Rory McGann

SQL Version Analysis

Structured Query Language, or SQL, is a powerful tool for interacting with and utilizing databases through the use of relational algebra and calculus, allowing for efficient and effective manipulation and analysis of data within databases. There have been many revisions of SQL, some minor and others major, since its standardization by ANSI in 1986, and in this paper I will discuss several of the changes that led to improved usefulness of the language.

In 1970, Dr. E. F. Codd published a paper in the Association of Computer Machinery titled *A Relational Model of Data for Large shared Data Banks*, which detailed a model for Relational database Management systems (RDBMS) [1]. In order to make use of this model, a language was needed to manage the data stored in these RDBMSs. In the early 1970's SQL was developed by Donald Chamberlin and Raymond Boyce at IBM, accomplishing this goal. In 1986 SQL was standardized by the American National Standards Institute as SQL-86 and also by The International Organization for Standardization in 1987. The structure of SQL-86 was largely similar to SQL as we know it today with functionality being implemented though Data Manipulation Language (DML), which defines verbs such as *select*, *insert into*, *update*, and *delete* that are used to query or change the contents of a database. SQL-86 defined two ways to process a DML, direct processing where actual SQL commands are used, and embedded SQL where SQL statements are embedded within programs written in other languages. SQL-86 supported Cobol, Fortran, Pascal and PL/1. One of the major criticisms of SQL-86 was that it was not orthogonal. Orthogonallity means that independent concepts are kept independent and not mixed together in confusing ways [2].

The next version of SQL was released in 1989, and is known as SQL-89. It was a minor revision in which a couple of useful features were added. Two more languages were made available for SQL embedding, C and ADA. This was the first version of SQL where Data Definition Language (DDL) commands were added which changes the structure, or schema, of a database. The commands that were added were *create table*, *create view*, and *grant privileges*, mostly relating to creating views for databases. This allowed for increased security, as limiting someone's view to only the parts they need will help protect the database**.** The most important features added in this release were related to integrity checks. This included defining *primary keys*, *check constraints* and *referential integrity*. A primary keys lets you uniquely identify an entity in a table, check constraints lets you limit the value range that can be placed in a column, and referential integrity ensures that relationships between tables remains consistent. All of these features are useful because they make sure that the values in your database are correct.

The next revision was in 1992, and is known as SQL-92 or SQL 2. This was a major revision, and added more features to the language. The most important updates were more features related to DDL, data types, orthogonallity and set operations. These DDL features included *alter* and *drop* commands, both powerful schema altering tools. New data types were added, such as *date* and *time*. This was useful because the comparing of dates and times became very quick. Orthogonallity means that concepts that make the language easier to understand were added, such as the ability to use sub queries whenever expressions are allowed. Sub queries were also made a lot more useful. Sub queries were now allowed to return a tuple instead of being forced to return a single column. Some new set operations were also added, including natural join, set difference and set intersection. These operations are beneficial in querying the database, which allows for output that is more useful. It is also important to note that SQL-92 attempts to be backwards compatible with early versions of SQL, something that all later versions of SQL also try to do.

The next revision of SQL was in 1999, and is named SQL:1999, or SQL 3. This revision added new data types, predicates, recursive queries, triggers and objects**.** The most significant of the new data types are the *Boolean* and the *Array*. An array allows for the storage of multiple values in one column of a database. Predicates allow for the use of regular expressions which is a very powerful tool for matching strings. Recursive queries are exactly as they sound, and they allow you to recursively call initiate queries. The most major change in SQL:1999 was the addition of *object orientation*. This was done mostly through the structures user defined type and objects. These user defined types have a number of characteristics, the most important of which are that they have one or more attributes, their behavior is defined through methods and functions, their attributes are encapsulated, and their comparisons are done through user defined functions. SQL objects are implemented though being “typed tables”, which means that the tables’ columns are structured types [3].

The next revision of SQL was in 2003 and it known as SQL:2003. The new features that were added in this revision include built-in scalar functions, the merge statement, the sequence generator, and support for Extensible Markup Language (XML). Scalar functions are functions such as power() that return a scalar value. This is very useful because as SQL allows for orthogonallity, which means that you can use these scalar functions in place of a scalar. The merge statement is a powerful command that is basically a combination of insert and update. It allows you insert into a table, but if a column is matched, it updates instead. A sequence generator is a mechanism that allows for automatic generation of sequential values which can be used to create primary key values. Finally XML is a format which is important because of its usefulness over the internet. It is widely used for the representation of data structures over the internet.

The next revision of SQL was in 2006, and is known as SQL:2006. The main changes in this revision were improvements in the XML support. It mostly had to do with defining how SQL can be used with XML, such as importing and storing XML data in an SQL database. It also allows for integration of XQuery into SQL code.

The next revision of SQL was in 2008, and is known as SQL:2008. The major changes in this version include changing merge, the truncation table, and more triggers. The merge statement was released in SQL:2003, and it was updated in this revision. The truncation table statement is a very useful command that quickly removes all the data from a table. It is useful for when you need to quickly delete all the elements from a table; the only downside is that it bypasses integrity enforcing mechanisms. SQL:2008 also adds more triggers, such as the *instead of* trigger. Triggers are useful because they activate when a certain event happens.

The most current revision of SQL in 2011, and is known as SQL:2011. The major new features are improved temporal database support. A temporal database is one that has support for handling data that involves time. It is different than a normal database, because normally the data in a database is only assumed to be valid now, whereas in a temporal database the data could be valid in the past, present or future. Two important concepts of temporal databases are the valid time, and the transaction time. The valid time is the period of time that the data is valid, for example if a database is storing information about the 18th century, the valid time would be somewhere between 1700-1799. The transaction time on the other hand would start when we entered the data, for example June 12, 2014. This is an important addition to SQL because temporal databases are very useful when you need to either store data that it no longer valid, or constantly changes with respect to time.

[1] Oracle (2003) *History of SQL* [online]. Available: http://docs.oracle.com/cd/B12037_01/server.101/b10759/intro001.htm

[2] Chris Date, in *An Introduction to Database Systems*, 8th ed, Addison Wesley, 2003, p230

[3] Eisenberg, A., and Melton, J., “SQL:1999, formerly known as SQL3”, ACM SIGMOD Record 28(l), March 1999.