Distr.
GENERAL

Working Paper No.12
14 March 2008

ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and ECONOMIC COMMISSION FOR EUROPE CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION STATISTICAL OFFICE OF THE EUROPEAN COMMUNITIES (EUROSTAT)

ORGANISATION FOR ECONOMIC COOPERATION AND DEVELOPMENT (OECD) STATISTICS DIRECTORATE

Meeting on the Management of Statistical Information Systems (MSIS 2008)
(Luxembourg, 7-9 April 2008)

Topic (ii): Statistical information systems architecture

FUZZY STRUCTURED QUERY LANGUAGE (SQL) FOR STATISTICAL DATABASES

Supporting Paper

Prepared by Miroslav Hudec, Infostat, Slovak Republic

I. INTRODUCTION

1. The structured query language (SQL) is used to obtain data from relational databases. Its advantages, among others, are the optimised work with relational database management systems (RDBMS) and understandable interpretation for users. The simply SQL query is as follows:

```
select attribute_1,…,attribute_n
from T                                                    (1)
where attribute_p > P and attribute_r < R.
```

2. The result of the query is shown in graphical mode in Figure 1. Values P and R delimit the space of interesting data. Small squares on the graph show records that satisfy and not satisfy the query criteria. In the graph is obviously shown that two records are very close to satisfy query criteria whereas other records either satisfy the query or are far for satisfing it.

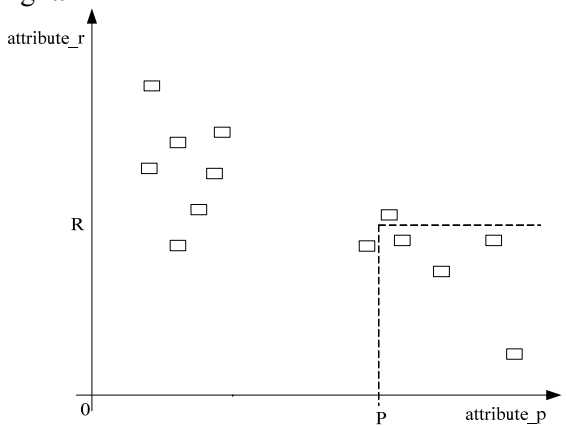

Figure 1: The result of classical query

3. An answer to the question why users search databases could be as follows: To retrieve data needed to make a decision. It is easy for users to determine, which columns (attributes) of records they want and from which tables or views. Boundary values separate interesting records from not interesting ones and they are defined in the WHERE clause of a query. It is not always simple to know or identify them. This is usually the case when the boundary values are numeric ones. In cases when the user can not unambiguously separate data he is interested in from data he is not interested in by sharp boundaries or when user wants to obtain data that are very close to satisfy queries and to know the index of distance to full query satisfaction, it is necessary to adapt the SQL to these requirements.

4. Classical SQL makes a brittle selection. This means that the record would not be selected even it is extremely close to the intent of the query. But this is the penalty to pay to use the crisp logic in selection criteria.

5. If the classical SQL is used for solving this problem, the SQL relaxation will have to be used in the following form:

```
select atribut_1,…,atribut_n
from T                                                    (2)
where atribut_p > P-p and atribut_r < R+r.
```

6. Values p and r are used to expand the initial query to encompass data that are very close to satisfy the initial query. This approach induces several additional problems. First, the meaning of the initial query is diluted in order to capture adjacent records. The meaning of a query: “where attribute_p is more than P” is changed and adjacent records satisfy a query in the same way as initial ones. More precisely, the difference between original and adjacent data (caught records along the “edge” of interesting space) does not exist. Second, problem rises from the question: what about records that are very close to satisfy new expanded query and it is useful to make another expanding of a query. In this way more data from the database are selected, but the user has lost the accuracy of his query. These two problems show that instead of changing the boundary conditions in the WHERE clause, it is necessary to change the way in which the WHERE clause is evaluated.

7. The aim of this work is to present the query improvement with the fuzzy SQL approach. This development enables supporting queries based on linguistic expressions from user’s point of view and also enables accessing classical relational databases in the unchanged structure.

8. The generalized logical condition (GLC) for the WHERE part of the SQL based on linguistic expressions is created. This GLC enables matching fuzzy and classical constraints in the same WHERE clause and to select only records that have the query satisfaction greater than zero. These records are transferred to the client side where t-norm and t-conorm functions, which can be easy aggregated to n variable case, are used to calculate query satisfaction index for each of these records. The query compatibility index (QCI) indicates how the selected record satisfies a query request. If the record fully satisfies query, the QCI value is 1 and if record partially satisfies query conditions, QCI value is in (0,1) interval and represents the distance to the full query satisfaction. The QCI value 0 means that the record does not satisfy a query. It is also possible to use additional filtering functions to choose appropriate number of records or to set the threshold value of the QCI.

9. This approach enables also to use the same query (on linguistical level) in different time periods of data selection. The meaning of the query remains the same, only the parameters and shapes of fuzzy sets are changeable to allow the query adaptation to new requirements and changed situations of analysed tasks.

10. Many applications have created incalculable accesses to wide variety of data, for example statistical data, which are in many countries more or less free. The access system based on the SQL is generally good and understandable. The data and the classical access to data are simply not enough in many cases. The improvement consists in the language and methods in „communication“ with data. In this research the language is presented by linguistic expressions. Fuzzy sets and fuzzy logic are methods for „communication“ with data on higher level in comparison with the classical SQL.

II. SQL WITH FUZZY COMPONENT

11. Fuzzy queries have emerged in the last 20 years to deal with the necessity to soften the Boolean logic in relational databases. A fuzzy query system is an interface to human users to get information from database using (quasi) natural language sentences. Many fuzzy query implementations have been proposed, resulting in slightly different languages. Although there are some variations according to the particularities of different implementations, the answer to a fuzzy query sentence is a generally a list of records, ranked by the degree of matching [2].

12. In this research the goal is to change values P and R from query (1) with linguistic expressions and to calculate the lower bound of QCI from these linguistic expressions. Thus calculated lower bound is used as a parameter for database queries to select records that have QCI>0. In the next step appropriate t-norms or t-conorms are used to calculate QCI values for all retrieved records. Figure 2. shows steps and modules of this approach. This approach decreases the amount of transferred data across nets and calculation of QCIs is not significant burden for client computers.

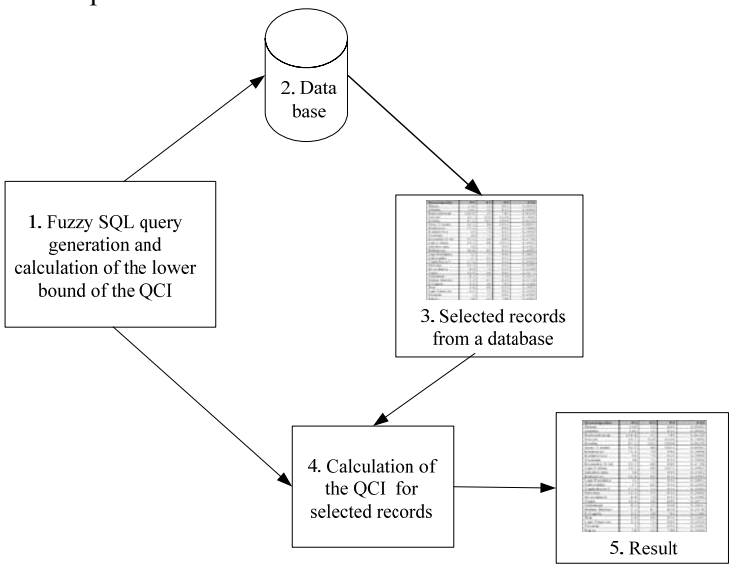

Figure 2: Structure of the SQL with fuzzy component

A. Linguistic expressions

13. Crisp values in the WHERE clause (P and R) are changed with linguistic expressions. The query (1) is transformed into query:

```
select attribute_1,..., attribute_n
from T                                                    (3)
where attribute_p is Big and attribute_r is Small.
```

14. Classical conditions in queries contain these comparison operators: $\geq$, $\leq$, = , $\neq$ and **between** when numerical attributes are used. Operator $\geq$ (greater than) could be described with fuzzy set „Big value“, operator $\leq$ (less than) could be described with fuzzy set „Small value“ and operator = (equal) could be described with fuzzy set „About value“. Operator $\neq$ is the negation of the operator = so this operator is not further analysed in this paper. Similar is valid for operator **between** because it is similar to the operator = from the fuzzy point of view. The WHERE clause contains one attribute or more attributes connected with logic operators.

15. According to the above analyse three types of linguistic terms in this research are supported: big value, small value and about value of an attribute. All types are shown on Figure 3. The figure shows two kinds of fuzzy sets for each linguistic expression. The shapes of fuzzy sets could be broadened to allow better interpretability of a particular selecting task. The lower and upper limit (L_d, L_g) have the same meaning in all

fuzzy sets for query creation. In the QCI calculation step, the differences between fuzzy set shapes become important.

16. In case when it is required to find records which have a big value of the attribute, the query has the form (4) and graphical presentation is in the Figure (3.a).

```
select attribute_1,…,attribute_n
from T                                                    (4)
where attribute_p is Big.
```

17. The user determines the shape of fuzzy set, lower limit of fuzzy set (L_d) and value of full query satisfaction (L_p). The lower limit becomes part of the WHERE clause. This clause access the database and selects records that have QCI>0. This query has the form shown in (5).

```
select attribute_1,…,attribute_n
from T                                                    (5)
where attribute_p >=Ld
```

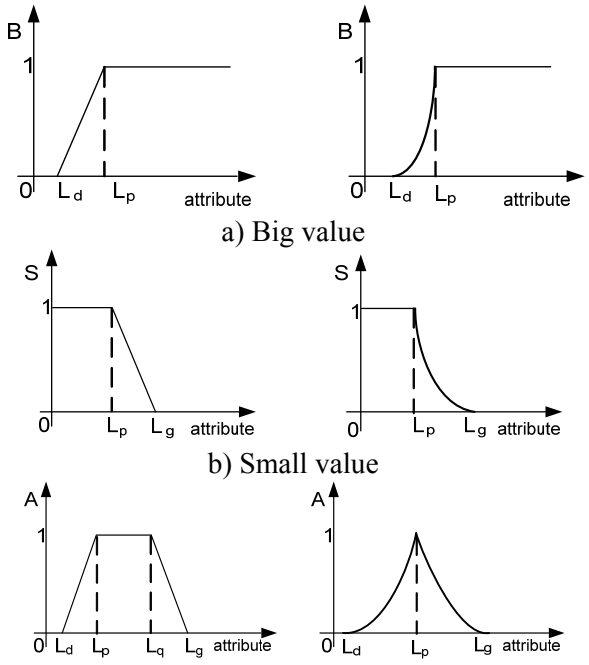

Figure 3: Fuzzy sets

18. In the case when it is required to find records, which have a small value of attribute (Figure 3b), the query has the following structure:

```
select attribute_1,…,attribute_n
from T                                                    (6)
where attribute_p <=Lg
```

19. In the case when it is required to find records that have attribute value about the exact value (Figure 3.c), the query has the following structure:

```
select attribute_1,…,attribute_n
from T                                                    (7)
where attribute_p >=Ld and attribute_p <=Lg
```

20. Values L_d, L_p, L_q, L_g are used to define the analytical form of the fuzzy set. According to this analytical form, the QCI values are calculated for all selected records.

B. Generalised logical condition for a fuzzy query

21. In real cases the WHERE clause of a query is a complex logical expression. It is not complicated to add constraints for string or date variables, as well as classical constraints for numeric variables into queries having fuzzy constraints.

22. According to research and above mentioned facts, the following GLC is achieved:

$$\text{where } \overset{n}{\underset{i=1}{\otimes}}(a_i \circ L_{ix}) \tag{8}$$

where n denotes number of attributes with fuzzy constraints in a WHERE clause of a query,

$$\otimes = \begin{cases} and \\ or \end{cases} \tag{9}$$

where *and* and *or* are fuzzy logical operators,
and

$$a_i \circ L_{ix} = \begin{cases} a_i > L_{id}, & \text{a}_\text{i} \text{ is Big} \\ a_i < L_{ig}, & \text{a}_\text{i} \text{ is Small} \cdot \\ a_i > L_{id} \text{ and } a_i < L_{ig}, & a_i \text{ is About} \end{cases} \tag{10}$$

where a_i is a database attribute and L is the parameter of a fuzzy set

23. If a query in the WHERE clause contains fuzzy as well as classical constraints, these classical constraints could be easy added to the WHERE clause:

where $\overset{n}{\underset{i=1}{\otimes}}(a_i \circ L_{ix})$ [and/or][atribute_m LIKE "*String"][and/or][atribute_l<Date]…. (11)

24. On the client side the QCI values for these records are calculated as the next step. In case when two fuzzy constraints are in a WHERE clause all t-norms or t-conorms can be used as an aggregation function. However, for fuzzy SQL to use more than two attributes in the WHERE clause of a query is required. From the associative rule of t-norms and t-conorms, the following functions can be easy aggregated for more than two attributes:

t-norms for *and* operator :

(a) min: $\text{QCI} = \min(\mu_F(\text{x}_{\text{ai}}))$, i=1,...,n (12)

(b) product: $\text{QCI} = \prod(\mu_F(\text{x}_{\text{ai}}))$, i=1,...,n (13)

(c) bounded difference (BD): $\text{QCI} = \max(0, \sum_{i=1}^{n} x_{ai} - n + 1)$ (14)

t-conorms for *or* operator:

(d) max: $\text{QCI} = \max(\mu_F(\text{x}_{\text{ai}}))$, i=1,...,n (15)

(e) bounded sum (BS): $\text{QCI} = \min(1, \sum_{i=1}^{n} x_{ai})$ (16)

where μ_{Fi} denotes the function for i-th fuzzy set and x_{ai} is the membership degree of the attribute a to the i-th fuzzy set.

25. The differences between t-norms are important when the correlation between attributes is taken into account. In case when the correlation is unknown or does not exist, min or product t-norms are used. In other cases, the BD t-norm is used. The similar is valid for t-conorms. If a correlation exists the BS t-conorm is used and in other cases the max t-conorm is used.

C. The GLC in object-oriented programming

26. The GLC can be designed as a generic class in an object-oriented programming language. The inheritance and replacement are used to obtain particular logical condition from the generic one. In this work the GLC is under development in the Microsoft Visual Basic 2005 programming language. In order that users could work with this system, the fuzzy module as well as modules for database connections and for providing the result in a usable and understandable form in print-ready tables and in the export formats for further use are under development. For statistical information systems, providing the result in a thematic map is also useful and is under consideration.

27. The fuzzy SQL is an independent module and it can be used when boundaries between variable values are vague. In other cases the classical SQL gives satisfying results and the request for the fuzzy SQL module does not exist.

28. The modularity of this approach supports changes and improvements for each module independently. For example a fuzzy module can be improved with the interpolative Boolean algebra while other modules remain the same.

29. In the further development of this application it is of the interest to improve fuzzy module with setting the α -cut value in order to select minimal or demanded number of records from a database. This is possible by relaxing original fuzzy set parameters to ensure that a useful number of records are retrieved from the database. If too many records are retrieved the query is made finer to reduce the number of retrieved records.

30. The duration and the state of art of this approach depends also on the theoretical and practical development of fuzzy database systems. In fuzzy databases the fuzzy query is a part of the fuzzy database management system and a special fuzzy query development is not necessary. Many statistical databases, known to the author, are developed in classical relational DBMS. This trend dominates in development of new statistical information systems and databases too. The fuzzy SQL presented in this work has the significant perspective for usage. Fuzzy SQL module could be extended to enable the data extraction from Excel spreadsheets too. Many statistical data are in CSV or data cube formats in Excel spreadsheets.

D. Interface design

31. The interface to the system is a standard windows desktop application shown in Figure 4. This interface is under design and it will be able to facilitate users to:

(f) choose indicators and territorial units from the database for the classical and fuzzy part of a query. This part also contains indicator testing for fuzzy conditions. If the indicator is appropriate then fuzzy value is 1 and the indicator could be moved to the fuzzy interface;
(g) manage fuzzy conditions by selecting types and parameters of fuzzy sets and the aggregation function;
(h) see the result of a query in the same screen with possibilities to export results in useful formats for additional work with data (xls, xml, doc,...) and to print it;
(i) save the query for later use or to retrieve the same query on linguistical level and to change shape of fuzzy sets and their parameters.

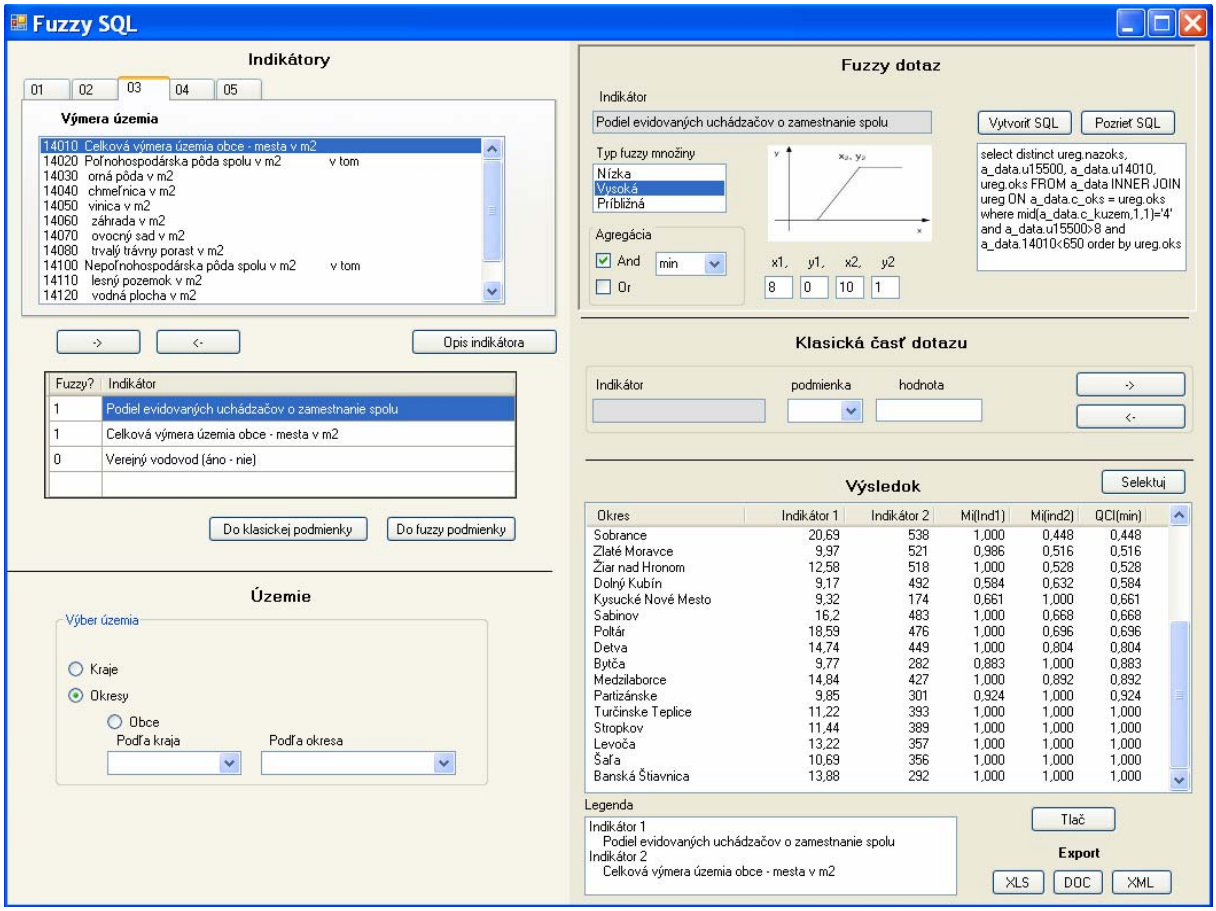

Figure 4: Fuzzy SQL Interface (proposal)

III. CASE STUDY

32. This fuzzy query approach is under development for statistical information systems. Statistics have become one of the potential applications of data mining techniques, due to the great volume of available data from various databases. Statistics also works on data dissemination for various kinds of statistical data users. Classical tools also enable useful data dissemination but the fuzzy approach satisfies the demand on data in a human understandable form and supports the human-computer interaction in area of obtaining data using linguistic expressions.

33. This system is tested on data from the Urban and Municipality Statistics. In this case study, districts with high unemployment rate and small area size are sought. The big unemployment density is analysed as an illustrative example. The simplified query has the following form (The simplification is done for better comprehensibility. In the WHERE and FROM part of the query the names of tables and join conditions connected by primary and foreign key are skipped.):

```
select district, unemployment, area
from T
where unemployment is Big and area is Small.
```

34. Unemployment is represented by „Big value“ fuzzy set with these parameters L_d=8% and L_p =10%. The „Small value“ fuzzy set with parameters L_p=400km^2 and L_g =650km^2 describes the area of district attribute. According to these parameters the query has the following form:

```
select district, unemployment, area
from T
where unemployment>8 and area<650.
```

35. The query selects 26 of 79 districts from database. Because of non existence of correlation between these two attributes the min and product t-norms are used for calculation of the QCI. The result of query is presented in the Table 1. The value of min aggregation is used for district ranking.

36. The result is not ranked like results from a classical query by one of selected attributes. The result is also not ranked by weighted coefficients of selected attributes. In the fuzzy query all attributes have the equal importance and the ranking is done by aggregation of membership degrees to particular fuzzy sets.

Table 1. Result of the fuzzy query

District	Unemployment [%]	Area [km2]	μ (unemp)	μ (area)	QCI (min)	QCI (prod)
Bánovce nad Bebravou	8,01	462	0,003	0,752	0,003	0,002
Ružomberok	9,63	647	0,813	0,012	0,012	0,010
Košice II	8,07	80	0,034	1,000	0,034	0,034
Stará Ľubovňa	8,92	624	0,461	0,104	0,104	0,048
Topoľčany	9,23	598	0,617	0,208	0,208	0,128
Košice III	8,44	17	0,220	1,000	0,220	0,220
Spišská Nová Ves	13,92	587	1,000	0,252	0,252	0,252
Krupina	14,48	585	1,000	0,260	0,260	0,260
Gelnica	16,99	584	1,000	0,264	0,264	0,264
Svidník	12,62	550	1,000	0,400	0,400	0,400
Sobrance	20,69	538	1,000	0,448	0,448	0,448
Zlaté Moravce	9,97	521	0,986	0,516	0,516	0,509
Žiar nad Hronom	12,58	518	1,000	0,528	0,528	0,528
Dolný Kubín	9,17	492	0,584	0,632	0,584	0,369
Kysucké Nové Mesto	9,32	174	0,661	1,000	0,661	0,661
Sabinov	16,2	483	1,000	0,668	0,668	0,668
Poltár	18,59	476	1,000	0,696	0,696	0,696
Detva	14,74	449	1,000	0,804	0,804	0,804
Bytča	9,77	282	0,883	1,000	0,883	0,883
Medzilaborce	14,84	427	1,000	0,892	0,892	0,892
Partizánske	9,85	301	0,924	1,000	0,924	0,924
Turčinske Teplice	11,22	393	1,000	1,000	1,000	1,000
Stropkov	11,44	389	1,000	1,000	1,000	1,000
Levoča	13,22	357	1,000	1,000	1,000	1,000
Šaľa	10,69	356	1,000	1,000	1,000	1,000
Banská Štiavnica	13,88	292	1,000	1,000	1,000	1,000

A. Multiple usage of the same query

37. In case of multiple usage of the same query the meaning of the query remains unchanged. Only parameters of fuzzy sets are changeable. As an example, the query from the case study can be used: **User wants to select territorial units with a high unemployment rate (a_1) and a small area (a_2)**. This linguistic expression is the base for multiple usages in two different ways:

(j) User wants to select territorial units which satisfy mentioned linguistic terms in different time periods. Parameters and shapes of fuzzy sets could be changed to satisfy new requirements. In this case the state in labour market or territorial unit structure can be changed and implemented into fuzzy sets to obtain adequate values;

(k) The same query could be used for territorial unit selection in other countries or for other territorial levels in same country with the different labour market situation and territory division. Parameters of fuzzy sets are also changed to better describe the situation.

38. The combination of both approaches is also possible. This small discussion shows the importance of queries based on linguistic expressions. Table 2 shows the evolution of one fuzzy query. Linguistic meaning of the query remains unchanged. Parameters of Small and Big fuzzy sets are changeable.

Table 2. Linguistic meaning of a query is not changed

Period/ Country	Period 1	…	Period n
Country 1	select * from T where a_1 is Big and a_2 is Small		select * from T where a_1 is Big and a_2 is Small
…			
Country n	select * from T where a_1 is Big and a_2 is Small		select * from T where a_1 is Big and a_2 is Small

IV. CONCLUSION

39. The SQL is used in all major RDBMSs and also in all major information systems. The SQL is optimised to work with RDBMS. The fuzzy solution that allows the creation of queries based on linguistic expressions from the users point of view and does not change the structure and concept of obtaining data from relational databases, enables an improved usage of the SQL.

40. Expressions like: “high rate of unemployment” or “high migration level”, etc. are very often used in statistics. The goal of this application is to capture these expressions and make them suitable for database queries. In many cases (not only for statistical data), the user (analyst, decision maker,…) cannot unambiguously separate data he is interested in from data he is not interested in by sharp boundaries or the user can not expressly argue, why the chosen boundary value is the best one. Users also want to obtain data that are very close to satisfy queries and to know the index of distance to full query satisfaction.

41. The SQL and fuzzy approach creates a simple and easy to use data mining tool. The information system users have, usually on the client side, contains some software for their daily work (database updating, reports creation, etc.), moreover the software for work with geographical data on thematic maps, etc. The fuzzy SQL tool could be used to improve their work for the wide area of working tasks.

42. The fuzzy SQL is in this approach an independent module and it can be used when the user wants to use a linguistic expression in queries. The modularity of here mentioned modules allow their modifications and improvements independently. The research done in this work would continue in the research of methods for the fuzzy query module improvements.

43. The fuzzy logic is a precise multivalued logic based on the truth functionality, so the structure is not included and some results of the fuzzy logic are not into frame of the Boolean algebra. To stay in the frame of the Boolean algebra, the interpolative Boolean algebra can be used to improve queries and to keep them in the frame of Boolean algebra. Further research could be directed towards improving the fuzzy SQL by the interpolative Boolean algebra.

44. The web application with a fuzzy module for data dissemination is another way of improving this fuzzy query approach.

REFERENCES

[1] Bosc P., Pivert O., “SQLf Query Functionality on Top of a Regular Relational Database Management System”, Studies in fuzziness and soft computing, 39, 2000.

[2] Branco A., Evsukoff A., Ebecken N., “Generating Fuzzy Queries from Weighted Fuzzy Classifier Rules“, ICDM workshop on Computational Intelligence in Data Mining, 2005.

[3] Carraso R., Vila M., Galindo J., “Using dmFSQL for financial clustering”, Enterprise information systems, VII ,2006.

[4] Cox E., Fuzzy modeling and genetic algorithms for data mining and exploration, Morgan Kaufman Publishers, San Fancisco, 2005.

[5] Galindo J., Urrutia A., Piattini M., Fuzzy Databases, Modeling, Design and Implementation, Idea Group Publishing, London, 2006.

[6] Gonclaves M., Tineo L., “A web tool for web document and data source selection with SQLFI”, ICEIS, Proceeding - Databases and Information Systems Integration, 2007.

[7] Hudec M., “Fuzzy improvement of the SQL”, Balcor, 2007.

[8] Klir G., Yuan B., Fuzzy sets and fuzzy logic, theory and applications, Prentice Hall, New Jersey, 1995.

[9] Teorey T., Linghstone S., Nadeau T., Database Modeling & Design:Logical Design, Morgan Kaufmann Publishers, 2006.

[10] Werro N., Meier A., Mezger C., Schindler G., “Concept and Implementation of a Fuzzy Classification Query Language”, DMIN ,2005.