Chapter XI
FSQL and SQLf: Towards a Standard in Fuzzy Databases

Angélica Urrutia
Universidad Católica del Maule, Chile

Leonid Tineo
Universidad Simón Bolívar, Venezuela

Claudia Gonzalez
Universidad Simón Bolívar, Venezuela

ABSTRACT
Actually, FSQL and SQLf are the main fuzzy logic based proposed extensions to SQL. It would be very interesting to integrate them with a standard for fuzzy databases. The issue is what to take from one or other proposal. In this chapter, we analyze FSQL and SQLf making a comparison in several ways: approach direction, fuzzy components, system architecture, satisfaction degree, evaluation mechanisms, and experimental performance. We observe that there are powerful and interesting features in both proposals that could be mixed in a unified language for fuzzy relational databases.

INTRODUCTION
In order to give greater flexibility to relational database management systems (RDBMS), different languages and models have been conceived with the incorporation of fuzzy logic concepts into information treatment. Two outstanding proposals in fuzzy logic application to databases are those of FSQL (Galindo, 1999, 2007) and SQLf (Bosc & Pivert, 1995a). This chapter shows a comparison on these two applications from different points of view.

FSQL was created in order to allow the treatment of the uncertainty in fuzzy RDBMS. It allows the representation and manipulation of precise and vague data. It distinguishes three data categories: crisp, referential ordered, and referential not ordered. It uses possibility distributions and similarity relations for the representation of vague data, using the model GEFRED (Medina, 1994; Medina, Pons, & Vila, 1994). For the manipulation of these data, FSQL extends some components of SQL with elements of fuzzy logic. It includes the use of possibility and necessity measures. Surroundings to
FSQL and SQLf

FSQL have been conceived as a catalogue named FMB to represent vague data and linguistic terms in a relational database. Additionally, FuzzyEER, an extension of the EER model (Extended Entity-Relationship), has been conceived to allow the conceptual design of databases that incorporate vague data (Urrutia, 2003; Urrutia, Galindo, Jiménez, & Piattini, 2006; Urrutia, Galindo, & Piattini, 2002). A mechanism for the translation of a conceptual scheme in FuzzyEER to FMB has been settled (Galindo, Urrutia, & Piattini, 2006). There exist two known implementations of FSQL at the present time, one in Oracle (Galindo, 1999, 2007) and the other in PostgreSQL (Maraboli & Abarzua, 2006).

SQLf was conceived in order to represent vague requirements in queries to relational databases. It includes extensions based on fuzzy logic for all the elements of the SQL standards until the SQL3. In this language, query conditions may involve diverse linguistic user defined terms that are specified through an extension of the DDL. SQLf allows fuzzy queries over precise data, producing discriminated answers. That is to say, each row in the answer has associated its satisfaction degree of the vague requirement represented by the query. In order to evaluate queries in SQLf, it has been proposed to take advantage of the existing connections between the fuzzy and classic sets. From a fuzzy query, the principle of the derivation allows obtaining a derived precise query. The processing of the fuzzy query is made on the result of the derived consultation. There are two known SQLf implementations, both on Oracle.

The comparison made in this work is related with the following aspects: variety in the use of fuzzy logic elements; satisfaction degree semantics of the answer set; evaluation mechanisms for query processing; proposed architectures for the implementation and performance experimental analysis with current prototypes. With the research work presented in this chapter, we open the way for the integration of FSQL and SQLf towards a new standard for fuzziness treatment in databases.

This chapter has been organized as follows:

The next section gives a basic background on fuzzy sets. You can read more about this in the first chapter of this handbook. In the following section, we present the approach directions of FSQL and SQLf prior to pointing out the fuzzy components of both languages. Then, we give a general view of the architecture in the SQLf and FSQL implementations. We also explain in one section the use of satisfaction of fulfillment degrees in these languages. Evaluation mechanisms for fuzzy queries, according to the two proposals, are discussed, along with experimental performance analysis of existing prototypes. Finally, we address some conclusions and future trends of this work.

FUZZY SETS BACKGROUND

Fuzzy sets were introduced in Zadeh (1965) to model fuzzy classes in control systems, and their use has been expanded to different domains: mathematics, classification, pattern matching, artificial intelligence, and so forth. In the first chapter of this volume, Galindo introduces fuzzy logic and fuzzy databases. See also the overview chapter by Kacprzyk, Zadrozny, De Tré, and De Caluwe in this book about fuzzy approaches to flexible database querying.

Fuzzy Sets

The fuzzy sets theory stems from the classic set theory, adding a membership function to the elements of the set, which is defined in a way that each element is assigned a real number between 0 and 1 (Zadeh, 1965, 1978). A fuzzy set \( A \) over the universe of discourse \( U \), is defined by means of a membership function \( \mu_u : U \rightarrow [0,1] \). This function indicates the degree to which the element \( u \) is included in the concept represented by the fuzzy set. The degree 0 means that the element is completely excluded of the set, while the degree 1 means that it is completely included. It is also possible to represent a fuzzy set with a set of pairs.