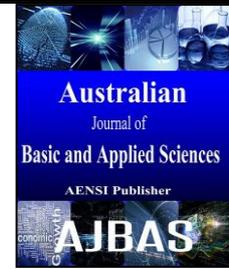




ISSN:1991-8178

## Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com



### Efficient Processing of Information Retrieval Using Fuzzy Sql Queries

<sup>1</sup>D. Veni and <sup>2</sup>Dr.K.R.Chandran<sup>1</sup>Research Scholar, Anna University, Coimbatore, Tamilnadu, India.<sup>2</sup>Information Technology Department, PSG College of Technology, Coimbatore, Tamilnadu, India

#### ARTICLE INFO

##### Article history:

Received 16 April 2015

Accepted 12 June 2015

Available online 26 June 2015

##### Keywords:

Fuzzy queries, Threshold, Fuzzy logic, Relational database, Fuzzy Structural Query Language.

#### ABSTRACT

The Database Management System permits the user to store and retrieve data along with security measure. The Structured query language (SQL) is developed for querying the relational database, but it provides the inefficiency while executes the complex queries. This paper proposes a new query system to help the user, to define their requirement for data extraction by applying fuzzy query on complex database and comparison of queries are made with priority condition and threshold. Some characteristics of FSQ are described in this work. This work reduces the complexity of the mathematical problem compared to the traditional approach which will be explained in the performance evaluation part.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: D. Veni and 2Dr.K.R.Chandran., Efficient Processing of Information Retrieval Using Fuzzy Sql Queries. *Aust. J. Basic & Appl. Sci.*, 9(20): 644-650, 2015

#### INTRODUCTION

The Database Management System (DBMS) (Robert J. Robbins, 1995) technology permits the user to store and retrieve data along with security measure. The main objective of the DBMS is to provide required information at the time of retrieval process. The retrieval process is performed by interacting with the user and database itself. It is used in various areas such as banking, finance, university and credit card transaction. Based on grossly imprecise perception of the world knowledge only, user knows what they need from database. To extract the data from database in a readable format, query is used.

The Structured query language (SQL) (Hossain, M.I., M.M. Ali, 2012; Melton, Jim; Alan R Simon, 1993) and (Kim, W., 1982) is developed for querying the relational database. It is introduced in (Muralikrishna, M., 1992). After that SQL I is used to manage variety of relational database management system, but it provides the inefficiency while executes the complex queries. The user faced problem in defining the requirement for data when applying the exact query condition. The proposed work improves the capability of traditional querying techniques by adding some additional features in it, to deal the uncertain or imprecise information (Chamberlin, D and Boyce Sequel; SIGMOD 1974). The classical query language is extended by integrating them with fuzzy logic system and fuzzy queries (Imielinski, T. and W. Lipski Jr., 1984;

Zemankova, M. and A. Kandel, 1985; Kacprzyk, J., et al., 2002; Yang, Q., et al., 2001). The fuzzy logic reduces the complexity of the mathematical problem better than traditional approach (Medina, J.M., O. Pons, M.A. Vila, 1994).

This paper proposes a new query system to help the user, to define their requirement for data by applying fuzzy query on complex database. Fuzzy SQL (FSQL) is SQL that operates on the fuzzy attribute values (Patrice Buche,). The FSQ returns the subset of database for each row that matches the particular search criteria, whereas the SQL just returns the subset of database as a query result which matches search criteria. This is the reason that FSQ reduces the complexity of the mathematical problem. This work proposes Fuzzy SQL Queries which can be applicable to both relation database and statistical database. This work supports the flexible queries based on linguistic expression from the user point of view.

The remaining work of this paper is organized as follows; Section 2 describes some of the related works which have been referred to implement this paper. Section 3 describes the architecture of database management system, a detailed explanation of the fuzzy logic, fuzzy set theory and the fuzzy SQL queries. Section 4 presents the experimental result of this work and Section 5 concludes this paper.

#### Related works:

The fuzzy queries with fuzzy logic and fuzzy set theory are mainly used for the information retrieval system. The researcher applies the fuzzy queries to extract various kind of information. Some of the researcher's approach relevant to this work is explained below.

Celia Rosline *et al* (2014) proposed a system to eliminate the usage of numerical threshold values in wireless sensor network (WSN). The author handles different types of fuzzy queries for animal monitoring application. Routing algorithm with the utilization of minimum Bounding Rectangle based routing and static nodes is applied to retrieve the query result efficiently.

Juan Miguel Medina *et al* (2012) presents a Fuzzy Object-Relational Database Management System to retrieve the medical images. Based on the pathologies indicator presence and absences, the system stores the medical data with content of image information. The author considers the X-ray image of scoliosis patient for this work. From this, the spine descriptions are collected and the queries are applied to extract the set of images.

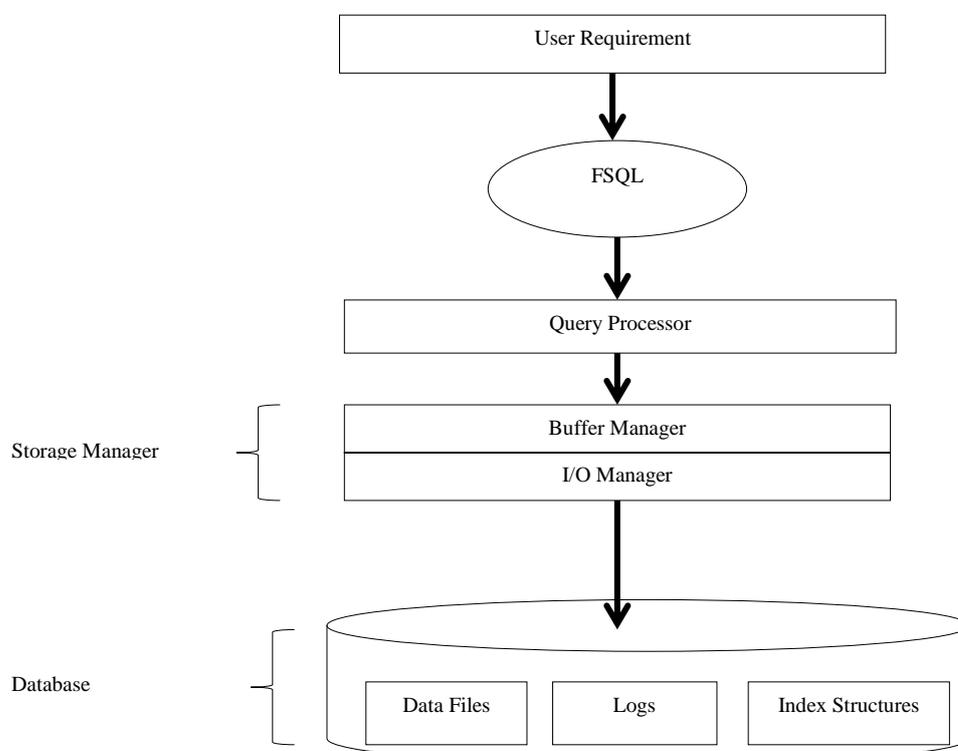
Patrice Buche *et al* (2005) proposed a multiview fuzzy query to retrieve the incomplete, heterogeneously structured data and imprecise data in a relational database. The author considers the Sym'Previous database which contains the above mentioned data. And a MIEL method is used to query this database. The heterogeneity data can be

retrieved by MIEL queries through several views. And to extract the inconsistent data, MIEL uses the fuzzy set along with the ontology. The fuzzy pattern matching is used by the MIEL to obtain the imprecise data stored in a database.

Shyi-Ming Chen and Yih-Jen Horng (1999) present a novel fuzzy queries system by expanding the fuzzy concept network to retrieve the documents. The expended fuzzy concept is modeled by relation matrix and relevance matrix where the relation matrix elements contain the fuzzy relationship and relevance matrix elements contain the degree of relevance between concepts. These relation and degree is inferred by the transitive closure of their respective matrix. The author said that this work allows performing the fuzzy queries in a flexible and an intelligent manner.

#### **Proposed work:**

The overall process of the proposed work is explained in the following architecture diagram. Initially the users describe their requirement by a FSQL query, and then these queries are processed by fuzzy logic through fuzzy set theory. With the help of the buffer manager and I/O Manager, the required information is retrieved from the database. The proposed work is focused on improving the fuzzy queries, which has the capacity to extract the data from both the relational database and the statistical database.



**Fig. 1:** Architecture of DataBase Management System

### 1.1. Fuzzy set theory and Fuzzy Database:

The fuzzy set theory is generalization of classical theory. In fuzzy set the elements represent their membership grade. Let  $S$  be the universe of discourse, 'a' be the generic element of  $S$  then,  $S = \{a, a \cup S\}$ . The membership function of the fuzzy set is  $\mu_A(a)$ , where  $A$  is fuzzy set element and 'a' is real number in the interval  $[0, 1]$ . If  $X$  is a collection of objects denoted by 'a', then a fuzzy set  $FS$  in  $X$  is denoted as  $FS = \{(x, \mu_A(a)) | a \in X\}$ .

### 1.2. Query capabilities:

The Existing Fuzzy SQL Query is given below

*Select \* from tbl\_dataset where age =(select max(age) from tbl\_dataset) AND salary =(select max(salary) from tbl\_dataset) ;*

The following query shows how the normal query is modified with linguist expression

*Select \* from tbl\_dataset  
where age="extremely old" AND  
salary ="a little high";*

Basically the condition is mentioned as age = "some numeric value" but in the above query, age is represented as extremely old. From the word itself, user can understand how the condition is given. Similarly salary= "a little high" is also given in a linguistic form. Likewise, fuzzy quantities are expressed by "about", "approximate", "short", "tall", "extremely", "very", "quite", "general". These kind of linguistic expressions are used in the relational database, whereas in statistical database Big value ( $\geq$ ), Small value ( $\leq$ ), About value ( $\approx$ ), are applied to extract the required data. Some of the other fuzzy characteristics are

**Table 1:** New Characteristics in Fuzzy SQL

Possibility	Necessity	Significance
FEQ or F=	NFEQ or NF=	Possibly/Necessarily Fuzzy Equal than...
FDIF, F!= or F<>	NFDIF, NF!= or NF<>	Possibly/Necessarily Fuzzy Different to...
FGT or F>	NFGT or NF>	Possibly/Necessarily Fuzzy Greater Than...
FGEQ or F>=	NFGEQ or NF>=	Possibly/Necessarily Fuzzy Greater or Equal than...
FLT or F<	NFLT or NF<	Possibly/Necessarily Fuzzy Less Than...
FLEQ or F<=	NFLEQ or NF<=	Possibly/Necessarily Fuzzy Less or Equal than...
MGT or F>>	NMGT or NF>>	Possibly/Necessarily Much Greater Than...
MLT or F<<	NMLT or NF<<	Possibly/Necessarily Much Less Than...
FINCL	INCL	Fuzzy Included in... / Included in...

Table 1 describes some of the characteristics of Fuzzy SQL; these characteristics are further improved by the following techniques in which the FSQL returns the subset of database for each row that matches the particular search criteria, whereas the SQL just returns the subset of database as a query result, which matches search criteria.

*select \* from tbl\_dataset where age =37 OR weight>1500;*

The above query is the example for existing SQL query for table selection. In this particular query, a table is selected based on two conditions. Among two conditions, if any one is true, the query executes. But the improved fuzzy query selects the required rows by satisfying all the two conditions. The Fuzzy SQL Query is as follows:

*SELECT \* FROM tbl\_dataset where Age FEQ 37 THOLD 0.5  
AND WEIGTH FGT 1500 THOLD 10;*

### 1.3. Flexible Data Queries:

The entity in database should satisfy the needs of the queries  $q_f$ , if it is based on the fuzzy logic. For the query  $q_f$  the set of answer  $C(q_f)$  is described as follows

$$C(q_f) = \{(u, \mu(u)) | u \in N \wedge \mu(u) > 0\} \quad (1)$$

Where  $\mu(u)$  refers ,how the selected entity  $u$  meets the query criteria. When  $\mu(u) = 1$  ,then the

entity fully meets the query criteria. The entity  $u$  partially satisfies the query criteria, when the value of the  $\mu(u)$  lies between the interval  $(0,1)$ . Existing SQL Queries formed as below to provide the flexibility

*select workclass,age from tbl\_dataset  
where age = (select Min(age) as middle from  
tbl\_dataset) OR weigh>1500  
OR salary >( select max(salary) from  
tbl\_dataset);*

Fuzzy SQL language is extended to satisfy the user to query the database in a linguistic expression. The extension designed for this work is, Fuzzy constant and fuzzy comparators. The fuzzy constants are UNKNOWN, UNDEFINED, NULL,  $[n,m]$ ,  $n+m$ . Instead of normal comparator ( $=, >, < \dots$ ), some new comparators are used, such as FEQ, to indicate the equal than operation, FGT- Fuzzy Greater than, FLT Fuzzy less than, MGT-much greater than.

According to the above mentioned facts, the query is designed as follows

*Select Age from tbl\_dataset where age is small  
And salary is high and weigh is about 1500*

In the above example, small, high and above fuzzy set is used instead of ( $<, >$ ).

### 1.4. Priority, Weight and Threshold in Fuzzy SQL Systems:

Priority can be applied to a fuzzy query through Prioritized Fuzzy Constraint Satisfaction Problem

(PFCSP). The PFCSP is derived from fuzzy constraint satisfaction problem. Priority e-norm is the key factor used in the PFCSP. To obtain an exact result with priority, the PFSQL set the condition in WHERE Clause. The basic Fuzzy Constraint Satisfaction Problem (FCSP) is defined with three tuples  $(V, K, F^c)$ , where V is set of variables and it is defined as  $V = \{v_i | i = 1, 2, \dots, n\}$ , K is finite set of domain, denoted as  $K = \{k_i | i = 1, 2, \dots, n\}$ . Finally  $F^c$  is a set of fuzzy constraints. From the above mentioned factor the fuzzy constraint satisfaction problem is defined as

$$F^c = \left\{ Z^f \left| \mu_{Z^f} : \left( \prod_{v_j \in \text{var}(Z^f)} K_j \right) \rightarrow [0,1] \right. \right\} \quad (2)$$

To make the PFCSP into FCSP, the satisfaction degree has to be calculated. The global satisfaction degree is calculated by the following equation

$$\beta_\sigma(s_x) = \bigoplus \left\{ \frac{\sigma(Z^f)}{\sigma_{max}} \diamond \mu_{Z^f}(S_{\text{var}(Z^f)}) \left| Z^f \in C^f \right. \right\} \quad (3)$$

Where  $\beta_\sigma$  refers to global satisfaction degree for a valuation  $s_x$ .  $\sigma$  refers to function, where  $\sigma: Z^f \rightarrow [0, \infty]$  which is used to evaluate the priority for  $Z^f$  constant. The large value of  $\sigma$  is the highest priority and  $\sigma_{max}$  is calculated by  $\sigma_{max} = \max\{\sigma(Z_f), Z_f \in F_c\}$ . The existing SQL for priority calculation

```
select MAX(
.hours_per_week)*0.2,Max(Age)*1,Max(salary)*1
from tbl_dataset WHERE (hours_per_week
=(Select Max(hours_per_week
from tbl_dataset)) OR (salary=(Select
Max(salary)from tbl_dataset))
OR (AGE=(Select Max(AGE) from
tbl_dataset));
```

Priority of each constant is represented by the function  $\sigma$ . The aggregation of each priority is constraint by the value of that function by the  $\diamond$  operator. Then the  $\bigoplus$  operator aggregates these constraint which results in the satisfaction degree of an evaluation. The example PFCSP query for an employee is as follows

```
SELECT * FROM tbl_dataset WHERE
(Age='tall') PR 1 AND(salary='Excellent') PR 0.6
AND (hours_per_week = 'high') PR 0.4;
```

The weight is similar to PFSQL and the global satisfaction degree for weight is calculated by,

$$\gamma_w(s_x) = N(y_1 * w_1, y_2 * w_2, \dots, y_n * w_n) \quad (4)$$

Where  $\gamma_i = \mu_{y_i}(s_x)$  the local satisfaction degree and N is the t-norm. The example WFCSP query for employees would be the following.

```
SELECT * FROM tbl_dataset WHERE
weigh=(select Max(weigh) from tbl_dataset)
OR
hours_per_week=(select Max(hours_per_week)
from tbl_dataset)
OR salary=(select Max (salary) from
tbl_dataset);
```

The queries explain that the top 100 rows are selected from the database table, where three conditions are given in the query. Based on any one condition the query is executed. The equivalent FSQL Query is

```
Select (MIN(Age*0.2,Salary*0.6,Weigh*1))
from tbl_dataset WHERE (Age='tall') AND
(salary='Excellent') AND (weigh='high');
```

The proposed query executes to find the weight of the employees based on three conditions, where all of them should have to satisfy the condition given. From these, the exact value is retrieved from the database.

Final step is adding the threshold to each condition. Similar to priority, threshold is also mentioned in the WHERE clause. If the threshold is not satisfied, this system drops the data row from query result. The following example is taken for applying threshold. This situation is completely different from priority and weighted queries.

```
SELECT * from tbl_dataset where Age=(Select
Max(Age)from tbl_dataset ) AND
Salary =(Select Max(Salary) from tbl_dataset )
And Weigh= (Select Max(Weight) from tbl_dataset )
SELECT * from tbl_dataset where (Age='tall')
THRESHOLD 1 AND
(Salary ='Excellent') And (Weigh= 'high')
THRESHOLD 0.4
```

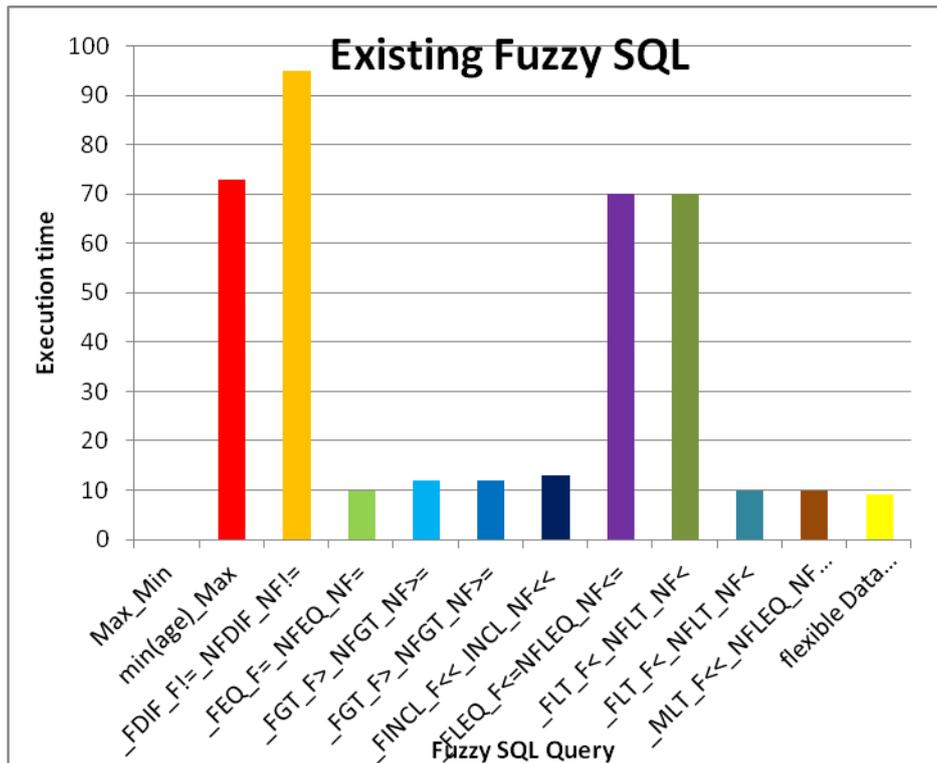
## 2. Experimental Result:

The performance of the proposed paper is described in this section. The proposed work is experimented by the adult dataset taken from UCI Machine Learning Repository. This dataset is extracted from the census database by Barry Becker. The experimental result is compared with the existing Fuzzy Sql algorithm.

Table 1 describes the execution time of the previous fuzzy sql queries; the amount of time to execute a particular query is high. The execution of comparison operation such as !=, <= and < took larger time. This will be represented in the below graph as figure 2.

**Table 2:** execution time of the existing fuzzy SQL queries

1	Existing FSQLExisting_Max_Min	0.9233576
2	Existing Fuzzy logic System_min(age)_Max	73.631983000000005
3	Existing Fuzzy SQL for Fuzzy database_FDIF_F!=_NFDIF_NF!=	95.0821959999999996
4	Existing Fuzzy SQL for Fuzzy database_FEQ_F=_NFEQ_NF=	10.470884
5	Existing Fuzzy SQL for Fuzzy database_FGEQ_F>=_NFGEQ_NF>=	11.549234
6	Existing Fuzzy SQL for Fuzzy database_FGT_F>_NFGT_NF>=	11.674948000000001
7	Existing Fuzzy SQL for Fuzzy database_FINCL_F<<_INCL_NF<<	12.088687
8	Existing Fuzzy SQL for Fuzzy database_FLEQ_F<=NFLEQ_NF<=	70.168136000000004
9	Existing Fuzzy SQL for Fuzzy database_FLT_F<_NFLT_NF<	11.744789000000001
10	Existing Fuzzy SQL for Fuzzy database_FLT_F<_NFLT_NF<	69.583145000000002
11	Existing Fuzzy SQL for Fuzzy database_MLT_F<<_NFLEQ_NF<<	10.391545000000001
12	Existing flexible Data Query_max_min	8.8268199999999997



**Fig. 1:** Execution Time of Existing Fuzzy Sql Queries

The proposed work improves the fuzzy queries, which reduce the complexity of the mathematical

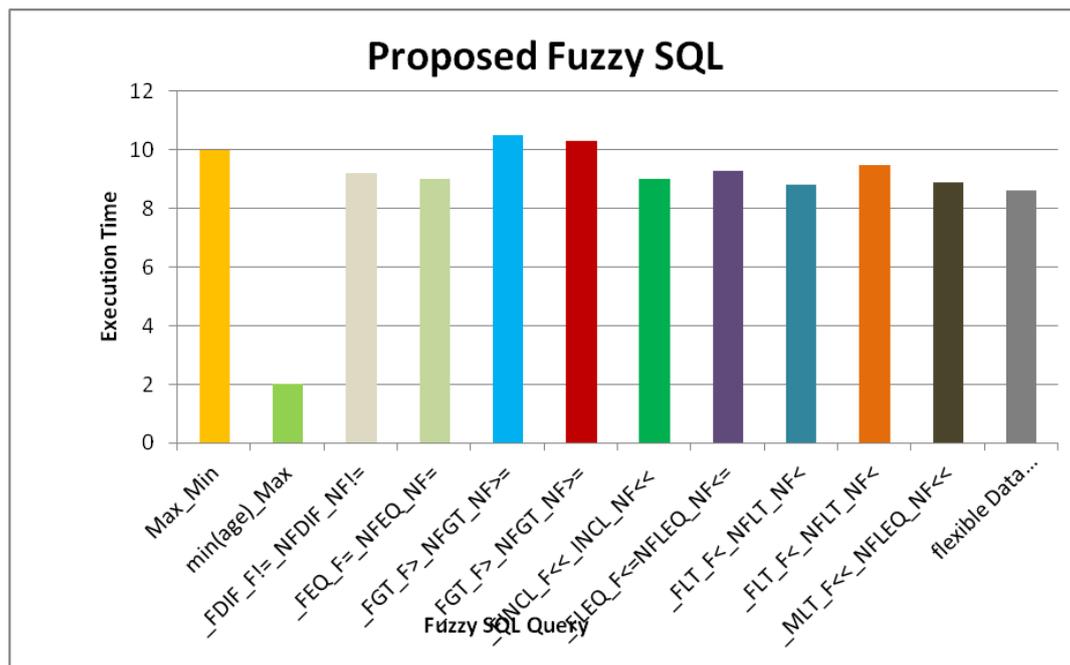
problem better than the traditional approach and the execution time is low.

**Table 2:** Proposed fuzzy sql queries execution time:

s.no	Proposed Algorithm Name	Elapsed Time
1	Proposed flexible Data Query_high_small	10.10
2	Proposed FSQLExtremely_middle_moderate	2.0376891000000001
3	Proposed Fuzzy logic system_Average_high	9.1682039999999994
4	Proposed Fuzzy SQL for Fuzzy database_FDIF_F!=_NFDIF_NF!=	8.9944389999999999
5	Proposed Fuzzy SQL for Fuzzy database_FGEQ_F>=_NFGEQ_NF>=	10.491557
6	Proposed Fuzzy SQL for Fuzzy database_FEQ_F=_NFEQ_NF=	10.164979000000001
7	Proposed Fuzzy SQL for Fuzzy database_FGT_F>_NFGT_NF>=	9.202007
8	Proposed Fuzzy SQL for Fuzzy database_FINCL_F<<_INCL_NF<<	9.3394560000000002
9	Proposed Fuzzy SQL for Fuzzy database_FLEQ_F<=NFLEQ_NF<=	8.822350000000001
10	Proposed Fuzzy SQL for Fuzzy database_FLT_F<_NFLT_NF<	9.464052000000006
11	Proposed Fuzzy SQL for Fuzzy database_FLT_F<_NFLT_NF<	8.8944270000000003
12	Proposed Fuzzy SQL for Fuzzy database_MLT_F<<_NFLEQ_NF<<	8.653893000000001

Table 2 explains in detail about the increased performance of this work and it shows the reduced execution time in the proposed work for the same

fuzzy queries which is explained in table 1. For all type of operation, the improved fuzzy queries take very less amount of time for execution.



**Fig. 2:** Execution Time of Proposed Fuzzy SQL Queries

### Conclusion:

The FSQL language permits the user to represent their requirement through a linguistic condition. Therefore the meaning of the queries can be understood easily for the first glance. This leads the queries into easily readable and modifiable. Fuzzy queries reduced the null value problems by proving some data that is relevant to the query condition. Through the integration of several fuzzy query approaches, an efficient solution is obtained for the end user. The proposed query system helps the user, to define their requirement by applying fuzzy query on complex database. Comparison of queries is made with priority condition with threshold. A simple data mining tool is designed with the sql and the fuzzy queries. The improved fuzzy queries and fuzzy data processing will be used in various domains. In future work, interpolative Boolean algebra will be applied to fuzzy queries to improve its performance.

### REFERENCE

- Robert J. Robbins, 1995. "Database Fundamentals".
- Hossain, M.I., M.M. Ali, 2012. "SQL query based data synchronization in heterogeneous database environment IEEE International conference on Computer Communication and Informatics (ICCCI), 1-5.
- Melton, Jim; Alan R Simon, 1993. "1.2. What is SQL?". Understanding the New SQL: A Complete Guide. Morgan Kaufmann. P: 536. ISBN 1-55860-245-3.
- Kim, W., 1982. "On Optimizing an SQL-like Nested Query," ACM Trans.Data Systems, Sept.

Muralikrishna, M., 1992. "Improved Unnesting Algorithms for Join Aggregate SQL Queries," Proc. Very Large Databases.

Chamberlin, D and Boyce Sequel: A Structured English Query Language. In ACM

SIGMOD 1974. Workshop on Data Description, Access and Control, Ann Arbor, USA, pp: 249-264.

Imielinski, T. and W. Lipski Jr., 1984. "Incomplete Information in Relational Databases, J. ACM, 31.

Zemankova, M. and A. Kandel, 1985. "Implementing Imprecision in Information Systems, Infomation Science.

Kacprzyk, J., G. Pasi, P. Vojtáš, S. Zadrozny, 2000. Fuzzy querying: Issues and perspectives. Kybernetika, 36(6): 605-616.

Yang, Q., W. Zhang, C. Liu, J. Wu, H. Nakajima and N.D. Rishe, 2001. "Efficient Processing of Nested Fuzzy SQL Queries in a Fuzzy Database", IEEE Trans. On Knowledge and Data Eng., 13(6): 884-901.

Medina, J.M., O. Pons, M.A. Vila, 1994. "GEFRED, A Generalized Model of Fuzzy Relational Databases". Information Sciences, 76(1-2): 87-109.

Patrice Buche, Catherine Dervin, Olivier Haemmerlé, and Rallou Thomopoulos, "Fuzzy Querying of Incomplete, Imprecise, and Heterogeneously Structured Data in the Relational.

Model Using Ontologies and Rules", 2005. IEEE Transactions on fuzzy systems, 13: 3.

Radojević, D., 2008. "Interpolative realization of Boolean algebra as a consistent frame for gradation and/or fuzziness". In Forging New Frontiers: Fuzzy Pioneers II Studies in Fuzziness and Soft Computing. Springer-Verlag, Berlin / Heidelberg, pp: 295-318.

Galindo, J., A. Urrutia, M. Piattini, 2006. *Fuzzy Databases: Modeling Design and Implementation*. Hershey, USA: IDEA Group.

Zimmerman, J., 2000. "Fuzzy Set Theory – And It's Applications", Kluwer Academic Publishers, Norwell, Massachusetts, U.S.A.

Galindo, J., A. Urrutia, M. Piattini, 2006. *Fuzzy Databases: Modeling Design and Implementation*. Hershey, USA: IDEA Group.

Shyi-Ming Chen and Yih-Jen Horng, "Fuzzy Query Processing for Document Retrieval

Based on Extended Fuzzy Concept Networks, 1999. *IEEE transactions on systems, man, and cybernetics—part b: cybernetics*, 29: 1.

Juan Miguel Medina, Sergio Jaime-Castillo, Carlos D. Barranco and Jesús R. Campaña, 2012. "On the Use of a Fuzzy Object-Relational Database for Flexible Retrieval of Medical Images", *IEEE Transactions on fuzzy systems*, 20: 4.

Celia Rosline, A., V.S. Felix Enigo, 2014. "Fuzzy query processing in wireless sensor networks for Animal Health Monitoring", *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp: 1094-1098.

Patrice Buche, Catherine Dervin, Ollivier Haemmerlé, and Rallou Thomopoulos, 2005. "Fuzzy Querying of Incomplete, Imprecise, and Heterogeneously Structured Data in the Relational Model Using Ontologies and Rules", *IEEE Transactions on fuzzy systems*, 13: 3.

Tré, G.D. et al., 2000. "A generalized object-oriented database model," in *Recent Research Issues on the Management of Fuzziness in Databases*, 53: 155-182.

Ma, Z.M. and Y. Li, 2008. "A literature overview of fuzzy database models," *J. Inf. Sci. Eng.*, 24: 189-202.