About the Performance of SQLf Evaluation Mechanisms

Yosmar López
Universidad Simón Bolívar, Departamento de Computación,
Apartado 89000, Caracas 1080-A, Venezuela
yosmara@hotmail.com

and

Leonid Tineo
Universidad Simón Bolívar, Departamento de Computación,
Apartado 89000, Caracas 1080-A, Venezuela
leonid@ldc.usb.ve

Abstract

In order to make more flexible database access the query language SQLf has been previously proposed. One of the SQLf features is the use of Fuzzy Quantifiers in Having Clause. For this kind of query, three evaluation mechanisms have been proposed: the Naïve, the Sugeno Integral Heuristics based and the Alfa-cut Derivation based. We present in this paper a formal performance study of these three mechanisms. This study has been made using a SQLf prototype build on top of a RDBMS.

Keywords: Database Fuzzy Querying, Query Performance.

1. INTRODUCTION

The problem of database fuzzy querying not is only a semantics issue but also a practical reality. The interest of some previous works has been to provide efficient evaluation mechanisms for fuzzy querying language SQLf [[1], [2], [3], [6]]. In case of fuzzy quantified queries, Bosc et al [[1]] have presented a strategy based in Sugeno Integral properties for improve query evaluation, we name it Sugeno Strategy. On the other hand, Tineo [[6]] has presented a strategy based in the distribution of the α-cut operator for evaluating fuzzy quantified queries, we name it Derivation Strategy. As ever, it is possible to apply an intuitive strategy pervaded of none improvement, we name it the Naïve Strategy. In this paper, we present the study of these strategies by mean of experimental proofs. We hope to determine which strategy gives the best performance and what are the conditions that ensure such behavior.

We present a system performance analysis that is based in experimentation and use of statistics models [[5]]. For so doing, we will make a design of experiments. As ever, the goal of this design will be to obtain the maxim of information with the minim quantity of experiments. The analysis of such experiments will lead us to distinguish the effects of factors that may inside in the system performance.

2. EVALUATION MECHANISMS

We consider the SQLf querying structure: *select t A from R group by A having Q are fc*. Being *t* a threshold associated with the query, *A* an attribute (attribute list) of *R* relation (relation list), *Q* a fuzzy quantifier, and *fc* a fuzzy condition. This query returns the fuzzy relation *Rf* on $\{a / (\exists x \in R / x.A=a) \wedge (\mu(Q(Xa,fc)) \geq t) \}$, being the membership degree of each element *a*: $\mu Rf(a) = \mu(Q(Xa,fc))$ (the truth degree of fuzzy quantified sentence *Q Xa's are fc)*, where $Xa = \{x \in R / x.A=a\}$. The sentence *Q Xa's are fc* is interpreted with the Yager's decomposition [[7]] interpretation. For example we may address a query to the employee relation of Table 1:

Table 1. Extension of EMP(#emp, e-name, salary, job, age, #dep)

#emp	e-name	Salary	Job	Age	#dep
10	Martin	2000	K1	40	1
22	Calvin	1000	K4	38	1
78	Luther	1500	K2	50	1
41	Johnson	1200	K3	40	2
35	Smith	1000	K3	39	2
90	Peters	1200	K2	41	2
56	Anderson	1500	K2	40	3
82	Dobson	1000	K4	36	3
64	Mc Dowell	2000	K1	50	3

If we want to f*ind the departments where most of the employees are about 40 years* we may use the SQLf query Ψ:

select 0.5 #dep from emp group by #dep having most_of age = about40.

Being *most_of* and *about40* the fuzzy defined in Fig. 1.

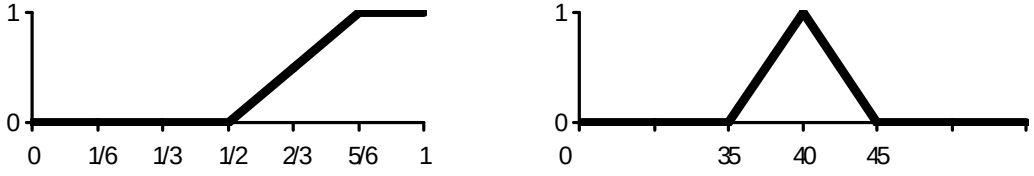

Fig. 1. Fuzzy Terms Membership Functions. At left, the proportional increasing quantifier *most_of*. At the right side, the fuzzy predicate *about40* that represents ages around 40 years old

According to the intended semantics of fuzzy quantified queries, we may compute the solution of the example query as Table 2 showns.

Table 2. Computation of Ψ query solution.

#dep	I	#eMp	age	$\mu_i = {}^{i}\sup_{x \in Xa}(\mu_{fc}(x))$	$\mu_{Qi} = \mu_Q\left(\frac{i}{\lvert Xa \rvert}\right)$	$\min(\mu_{Qi}, \mu_i)$	$\mu\ (Q(Xa, fc))$
1	1	10	40	1	0	0	
	2	22	38	.6	.5	.5	
	3	78	50	0	1	0	.5
2	1	41	40	1	0	0	
	2	35	39	.8	.5	.5	
	3	90	41	.8	1	.8	.8
3	1	56	40	1	0	0	
	2	82	36	.2	.5	.2	
	3	64	50	0	1	0	.2

As the query specify an user desired threshold, the result set contains only those elements having satisfaction degree greater or equal to the threshold. We obtain finally Table 3.

Table 3. Result of Ψ query

#dep	Membership degree
1	.5
2	.8

The main idea of the studied mechanisms is to allow adding fuzzy querying capabilities on top of an existing RDBMS. Such mechanisms perform fuzzy query evaluation on the result of an underlying regular query addressed to the RDBMS. In this context a theoretical measure of mechanisms behavior is the number of accessed database rows. Anyone may think that whenever more rows are accessed more time is spent. Thus evaluation mechanisms have been proposed in order to keep low the number of accessed rows y query evaluation. Nevertheless, for so doing selection criteria may be of high complexity. Therefore it is necessary to perform an experimental study to show whether such mechanisms have or have not better performance than naïve solution.

Naïve Strategy[[1],[2]] consists in a program scanning the whole database relation and computing the satisfaction degrees. For previous example query, Naïve strategy will make all the computation shown in Table 2. We would like to avoid the whole database relation scanning. As ever, row access in expensive in time. Moreover, in fuzzy queries, satisfaction degree computation is also time consuming.

Sugeno Strategy[[1]] consists in scanning the whole table as in Naïve one, but with a halting conditions for each group. Conditions involve the minimum number of elements LB satisfying the fuzzy quantifier with a degree greater or equal than. In the example, for each group LB is N*2/3, being N the group cardinality. The failure condition when the number of group's scanned rows with satisfaction degrees under the desired threshold is greater than N-LB. In this case remaining rows in the group are ignored and the grouping attribute value is not part of the solution. In the example only the row of #emp=56 could be avoid in computation. We can see that this strategy avoids some rows access. The benefit of this technique is that it does not use any complex selection criteria. On the other hand the number of accessed rows depends of rows retrieving ordering. If fact, in worst case all rows are accessed despite the group doest not meet the threshold for satisfaction degree.

Derivation Strategy[[6]] consists in scanning the rows selected by a regular query intended for retrieving only those rows whose satisfaction degree is greater or equal to the satisfaction threshold and is member of a group containing at least LB such rows. This strategy avoids extra computation. In the example, only the rows of #emp=10, 22, 41, 35 and 90 are scanned. This strategy only accesses rows that are really relevant for satisfaction degree computation of groups that do meet the threshold. Regular query addressed to the RDBMS contains derived crisp selection criteria. The advantage of using such criteria is that we may think that RDBMS will make use of any optimization mechanism inside it. Nevertheless such criteria may be of high complexity. It may have a bad influence on total spent time.

Due to difference presents in these strategies for fuzzy query evaluation, it is an open issue their behavior in term of total spent time. As we have mentioned before, it is the main contribution of work presented in this paper.

3. EXPERIMENTS' DESIGN

The performance evaluation will be made using formal model statistic method. The idea of this method is to obtain a model that explains the influence of several considered factors in the observed values from experiments. For this kind of study we must establish the data that will be used for the experiment, the answer variables that will be measured and the factors that will be taken in account. Furthermore, we must propose the different experiments that will be performed. These experiments are determined by the different considered factors and the different levels for each factor. Finally, an initial model must be proposed in order to make the statistic analysis with the experimental results.

The queries will be addressed to a database relation. This relation will contain the experimental data for our study. Therefore we must define the scheme and the extension of this database relation.

We don't want to use a complex relation structure; rather we will propose a minimal structure. We will define the relation scheme with one attribute that may be used as primary key, another attribute susceptible of a fuzzy treatment and finally an attribute that may be used for grouping. With these criteria we define the relation *employee(identification_card,age,departament_code)*.

The relation extension is generated as follows: Values for identification_card are sequentially generated numbers. Values for age are uniform random generated numbers between 18 and 65. Values for deparment_code are uniform random generated numbers between 1 and 7.

The variables that are observed in the experiments are called answer variables. They usually are measures of the system behavior. In our case, we observe the time. The time of response refers to the total expend time of processing the fuzzy query. In other words is the time that the user waits for the complete system answer. This time is measured by the SQLf system prototype.

The number of possible experimental studies is infinite. Therefore we must fix some conditions in our study in order to limit it. However, we must ensure that the experimental study to be as general or representative as possible. In our case of study we will fix all query parameters except the fuzzy quantifier selectivity. We will use a query like: *select 0.5 departament_code from employee group by department_code having <fuzzy_quantifier> are age=Around40.*

We choose 0.5 as threshold. This chose allows us to take no care of calibration as a factor due to the following reasons: Fixed a value for the calibration, the selectivity of the query will be determined by the used fuzzy predicate and quantifier. The extreme values 0 and 1 lake of sense for selection, the level 0 is equivalent to do not establish a calibration, and the level 1 is equivalent to perform a regular query. The use of a middle value is imposed, we prefer 0.5 because it is the hope of a uniform variable into the unit interval.

In order to isolate the problem of strategy performance for quantifiers, we use a single fuzzy predicate. We also fix the fuzzy predicate as Around40 in Fig. 2.

Fuzzy quantifiers are classified in six categories. The classification obeys to the fuzzy quantifier's nature (absolute or proportional) and its membership function behavior (increasing, decreasing or unimodal). Nevertheless, fuzzy quantified sentences using any kind of quantifier may be transformed into sentences using only increasing proportional quantifiers. This argument has been used in previous works in order to simplify the study of fuzzy quantified sentences.

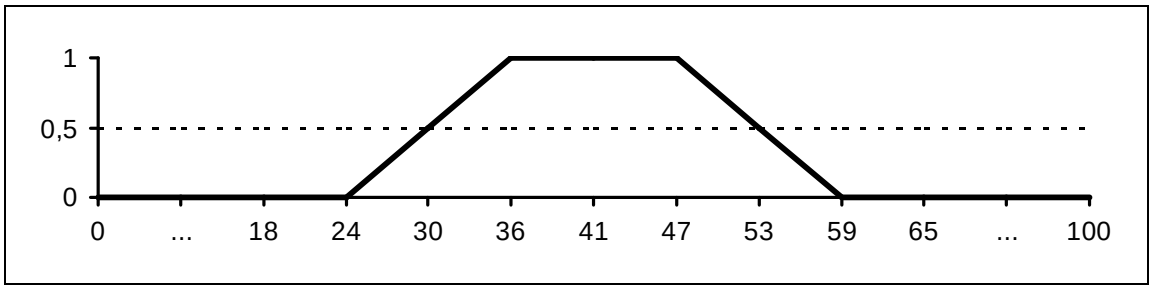

Fig. 2. Around40 Fuzzy Predicate. This user definition is not centered in 40. It was chosen for convenience of equal number of elements in the 0.5-cut and out of it for experimental data.

The experimental factors are those variables whose values are changed in the experiments in order to determine their effect in the answer variable. Factors may take different values. For the experiments' design it is necessary to choose some possible values than will be used for each factor. These chosen values are called the factors' levels. The combination of factors and levels used in the experiments will determine the used kind of design. Factors and levels of our study are described hereafter.

The objective of this study is to establish a comparison of the proposed strategies based in experimental results. We expect this factor to have a high influence in the performance of the query evaluation. Its levels are: Naive Strategy, Derivation Strategy and Sugeno Strategy.
As ever, the volume of data is a factor that must be considered. Despite in the different strategies the proportion of accessed registers does not depends on how large is the database, it is reasonable to think that an interaction might exists between the volume factor and the strategy factor. We define three levels for the volume factor. They are: low (1000 rows), middle (10000 rows), and high (100000 rows).
As we have said before, we have restricted our experiments to increasing proportional quantifiers. We define three quantifiers to be used in the experiments, they are represented in Fig. 3. These quantifiers will establish the three levels of the quantifier factor. We think that these three levels are representative of the different scenarios of selection imposed by the fuzzy quantifiers.

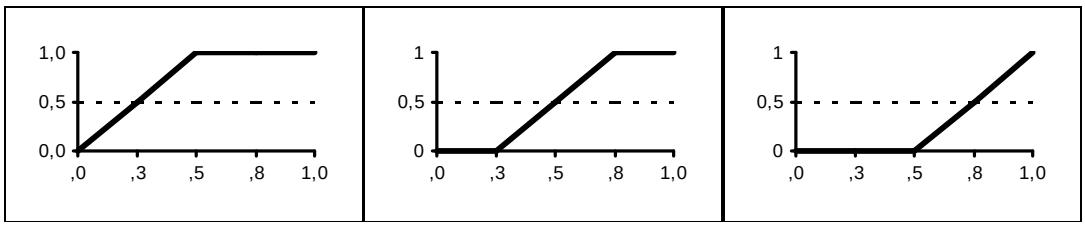

Fig. 3. Fuzzy Quantifiers Membership Functions. Left: *HalfOf* quantifier, the least selective. Center: *MostOf* quantifier, the middle in selectivity level. Right: *QuasiAll* Quantifier the most selective of considered quantifiers.

We have chosen a full factorial design for our experimental study. That is, we will consider all the mentioned factors and all their levels. This kind of design allows study the influence of each factor and all theirs interactions.

Table 4. Sumary of Factors. Each factor is denoted by a Symbol. Also shows factors' levels.

F	Symbol	Name	lf	Level 1	Level 2	Level 3
1	E	Strategy	3	Naïve	Derivation	Sugeno
2	V	Volume	3	Low	Middle	High
3	Q	Quantifier	3	Half Of	Most Of	Quasi All

The proposed experimental model for our study is:

$$y_{ijkl} = \begin{pmatrix} y_{\dots} \\ + E_i + V_j + I_k + Q_l \\ + EV_{ij} + EI_{ik} + EQ_{il} + VI_{jk} + VQ_{jl} + IQ_{kl} \\ + EVI_{ijk} + EVQ_{ijl} + EIQ_{ikl} + VIQ_{jkl} \\ + EVIQ_{ijkl} \end{pmatrix}$$

Being:

- y_{ijk} the observed value for levels i, j, k of factors E, V and Q, respectively.
- $y_{\dots}$ the arithmetic mean of the observed values of all experiments.
- F_m the effect of the factor F at the level m, $F \in \{E,V,Q\}$.
- $F'F''_{m'm''}$ the effect of the Interaction between factors F' and F'' at the levels m' and m'', respectively, with different $F',F'' \in \{E,V,Q\}$.
- EVQ_{ijk} the effect of the Interaction between all the factors E, V and Q for the levels i, j and k respectively.

4. Experimental Results

We have performed the experiments with the design presented in pervious chapter. The fuzzy queries where addressed to a SQLf prototype that we have developed [[4]] on top of Oracle 8i DBMS. This prototype allows the use of any of the three evaluation mechanisms. The prototype computes the total spent time for the evaluation of the

fuzzy query. We use (in dedicated mode) a SUN Enterprise 450 architecture server of two 250 MHz. processors with 512MB RAM, four 4GB SCSI hard disks and Solaris 8.
With the same environment conditions, we have run three times the experiments, obtaining similar results. We have made a test of standard deviation of these three replicas, showing that is not relevant to consider replicas in the model. Therefore, we only present here results of one replica. As resulting times difference is very high, we normalize them applying the logarithmic transformation: $y_{ijk} = \ln\left(\tau_{ijk} + 1\right)$, being τ_{ijk} the observed times.

Table 5. Logarithm Transformed Times. Experimental times have been normalized. We can see that transformation gives values of no more than 1 magnitude order of difference.

Strategy↓	Quantifier→ Volume↓	HalfOf	MostOf	QuasiAll
Naïve	Low	0.760806	0.732368	0.779325
	Middle	2.636912	2.597491	2.568788
	High	5.616662	5.550592	5.455107
Sugeno	Low	0.792993	0.625938	0.518794
	Middle	2.614472	2.618855	2.080691
	High	5.782224	5.719295	5.141488
Derivation	Low	0.431782	0.285179	0
	Middle	2.102914	1.759581	0.009950
	High	5.355642	4.011506	0

We use R Statistical Software for the analysis of variance. We introduce in this tool the normalized experimental. Thereafter, we specify the model and inspect it.
The analysis is made with the statistical F distribution proof. We may observe in Table 6 (ANOVA Table) that: All factors, Strategy, Volume and Quantifier and their interactions have a high influence in the behavior of the observed value. In these cases the computed F values are very close to the tabled F values (they are '***' marked). It tells us that we may conclude about the answer variable behavior respects to the factors and its interactions, with statistics certainty.

Table 6. Analysis of Variance Full Factorial Model. Asterisk marks in rows denote the relevance of the factor or factor interaction in explaining the experimental results.

	Df	Sum Sq	Mean Sq	F value		Pr(>F)	
Strategy	2	33.946	16.973	1,32E+36	<	2.2e-16	***
Volume	2	242.048	121.024	9,44E+36	<	2.2e-16	***
Quantifier	2	16.645	8.322	6,49E+35	<	2.2e-16	***
Strategy:Volume	4	11.765	2.941	2,29E+35	<	2.2e-16	***
Strategy:Quantifier	4	18.601	4.650	3,63E+35	<	2.2e-16	***
Volume:Quantifier	4	8.528	2.132	1,66E+35	<	2.2e-16	***
Strategy:Volume:Quantifier	8	12.145	1.518	1,18E+35	<	2.2e-16	***
Residuals	54	6,93E-27	1,28E-28				

Signif. codes: 0 ` ***' 0.001 ` **' 0.01 ` *' 0.05 `.' 0.1 ` ' 1							

In order to show the influence of factors, we plot the observed results as functions of each couple of relevant factors. Remember that we have made a logarithmic transformation of the observed data. The graphics forms will lead us to understand the influence of factors. We present only interaction with Strategy factor because it is the main interest of this study.
Influence in response time of Strategy and Volume interaction plot is shown in Fig. 4- We may observe there that the behavior of times for any strategy is increasing respect to the growth of the data volume. This result is not surprising at all; we expected this influence of the volume factor as ever. The volume factor is very important in the explanation of the studied performance.
On the other hand, we may remark the quasi-equal graphics of Naïve and Sugeno strategies; it is little the benefit observed of Sugeno strategy respect the Naive one. Nevertheless, we may note a high benefit of using the Derivation

strategy respect the other ones; the values for the Derivation strategy are lower than values for either Sugeno or Naive strategies. This tells us that the strategy factor is definitively influencing the performance of query evaluation. Moreover, this leads us to affirm that the Derivation strategy ensures the better performance for the query evaluation.

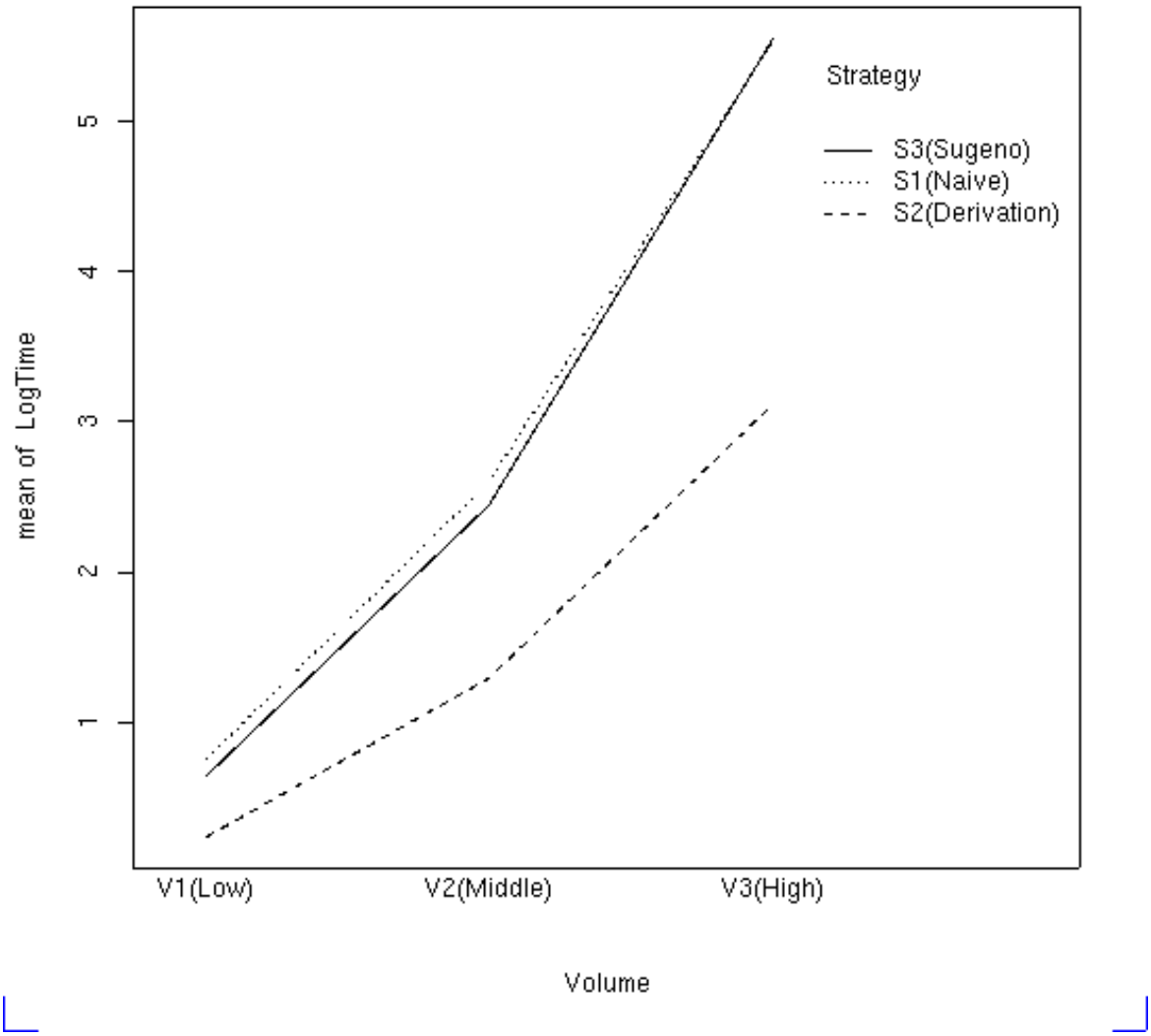

Fig. 4. Influence in Response Time of Strategy and Volume Interaction.

In Fig. 5 plot, corresponding to influence of Strategy and Quantifier interaction, once again we observe that the Derivation strategy presents better performance than Sugeno and Naive ones. In this interaction the behavior of the Sugeno strategy is different to the behavior of the naive Strategy. In case of a quantifier that imposes a more strict selection condition, Sugeno strategy has better performance than Naive one. The mean time for the Naive strategy stays constant no matter the used quantifier. It obeys to the fact that the Naive strategy takes no advantage from the fuzzy selection conditions. There is a clear influence of the Quantifier in answer time, it is evidenced by the graphics for the Sugeno and the Derivation strategies. Interaction of Strategy and Quantifier has the higher significance in explaining query spent time according to Analysis of Variance (Table 6).

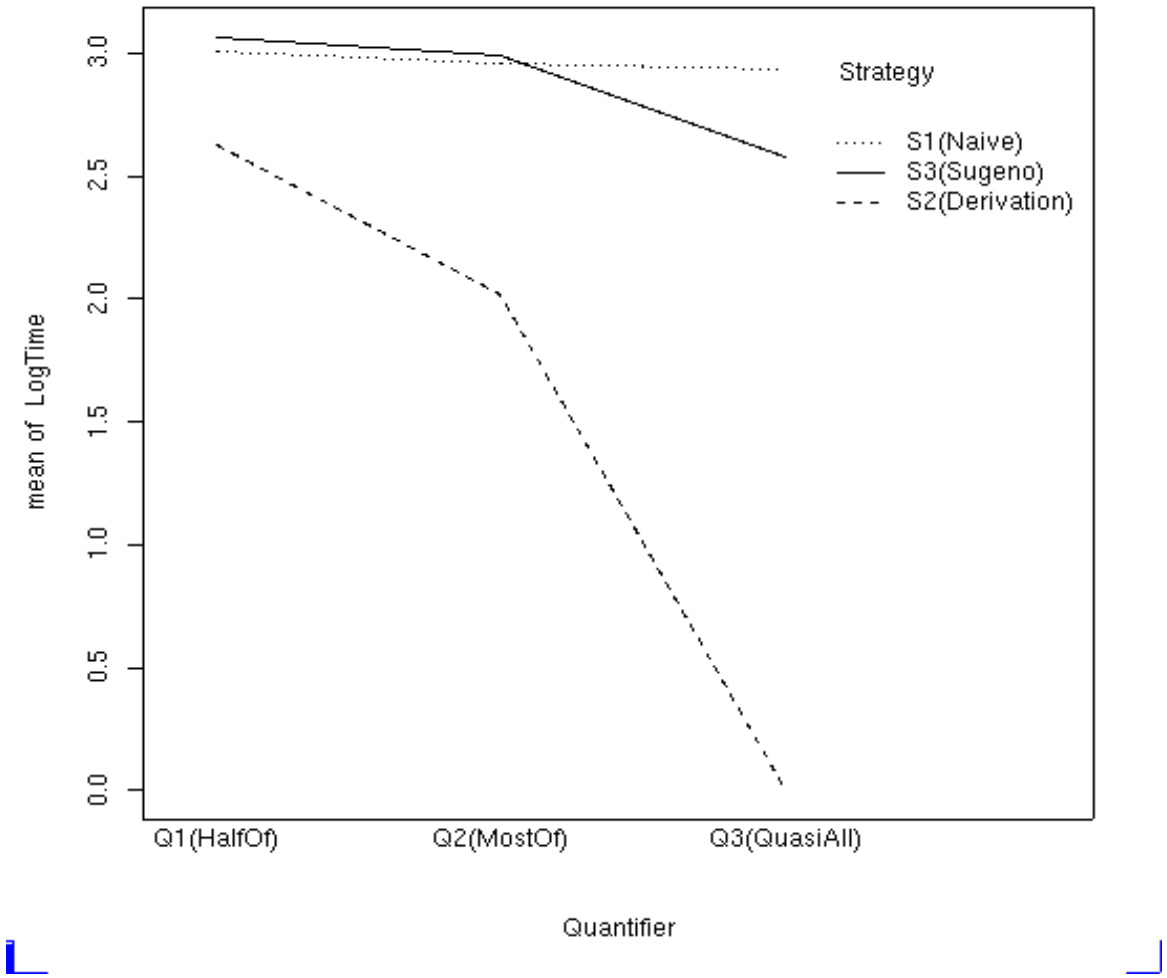

Fig. 5. Influence in Response Time of Strategy and Quantifier Interaction.

5. CONCLUDING REMARKS

We have made formal performance study of query evaluation mechanisms: Naïve, Sugeno and Derivation. In the study we have observed as answer variable the total spent time of the evaluation algorithm. This time is given by a SQLf prototype.

We have limited the study to partitioned queries. This kind of query is representative enough for all fuzzy quantified queries because any query involving a fuzzy quantifier may be transformed into this structure. We choose the level 0.5 as threshold for the calibration of the answers. Fixed a value for the calibration, the selectivity of the query will be determined by the used fuzzy predicate and quantifier. We prefer 0.5 because it is the hope of a uniform variable into the unit interval. We have also fixed the predicate under the fuzzy quantifier scope as a single predicate defined by a trapezium. Fuzzy quantified sentences using any kind of quantifier may be transformed into sentences using only increasing proportional quantifiers. Therefore we restrict the study to this kind of quantifiers.

We have chosen a full factorial design with the factors: E: Strategy (Naive, Derivation, Sugeno); V: Volume (Low, Middle, High); and Q: Quantifier (HalfOf, MostOf, QuasiAll). The analysis leads us to conclude that all these factors and interaction are very significant in explaining observed times.

As ever, the factor of higher influence in the performance is the Volume of data. The factor with a second high influence is the Strategy. It confirms the great importance of the strategy choice in query evaluation.

We must remark it is little the benefit observed of Sugeno strategy respect the Naive one. Nevertheless, we may note a high benefit of using the Derivation strategy respect the other ones. The relevance of this factor and its interactions with others considered factors leads us to conclude that the use of the Derivation strategy guaranties the best performance.

We work now in proposing new evaluation methods for fuzzy quantified queries on top of a RDBMS. We are also performing tests as that presented here for others fuzzy quantified querying structures. In further works it is possible to study the problem of fuzzy querying with acceleration structures such as indexes and also combine these evaluation methods with regular query optimization techniques. Another topic of further works interest is the problem of the interpretation of sentences of form *Q B X's are A*, its application to database querying in the context of SQLf, the evaluation mechanisms for queries involving this kind of sentences and the performance study of such mechanisms

References

[1] Bosc P., Liétard L., Pivert O., "Evaluation of Flexible Queries: The Quantified Statement Case", Proceedings of the 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'2000, Madrid, España, (2000), Pp. 1115-1122.

[2] Bosc P., Pivert O., "SQLf query functionality on top of a regular relational DBMS", in: Knowledge Management in Fuzzy Databases, O. Pons, M.A. Vila, and J. Kacprzyk (Eds.), Heidelberg: Physica-Verlag, to appear.

[3] Liétard L., "Contribution à l' interrogation Flexible de Bases de Données: étude des propositions quantifiées floues" Thèse de Docteur Université de Rennes. (1995)

[4] López Y., "Evaluación de Estrategias para Consultas Cuantificadas en SQLf", Informe final de proyecto de grado en la Universidad Simón Bolívar, abril 2002.

[5] Raj J., "The Art of Computer Systems Performance", John Wiley/Sons, Inc, 1991.

[6] Tineo L., "Extending the power of RDBMS for Allowing Fuzzy Quantified Queries". Lecture Notes in Computer Sciences, September 2000. Vol. 1873, pp 407-416.

[7] Yager, R., Interpreting Linguistically Quantified Propositions, International Journal of Intelligent Systems, Vol. 9, (1994), Pp. 541-569.