A Survey of SQL Language

Hock Chuan Chan, Hong Jun Lu, and Kwok Kee Wei
National University of Singapore

SQL is the standard (ANSI and ISO) query language for relational database systems. It is increasingly used widely on various platforms as the main language to access databases. Beside theoretical analyses and laboratory studies with students, there is little information on SQL usage in practice. A survey was executed to identify major SQL users and applications, users’ perception of SQL, utilization of SQL features, and errors and problems that users encounter. The main SQL users are analysts and database administrators, which together formed 70.6%. There are very few end-users. SQL is very widely used in various applications. Queries are quite evenly spread from the very simple to the very complex, with users wishing for more complex capabilities. A surprisingly large 82% of queries contain syntax or logical errors, and require multiple attempts by users. Notwithstanding these, users have very good perception of SQL.

Growing importance of data resource

In recent years, considerable attention has been given to identifying the most critical issues facing IS management. The surveys of these studies have been extended to cover both the private and the public/non-profit sectors (Caudle, et al, 1991; Moynihan, 1990). Among the top ten issues for IS managers as identified by Brancheau and Wetherbe (1987) was the need to promote effective use of data resource. Selected chief executive officers, senior functional managers and IT managers have expressed a common desire to increase the level of data sharing taking place across systems and departments (Moynihan, 1990). In another report (Niederman, et al, 1991), data-related issues have steadily gained recognition to eventually occupy the top two slots, as shown in Table 1. Furthermore, a group of IS executives in a recent survey (Grover and Goslar 1993) rated query languages as the most important technology under the class of human interface technologies.

The issue on making effective use of data resource has steadily gained importance over the years. Given its ability to leverage both labour and capital, information is now viewed as an essential factor of production. As information technology continues to penetrate the value chain, firms are reducing the level of inventory they carry and are relying on timely information for management production and distribution processes. Many firms are gathering increasing volumes of information from customers, suppliers, financial partners, and the economic environment. An intelligent combination of these disparate streams of information for purposes of making better decisions, gaining insight into potentially useful innovations, and streamlining operations without being overwhelmed by
information overload remains both a major challenge and opportunity (Frenzel 1991).

Relational concepts and SQL

The major database systems in commercial use are relational. The relational model, introduced by E.F. Codd in 1970, has evolved through a series of valuable and influential works (e.g. Codd 1982, 1990, Date 1989). The combination of simplicity and sound mathematical foundation that has made the relational model widely accepted.

In the early eighties, several relational database management systems (DBMS) became commercially available (Dieckmann, 1981). These systems were not necessarily designed for very large volumes of data or for high update transaction volumes. They were intended for applications where preplanning is difficult, and were expected to provide ease of programming and, in general, ease of use for everyone, from the database administrator to the uninitiated end user.

The relational query language SQL, like the relational model definition, has been evolving since the early seventies. It was first developed for system R at IBM Research Laboratory, San Jose, California. Although not part of relational model theory, SQL is considered by some people to be equal in importance to the relational model as a foundation for relational DBMS products (Fleming and von Halle, 1989). It was adopted by the American National Standard Institute (ANSI) in 1986 as a standard language for interacting with relational databases. Since the early eighties, SQL has been achieving acceptance as a de facto standard and has prevailed as the database language implemented in most commercial relational DBMS products. The popularity of SQL and the relational model has even prompted the makers of many non-relational database products to provide SQL as a means of access, or “front-end” to their products (Lusardi, 1988). SQL as a common relational database language enables consistency across product implementations, at least in the way that users, application developers, and, to some extent, database designers interface with the products. Using a common language allows users to deal with only one syntax for invoking those mechanisms.

Motivation

The concern for both the management of data resources and information architecture indicates that IS executives are shifting toward an increasingly data-oriented view and are looking for ways to leverage that view into increased IS effectiveness. With the ever increasing importance for more effective control, usage, and timely access to data, there arises the need for a better understanding of the means by which data is accessed and used.

Literature materials that compare SQL with other database query languages are not hard to find. These are mostly done in the laboratories with student users. Limited research has been done to study the language in terms of drawing up a classification of SQL users, SQL features commonly made used, and problems SQL users encountered. More is required to facilitate a thorough understanding of the language in actual use.

SQL Review

SQL as a query language

Let us first take a brief look at query languages in general before examining SQL constructs in greater detail. A query language is a special-purpose high-level language oriented towards the construction of queries to retrieve information from a database. It has a fairly limited number of functions usually intended to be used by people who are not professional programmers (Reisner, 1981). Traditional programming languages like FORTRAN and COBOL are inadequate interfaces between end-users and databases because the user must specify how to manipulate the database to get the desired result. Query languages allow users to specify what information is needed from a database without specifying how it is to be accessed. Samet (1981) identified several characteristics normally applicable to query languages:

- the language allows on-line and typically ad-hoc data retrieval;
- the primary users have little or no data processing experience;
- the language is more declarative;
- a query often consists of one statement;
- data entry is limited to a single record or field;
- calculations and output formats are limited in complexity;
- development speed is more important than run-time efficiency; and
- output is limited to a few lines of record.

Like other computer languages, SQL uses words and symbols to construct sentences which express ideas. The number of sentences which it may form and, therefore, the number of ideas which it may express, is without bound (Date, 1987). SQL is extremely limited in its verbs. There are very few types of action that it can perform. The verbs may be grouped into four categories (Lusardi 1988): Query: SELECT; Data Manipulation Language (DML): UPDATE, IN-
Related Content

Active Video Annotation: To Minimize Human Effort
www.igi-global.com/chapter/active-video-annotation/28839?camid=4v1a

Common Sense Reasoning in Automated Database Design: An Empirical Test
Veda C. Storey, Robert C. Goldstein and Jason Ding (2002). Journal of Database Management (pp. 3-14).
www.igi-global.com/article/common-sense-reasoning-automated-database/3272?camid=4v1a

Towards a Fuzzy Object-Relational Database Model
www.igi-global.com/chapter/towards-fuzzy-object-relational-database/20363?camid=4v1a

RORIB: An Economic and Efficient Solution for Real-Time Online Remote Information Backup
Scott J. Lloyd, Joan Peckham, Qing Yang and Jian Li (2003). Journal of Database Management (pp. 56-73).
www.igi-global.com/article/rorib-economic-efficient-solution-real/3299?camid=4v1a