 0010 1*** ****
0100 1111 0110 1101 0111 0011 **** *****
0100 1111 0110 1101 0111 01***** **
0100 1111 0110 1101 0111 1*** ***** **
0100 1111 0110 1101 1*** ***** ***** **
0100 1111 0110 111* **** ***** ***** **
0100 1111 1*** ***** ***** ***** ***** **
0101 **** ***** ***** ***** ***** **
0110 **** ***** ***** ***** ***** **
0111 **** ***** ***** ***** ***** **
1**** ***** ***** ***** ***** ***** **

10) Negative join

TABLE XXI  
THE RESULT OF DATA NEGATIVE AND POSITIVE JOIN REPRESENTATION

Result of Negative Join	Min	Max	MergeSort(Min,Max)	Result in Positive Representation
00** **** ***** ***** ***** ***** **	0	1073741823	(0,1073741823,1073741824	(1097623873) <sub>10</sub>
0100 0000 **** ***** ***** ***** ***** **	1073741824	1090519039	,1090519039,1090519040	=(100 0001 0110 1100 0110 1001
0100 0001 00** **** ***** ***** ***** **	1090519040	1094713343	,1094713343,1094713344	0100 0001) <sub>2</sub>
0100 0001 010* **** ***** ***** ***** **	1094713344	1096810495	,1096810495,1096810496	(1332572793) <sub>10</sub>
0100 0001 0110 0*** **** ***** ***** ***** **	1096810496	1097334783	,1097334783,1097334784	=(100 1111 0110 1101 0111 0010
0100 0001 0110 10** **** ***** ***** ***** **	1097334784	1097596927	,1097596927,1097596928	0111 1001) <sub>2</sub>
0100 0001 0110 1100 00** ***** ***** **	1097596928	1097613311	,1097613311,1097613312	<b>R.H.S</b>
0100 0001 0110 1100 010* ***** ***** **	1097613312	1097621503	,1097621503,1097621504	
0100 0001 0110 1100 0110 0***** ***** **	1097621504	1097623551	,1097623551,1097623552	
0100 0001 0110 1100 0110 1000**** ***** **	1097623552	1097623807	,1097623807,1097623808	
0100 0001 0110 1100 0110 1001 00** ****	1097623808	1097623871	,1097623871,1097623872	

0100 0001 0110 1100 0110 1001 0100 0000	1097623872	-	,1097623874,1097623875
0100 0001 0110 1100 0110 1001 0100 001*	1097623874	1097623875	,1097623876,1097623879
0100 0001 0110 1100 0110 1001 0100 01**	1097623876	1097623879	,1097623880,1097623887
0100 0001 0110 1100 0110 1001 0100 1***	1097623880	1097623887	,1097623904,1097623935
0100 0001 0110 1100 0110 1001 0101 ****	1097623888	1097623903	,1097623936,1097624063
0100 0001 0110 1100 0110 1001 011* *****	1097623904	1097623935	,1097624064,1097624575
0100 0001 0110 1100 0110 1001 1*** *****	1097623936	1097624063	,1097624576,1097625599
0100 0001 0110 1100 0110 101***** ****	1097624064	1097624575	,1097625600,1097629695
0100 0001 0110 1100 0110 11***** ****	1097624576	1097625599	,1097629696,1097662463
0100 0001 0110 1100 0111 ***** ****	1097625600	1097629695	,1097662464,1097727999
0100 0001 0110 1100 1***** *****	1097629696	1097662463	,1097728000,,1097859071
0100 0001 0110 1101 ***** *****	1097662464	1097727999	,1097859072,1098907647
0100 0001 0110 111***** *****	1097728000	1097859071	,1098907648,1107296255
0100 0001 0111 ***** *****	1097859072	1098907647	,1107296256,1124073471
0100 0001 1***** *****	1098907648	1107296255	,1124073472,1140850687
0100 0010 ***** *****	1107296256	1124073471	,1140850688,1157627903
0100 0011 ***** *****	1124073472	1140850687	,1157627904,1174405119
0100 0100 ***** *****	1140850688	1157627903	,1174405120,1191182335
0100 0101 ***** *****	1157627904	1174405119	,1191182336,1207959551
0100 0110 ***** *****	1174405120	1191182335	,1207959552,1275068415
0100 0111 ***** *****	1191182336	1207959551	,1275068416,1308622847
0100 10***** *****	1207959552	1275068415	,1308622848,1325400063
0100 110* ***** *****	1275068416	1308622847	,1325400064,1327497215
0100 1110 ***** *****	1308622848	1325400063	,1327497216,1329594367
0100 1111 000* ***** *****	1325400064	1327497215	,1329594368,1331691519
0100 1111 001* ***** *****	1327497216	1329594367	,1331691520,1332215807
0100 1111 010* ***** *****	1329594368	1331691519	,1332215808,1332477951
0100 1111 0110 0***** *****	1331691520	1332215807	,1332477952,1332543487
0100 1111 0110 10***** *****	1332215808	1332477951	,1332543488,1332559871
0100 1111 0110 1100***** *****	1332477952	1332543487	,1332559872,1332568063
0100 1111 0110 1101 00** *****	1332543488	1332559871	,1332568064,1332572159
0100 1111 0110 1101 010* *****	1332559872	1332568063	,1332572160,1332572415
0100 1111 0110 1101 0110 *****	1332568064	1332572159	,1332572416,1332572671
0100 1111 0110 1101 0111 0000 *****	1332572160	1332572415	,1332572672,1332572735
0100 1111 0110 1101 0111 0001 *****	1332572416	1332572671	,1332572736,1332572737
0100 1111 0110 1101 0111 0010 00** *****	1332572672	1332572735	,1332572738,1332572740
0100 1111 0110 1101 0111 0010 0100 000*	1332572736	1332572737	,1332572743,1332572744
0100 1111 0110 1101 0111 0010 0100 0010	1332572738	-	,1332572751,1332572752
0100 1111 0110 1101 0111 0010 0100 01**	1332572740	1332572743	,1332572767,1332572768
0100 1111 0110 1101 0111 0010 0100 1***	1332572744	1332572751	,1332572799,1332572800
0100 1111 0110 1101 0111 0010 0101 ****	1332572752	1332572767	,1332572927,1332572928
0100 1111 0110 1101 0111 0010 011* *****	1332572768	1332572799	,1332573183,1332573184
0100 1111 0110 1101 0111 0010 1*** *****	1332572800	1332572927	,1332574207,1332574208
100 1111 0110 1101 0111 0011 *****	1332572928	1332573183	,1332576255,1332576256
0100 1111 0110 1101 0111 01***** ****	1332573184	1332574207	,1332609023,1332609024
0100 1111 0110 1101 0111 1*** *****	1332574208	1332576255	,1332740095,1332740096
0100 1111 0110 1101 1***** *****	1332576256	1332609023	,1333788671,1333788672
0100 1111 0110 111* ***** *****	1332609024	1332740095	,1342177279,1342177280
0100 1111 0111 ***** *****	1332740096	1333788671	,1610612735,1610612736
0100 1111 1***** *****	1333788672	1342177279	,1879048191,1879048192
0101 ***** *****	1342177280	1610612735	,2147483647,2147483648
0110 ***** *****	1610612736	1879048191	,4294967295
0111 ***** *****	1879048192	2147483647	
1***** *****	2147483648	4294967295	

Comparing between L.H.S and R.H.S, they are equal .This prove the correctness of the proposed algorithm where processing Join operation.

11) Positive Set Difference

```
SELECT Student_Name1 from DB1
MINUS
SELECT Student_Name2 from DB2
```

TABLE XXII  
THE POSITIVE DATABASE OF STUDENT \_NAME1 WITH 24- BITS STRINGS. AND THE RESULT OF APPLYING SET DIFFERENCE OPERATION

Student_Name1	Student_Name2	Result Query in Positive
0100 0001 0110 1100 0110 1001	0100 0001 0110 1100 0110 1001	0100 0001 0111 1010 0110 0001
0100 0001 0111 1010 0110 0001	0100 1111 0110 1101 0111 0010	<b>L.H.S</b>

0100 1111 0110 1101 0111 0010		
-------------------------------	--	--

12) Negative Set Difference

TABLE XXIII

THE RESULT OF NEGATIVE AND POSITIVE SET DIFFERENCE REPRESENTATION

Result of Negative Set Difference	Min	Max	MergeSort(Min,Max)	Result in Positive Representation
0100 0001 0111 1010 0110 0000	4291168	-	(4291168,4291170,	(4291169) <sub>10</sub> =(0100 0001 0111 1010 0110
0100 0001 0111 1010 0110 001*	4291170	4291171	4291171)	0001) <sub>2</sub> <b>R.H.S</b>

Comparing between L.H.S and R.H.S, if they are equal .This prove the correctness of the proposed algorithm where processing Set Difference operation.

V. CONCLUSIONS

This research introduced a new algorithm to process query with positive data on NDB and get result in positive data representation. This algorithm introduced a secure methodology to query on NDB. Consequently, when NDB used to secure database, this proposed algorithm will become valuable to query safety on this database. This methodology may introduce strong tool to secure databases.

REFERENCES

- [1] Esponda, F., Forrest, S. and Helman, P. (2004b). Enhancing Privacy through Negative Representations of Data. Technical Report, Department of Computer Science, University of New Mexico: A667894.
- [2] Esponda, F., Forrest, S. and Helman, P.A.(2004). formal framework for positive and negative detection. s.l. : IEEE Transactions on Systems, Man, and Cybernetics.
- [3] Esponda, F., Ackley, E. S., Forrest, S. and Helman, P. (2005). On-line Negative Databases (with Experimental Result). *International Journal of Unconventional Computing*, 1(3):201-220.
- [4] Esponda, F. (2005). Negative Representations of Information, Ph.D. thesis. s.l. : University of New Mexico.
- [5] Jia, Haixia, Cristopher Moore, and Doug Strain (2005). "Generating Hard Satisfiable Formulas by Hiding Solutions Deceptively." *Proceedings of the National Conference on Artificial Intelligence*. Vol. 20. No. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press.
- [6] Esponda, F. (2006). Negative Surveys. *Arxiv: math/0608176*.
- [7] Esponda, F., Trias, S., and Forrest, S.,A (2007). relational algebra for negative databases. s.l. : Technical report TR-CS, University of New Mexico.
- [8] Esponda, F., Ackley, E. S., Helman, P., Jia, H. and Forrest, S.(2007a). Protecting Data Privacy through Hard-to-Reverse Negative Databases. *International Journal of Information Security*, 6(6): 403-415.
- [9] Dasgupta, D. and Azeem, R. (2007). A Negative Authentication System. Technical Report, Department of Computer Science, The University of Memphis: CS-07-001.
- [10] Esponda, F. (2008). Everything That is Not Important: Negative Databases. s.l. : IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE.
- [11] Dasgupta, D. and Azeem, R. (2008). An Investigation of Negative Authentication Systems. *Proceedings of the Third International Conference on i-Warfare & Security (ICIW 2008)*, Omaha, Nebraska, USA.
- [12] Esponda, F., Forrest, S. and Helman, P. (2009). Negative Representations of Information. *International Journal of Information Security*, 8(5): 331-345.