

An Overview of the Emerging Third-Generation SQL Standard

Jim Melton and Nelson Mendoca Mattos

The emerging new generation of the SQL Standard, called SQL3, have been under development by ANSI X3H2 and ISO/IEC JTC1/SC21 over the last 4 years. SQL3 is a multi-volume, upward-compatible extension to SQL-92 containing a large number of extensions over its predecessor. These new capabilities are specified in separate, independent volumes that are expected to start emerging as new SQL standards this year

A Call-Level Interface described in the first SQL3 volume (SQL/CLI) to be published (likely this year) is a new way to bind applications to an SQL implementation. Since it does not require precompilation of application programs containing SQL calls, it is especially suitable for providers of shrink-wrap database applications

Procedural extensions, specified in another emerging SQL3 volume (SQL/PSM), targeted for completion in 1996, include SQL statements to define stored procedures and user-defined functions. These procedures and functions, called routines, can be written purely in SQL, as the language has been extended with procedural constructs found in modern programming languages — such as statements to support control flow (*e.g.*, IF/THEN/ELSE, WHILE and DO/UNTIL loops, CASE statements, *etc.*), to group multiple statements in BEGIN/END blocks where local objects (including variables, tables, cursors, routines, *etc.*) can be declared, to define exception handlers, *etc.* However, the language used to implement such routines is not restricted to SQL. Since it is expected that some users will prefer to use an established programming language and development environment, SQL3 also allows routines to be written in an external language (C, Fortran, PL/I, COBOL, ADA, PASCAL, or MUMPS) Users can create function libraries or attach existing function libraries to an SQL implementation, allowing applications to share these functions via SQL. They can be used in any SQL expression, allowing queries to contain powerful computation and search predicates to filter irrelevant data close to the data source. Both functions and procedures support full overloading and a PATH mechanism (similar those commonly found in operating systems). Resolution of overloaded routines is performed based on the current PATH and a non-selfish algorithm in which all argument types in the invocation are taken into account.

An extensible, object-oriented type system defined around the notion of abstract data types (ADTs) is described in the major volume of SQL3 (SQL/Foundation), which is expected for publication in 1997. ADTs are, like classes in object-oriented programming languages, used to capture application-specific

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association of Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
SIGMOD '95, San Jose, CA USA
© 1995 ACM 0-89791-731-6/95/0005..\$3.50

semantics and behavior in the database. They are completely encapsulated, can be related to other ADTs in subtype-supertype relationships, and inherit the attributes of their (possibly multiple) supertypes. Substitutability is also supported, *i.e.*, subtype instances can be substituted wherever instances of supertypes are expected (including columns of tables, query results, arguments and return values of functions, *etc.*) Dynamic binding is provided, but type checking is always performed at compile time. In addition to ADTs, SQL3 supports the definition of row types (the type of rows in tables), distinct types (a mechanism to create a user-defined type based on an existing type), and type templates. Collection types (SET, LIST) are also provided as built-in type constructors. Users can exploit these types to represent complex data structures (*e.g.*, nested tables can be modeled as collections of rows)

This volume of the SQL3 draft standard also contains important relational extensions, such as the ability to write recursive queries (*e.g.*, bill of material), sharable query expressions, cursor enhancements, and improvements in view updatability. It also provides mechanisms for active database support such as triggers and extensions to integrity constraints.

It is the belief of the ANSI X3H2 Technical Committee and the corresponding ISO committee that the emerging, new generation of the SQL standard will offer extensive support for so-called non-traditional applications in addition to improved support for traditional ones.