Chapter VIII
Customizable Flexible Querying in Classical Relational Databases

Gloria Bordogna
CNR IDPA, Italy

Guiseppe Psaila
University of Bergamo, Italy

ABSTRACT

In this chapter, we present the Soft-SQL project whose goal is to define a rich extension of SQL aimed at effectively exploiting flexibility offered by fuzzy sets theory to solve practical issues when querying classic relational databases. The Soft-SQL language is based on previous approaches that introduced soft conditions on tuples in the classical relational database model. We retain the main features of these approaches and focus on the need to provide tools allowing users to directly specify the context dependent semantics of soft conditions. To this end, Soft-SQL provides a command (named CREATE TERM-SET) to define the semantics of linguistic values with respect to a context represented by a linguistic variable (Zadeh, 1975); the SELECT command is extended in order to support soft predicates based on the user defined term sets, the semantics of grouping and aggregation can be modified, and finally, the clauses in the SELECT command can be combined effectively.

INTRODUCTION

The need to flexibly query relational databases has been widely recognized as a means to improve the effectiveness of the retrieval in current systems using SQL for expressing information needs. The main inadequacy of the SQL language is caused by the crisp algebra on which it is founded that does not support the ranking of the results with respect to their relevance to user needs. In this book, the chapter by Kacprzyk et al. provides an extensive survey on flexible querying approaches. For many categories of users, the possibility to express tolerant conditions and to retrieve discriminated answers in decreasing order of relevance can greatly simplify users’ tasks that generally are performed through a sequence of trial and error phases. The problem of false drops when querying databases by specifying crisp selection conditions is well known. Several approaches have been proposed either based on preference specifications or on soft conditions tolerating degrees of undersatisfaction to overcome this drawback of SQL language use (Bosc & Pivert, 1992; Dubois...
Customizable Flexible Querying

& Prade, 1997; Eduardo, Goncalves, & Tineo, 2004; Kießling, 2002, 2003; Petry, 1996; Rosado, Ribeiro, Zadrozny, & Kacprzyk, 2006; Tineo, 2000). The foundations of Kießling for preferences in databases are the basis for an intuitive valuation of search results in which it is assumed that people naturally express their requests in terms like “I like A better than B.” All these preferences can be formulated as strict partial orders. Based on this formulation, the Preference SQL language (Kießling, 2002, 2003) has been defined as an extension of SQL. Several built-in base preference types, combined with the adherence to declarative SQL programming style, guarantees great programming productivity. Further, the Preference SQL optimizer does an efficient rewriting into standard SQL.

Another approach for the specification of preferences in queries is based on soft constraints, that is, tolerant selection conditions formalized within fuzzy set theory (Zadeh, 1965). Several extensions of SQL to allow the specification of soft selection conditions in queries have been proposed. A rich taxonomy that helps in understanding the various proposals of extension of SQL by fuzzy set theory is outlined in Rosado et al. (2006).

In Dubois and Prade (1997), two reasons for using fuzzy set theory (Zadeh, 1965) to make querying more flexible are discussed. First, fuzzy sets provide a better representation of the user’s preferences. One reason is that users feel much more comfortable using linguistic terms instead of precisely specified numerical constraints when expressing in a query some condition such as when asking for some hotel “not too expensive and not too far from the beach.” Furthermore, the semantics of these linguistic terms can be exactly “d” (i.e., after Zadeh, 1999, defined as a function on the basic domain of a variable) by fuzzy sets (Zadeh, 1965) so that we can have a price definitely matching or definitely not matching the user’s request, but also a price that matches to a certain degree. The second reason is that a direct consequence of having a matching degree is that answers can be ranked according to users’ requirements. Furthermore, the possibility to “precisiate” the semantics of the linguistic terms makes it possible to implement mechanisms that offer users a full control on the semantics of their flexible queries (Bordogna & Psaila, 2005).

According to many authors (Bosc & Pivert, 1992, 1995; Kacprzyk & Zadrozny, 1995; Medina, Pons, & Vila, 1994; Petry, 1996), there are two main lines of research in the use of fuzzy set theory in the database management system (DBMS) context. The first one assumes a conventional database and, essentially, develops a flexible querying interface using fuzzy sets, possibility theory, fuzzy logic, and so forth (Bosc & Pivert, 1992, 1995; Bosc, Buckles, Petry, & Pivert, 1999; Dubois & Prade, 1997; Galindo, Medina, Cubero, & Garcia, 2000; Goncalves & Tineo, 2003, 2005; Kacprzyk, Zadrozny, & Ziolkowski, 1989; Ribeiro & Moreira, 1999; Tahani, 1977; Takahashi, 1991, 1995; Tineo, 2000). In the chapter of this book by Urrutia et al., a review of two extensions of SQL, namely FSQL (Galindo, Urrutia, & Piattini, 2006) and SQLf (Bosc & Pivert, 1995) is presented.

The second line of research uses fuzzy or possibilistic elements for developing a fuzzy database model to manage imprecise and vague data (Bosc & Prade, 1994; Umano & Fukami, 1994). Also in this case querying constitutes an important element of the model (Baldwin, Coyne, & Martin, 1993; Bosc & Pivert, 1997a, 1997b; Buckles & Petry, 1985; Buckles, Petry, & Sachar, 1986; Galindo, Medina, & Aranda, 1999; Galindo, Medina, Pons, & Cubero, 1998; Galindo et al., 2006; Prade & Testemale, 1984, 1987; Shenoi, Melton, & Fan, 1990). For a description of a fuzzy extension of SQL that works on crisp and fuzzy relations, see Galindo et al. (2006).

However, these proposals missed addressing some practical and not negligible aspects related to the effective usage of the flexible query language. Mainly, they do not exploit one of the main features offered by fuzzy set modeling, that is, the possibility to “precisiate” the semantics of linguistic terms (Zadeh, 1999) used in the flexible queries, thus not making users capable of having