Chapter V
From User Requirements to Evaluation Strategies of Flexible Queries in Databases

Noureddine Mouaddib
Université de Nantes, France

Guillaume Raschia
Université de Nantes, France

W. Amenel Voglozin
Université de Nantes, France

Laurent Ughetto
Université de Rennes 2, France

ABSTRACT

This chapter presents a discussion on fuzzy querying. It deals with the whole process of fuzzy querying, from the query formulation to its evaluation. Mainly, it advocates the use of index structures in the evaluation of fuzzy queries. First, various ways of introducing flexibility in querying processes are discussed, especially the most represented in the literature, which are based on rankings of the answers or which are using user-oriented fuzzy labels in the queries. Current methods for evaluating fuzzy queries are also reviewed. Then, properties of access methods are given in the context of fuzzy querying. Last, SaintEtiQ, the method developed in our team, is briefly presented.

INTRODUCTION

First intended to provide efficient transactional mechanisms satisfying ACID (atomicity, consistency, isolation, durability) properties, query processing has been intensively studied from the machine-machine interface point of view. As a consequence, current commercial database systems are all able to handle very high workloads with update queries and succeed in storing very large data sets. However, the rapid development of personal numerical devices combined with the widely spread high-bandwidth network connections has lead to the point where huge amounts of
data are instantaneously available to the end user. Obviously, the user asks for intelligent and modern tools to query, analyse, and browse the data, especially to support complex decision-making processes. Hence, the database community faces new challenges regarding the key technology of query processing in order to fill the gap between its current system-optimised features and the strong requirements for human-centric capabilities.

To achieve this goal, recent works try to integrate well-established IR (information retrieval) technologies into database systems. The database community also looks at all theoretical backgrounds to introduce flexibility within the query processing. One of the promising research directions is towards the fuzzy set theory. In a broader sense, the introduction of fuzzy set theory into database management systems raises new fields of studies: (1) fuzzy data models for management of uncertainty, (2) fuzzy dependencies obtained in relaxing integrity constraints, and (3) fuzzy queries to provide flexibility within the querying process. Regarding fuzzy querying, research works often deal with the definition of a new language, able to accommodate the new formalism (Bosc & Pivert, 1995; Galindo, Medina, Pons, & Cubero, 1998; Galindo, Urrutia, & Piattini, 2006; Kacprzyk & Ziółkowski, 1986). The literature also offers studies of how to express concepts or needs through constructs such as operators or linguistic variables. Although the subjects are not limited to these two examples, almost all studies are on the formal or conceptual side of the new situation. The question of the evaluation of queries has seldom been treated.

What does the phrase “evaluation of a query” encompass? The evaluation of a query refers to the whole process of finding, in the queried database, all the records that satisfy the query. Thus, the evaluation lies on a more “physical” or “technical” level than formal. For instance, the query “Find all the students who registered for at least one course this semester” may be evaluated by sequentially searching the REGISTRATION relation for entries that have the same value in student ID. Finally, if such entries exist, the student ID is returned.

With fuzzy sets, the main new element in query evaluation is the gradual aspect. The immediate consequence of graduality is that many binary concepts from the database management system (DBMS) have to be adapted.

### Need for Flexibility into Database Query Processing

Flexible querying in databases starts with the simple requirement for user-friendly interfaces to access the data. A query interface usually provides several components to formulate, evaluate, and give the answer to a query. At the end of the process, flexibility can be introduced with both the ideas of approximate answers and partial answers to queries. It allows computing quickly a rough answer rather than an exact result set that requires more time. It also provides mechanisms to deal with excessively large result sets as well as empty result sets of complex queries.

In contrast, at the very beginning of the query chain is the analysis and capture of user requirements. What are the main concepts users would like to express into a query? The database community answers the question in two ways: user preferences and user perceptions of the queried domain. It could be synthesised into the very obvious idea of personalisation of the query. Since data are no more represented according to the user’s requirements, the query should, acting as an interface between the user’s language and the one of the DBMS. Hence, declarative languages supported by databases have to provide new constructs dealing with such revolutionary concepts (user preferences and user perceptions of the domain) in regard to the past database fundaments. There is another chapter in this volume by Kacprzyk, Zadrożyń, de Tré, and de Caluwe that includes a review about flexible querying.