A BRIEF HISTORY OF THE SQL STANDARDS

Keith W. Hare
JCC Consulting, Inc.

October 05, 2015

Abstract

The SQL eco-system contains a rich set of features for managing and manipulating data. In the rush to new technologies such as NoSQL and Big Data, important and useful capabilities are being overlooked, discarded, rejected, and re-invented. Reviewing the features in the SQL eco-system provides a basis for understanding what is new in the Big Data paradigm.

Introduction

- SQL Standards Process
- What is SQL?
  - Transactions
  - Data Manipulation Language
  - Data Definition Language
  - SQL Interfaces
- When did these features appear in the SQL Standard?
  - SQL-87
  - SQL-89
  - SQL-92
  - SQL:1999
  - SQL:2003
  - SQL:2008
  - SQL:2011
  - SQL:201x
  - SQL Technical Reports
- SQL Futures
- Why should you care?
- Summary

Who am I?

- Senior Consultant with JCC Consulting, Inc. since 1985
  - High performance database systems
  - Replicating data between database systems
- SQL Standards committees since 1988
  - Convenor, ISO/IEC JTC1 SC32 WG3 since 2005
  - Vice Chair, ANSI INCITS DM32.2 since 2003
  - Vice Chair, INCITS Big Data Technical Committee since 2015
- Education
  - Muskingum College, 1980, BS in Biology and Computer Science
  - Ohio State, 1985, Masters in Computer & Information Science

SQL Standards Process

The database language committees started in the late 1970s

- US committee was ANSI X3H2, now INCITS DM32.2
  - Don Deutsch has been the chair since 1980
- International process
  - Started as the CODASYL Database Languages Rapporteur Group
  - Transitioned to ISO/IEC JTC1 SC21 (OSI network model)
  - Moved to ISO/IEC JTC1 SC32 (about 1998) as WG3
  - Len Gallagher – Convenor 1987(?) to 1996
  - Stephen Cannan – Convenor 1996 to 2005
  - Keith Hare – Convenor 2005 to present
- Editors
  - Phil Shaw (IBM) SQL-87 and SQL-89
  - Jim Melton (DEC, Sybase, Oracle) SQL-92 to present
- Continuity
  - committee leadership
  - committee participation.

Standards Hierarchy Translation

ISO/IEC JTC1 SC32 WG3

- ISO – International Organization for Standardization (l'Organisation internationale de normalisation)
- IEC – International Electrotechnical Commission
  - JTC1 – Joint Technical Committee 1 – computer related standards
    - SC32 – Sub Committee 32 – Data Management and Interchange standards
      - WG1 – eBusiness
      - WG2 – Metadata
      - **WG3 – Database Languages – SQL Standards**
      - WG4 – SQL/Multimedia and application packages

What is SQL?

- SQL is a language for defining data bases and manipulating the data in those data bases
- SQL Standard uses SQL as a name, not an acronym
  - Might stand for Standard Query Language
  - Might stand for SQL Query Language
- SQL queries are independent of how the data is actually stored – specify what data you want, not how to get it

SQL Capabilities

Important SQL capabilities include:

- ACID Transactions
- Data Manipulation Language
- Data Definition Language
- SQL Interfaces

Transactions – ACID Properties

- Atomic – All of the work in a transaction completes (commit) or none of it completes
- Consistent – A transaction transforms the database from one consistent state to another consistent state. Consistency is defined in terms of constraints.
- Isolated – The results of any changes made during a transaction are not visible until the transaction has committed.
- Durable – The results of a committed transaction survive failures

Data Manipulation Language (DML)

- Data manipulated with Select, Insert, Update, & Delete statements
- Select T1.Column1, T2.Column2 …
  From Table1, Table2 …
  Where T1.Column1 = T2.Column1 …
- Data Aggregation
- Compound statements
- Functions and Procedures

Data Definition Language

- Create Table (Column1 Datatype1, Column2 Datatype 2, …)
- Constraints to define and enforce relationships
  - Primary Key
  - Foreign Key
  - Etc.
- Triggers to respond to Insert, Update, & Delete
- Stored Modules
- Alter …
- Drop …
- Security and Access Control

SQL Interfaces

Four interfaces commonly used to access SQL databases:

- Interactive SQL command line
- SQL Embedded in programming languages
- ODBC
- JDBC

Interactive SQL command line

- Test queries
- Create tables
- Explore database
- Typing intensive
  - Some SQL client support cut and paste
  - Some SQL clients support executing scripts
- Some graphical tools that use ODBC or JDBC
  - Oracle SQL Developer
  - DBVisualizer
  - TOAD
  - Etc.

Embedded SQL

- SQL statements are embedded in a programming language
  - usually delineated by an “EXEC SQL” statement, shorthand for Execute SQL.
  - The SQL can be static, defined at edit time, or dynamic, created at runtime in a program variable and prepared.
- Pre-compiler processes the embedded SQL statements
  - Turns them into routine calls specific for the programming language and SQL implementation.
- A variation of this is to create all of the SQL as callable procedures in a separate source code module.

ODBC

- Microsoft ODBC specification is loosely related to SQL/CLI.
- Each SQL implementation provides an ODBC driver that can communicate with a specific SQL implementation.
- The call interface for the ODBC client is common across ODBC drivers.
- The communications protocol and the database server side are implementation specific.

JDBC

- Similar to ODBC but specific to the Java™ programming language.
- Each vendor provides JDBC drivers specific to an implementation. (There are also some non-vendor JDBC drivers).
- Class 4 JDBC drivers are written entirely in Java
- Can be used on any client with the appropriate Java Runtime Environment.
- JDBC specification defined by the Java Community Process

SQL-87

- Published by ISO in 1987 as
  - ISO/IEC 9075-87 Database Language SQL
- Published by ANSI in 1986 as
  - ANSI X3.135 Database Language SQL.
- The technical content was identical
- Editor for both editions was Phil Shaw

SQL-87 Features

- Support for Tabular data – columns and rows
- Select expressions and join syntax
- Arithmetic operators
- Insert, Update, and Delete
- Group by
- Serializable ACID transactions
- Three valued logic – NULL
- Schema creation statements including tables and views

SQL-87 Features

- Access Control
  - Grant, Revoke
- Query language isolated from underlying physical structure
- Exceptions returned using SQLCode values
- “module language” to describe how SQL was effectively executed on a server.
- Specifications for embedding SQL in programming languages were included in an informative annex.

SQL-89

In 1989, SQL-89, support was added for:

- Referential Integrity – constraints
- Program Language Bindings – SQL embedded in COBOL, Fortran, C, etc.
- Editor was Phil Shaw

SQL-92

- SQL-92 was a major expansion.
- Editor: Jim Melton
- ISO and ANSI versions published simultaneously

SQL-92 Features

- Additional transaction modes
  - Read Committed
  - Repeatable Read
  - Serializable remains the default mode
- Multi-national character sets
- Dates, Time, Timestamps, Intervals
  - Date arithmetic operations
- Data Definition enhancements including domains

SQL-92 Features

- Triggers
- Declarative Constraints
- Left, Right, and Full outer joins
- Schema modification statements
- Expanded access control
- Exceptions returned using SQLState

Post SQL-92 Expansions

- SQL/CLI (Call Level Interface) was added in 1995
  - SQL/CLI is roughly equivalent to Microsoft's ODBC.
- SQL/PSM – Persistent Stored Modules published in 1996.
  - Added procedural language constructs

SQL:1999

- SQL:1999 was published in multiple parts
  - ISO/IEC 9075:1999
  - ISO/IEC version adopted as a US ANSI standard
- Editor: Jim Melton

SQL:1999 Features

The significant additions in SQL:1999 were:

- Object/Relational – User Defined Types including methods, subtypes, etc.
- User defined Functions
- Cascaded referential integrity
- Online Analytical Processing (OLAP)
  - Added as an amendment in 2001 (or so)
  - Incorporated into SQL/Foundation in 2003

SQL:2003

Published in nine parts, including:

- SQL/Framework – Common information across all parts
- SQL/Foundation – Capabilities that are not separable
- SQL/CLI – Call-Level Interface
  - Approximately equivalent to Microsoft’s ODBC
- SQL/PSM – Persistent Stored Modules
  - Procedural language extensions

SQL:2003

- SQL/MED – Management of External Data
  - Support for mapping external data to allow SQL table access
- SQL/OLB – Object Language Bindings
  - SQL embedded in the Java programming language
- SQL/Schemata – the Schema Information Tables
  - had been included and expanded since SQL-87.
  - In SQL:2003, extracted into a separate part and expanded.
- SQL/JRT – Java Routines and Types
  - Ability to embed Java objects and libraries in an SQL database
- SQL/XML
  - Store, query, and retrieve XML data in an SQL database
  - Expanded version of SQL/XML published in 2006.

SQL:2008

- Regular Expression Enhancements
- Minor enhancements & Corrections
- Additional expansion of SQL/XML
- Primary Editor Jim Melton
    - Associate Editors
    - SQL/MED – Krishna Kulkarni
    - SQL/OLB and SQL/JRT – Chris Farrar
    - SQL/PSM – Stephen Cannan
    - SQL/Schema – Jörn Bartels

SQL:2008 Parts

- Part 1: Framework (SQL/Framework)
- Part 2: Foundation (SQL/Foundation)
- Part 3: Call-Level Interface (SQL/CLI)
- Part 4: Persistent Stored Modules (SQL/PSM)
- Part 9: Management of External Data (SQL/MED)
- Part 10: Object Language Bindings (SQL/OLB)
- Part 11: Information and Definition Schemas (SQL/Schemata)
- Part 13: SQL Routines and Types Using the Java™ Programming Language (SQL/JRT)
- Part 14: XML-Related Specifications (SQL/XML)

SQL:2011

- Minor Enhancements
- Corrections
- Temporal Data
  - System Versioned Tables
    - Transaction Time
    - What did the data look like yesterday?
  - Application Time Period Tables
    - Valid Time
    - What should the data have looked like yesterday?

SQL:2011 Parts

- Part 1: Framework (SQL/Framework)
- Part 2: Foundation (SQL/Foundation)
- Part 4: Persistent Stored Modules (SQL/PSM)
- Part 11: Information and Definition Schemas (SQL/Schemata)
- Part 14: XML-Related Specifications (SQL/XML)

SQL:201x (Probably early 2017)

- Row Pattern Recognition
  - Regular Expressions across sequences of rows
- Support for Java Script Object Notation objects
  - Store, Query, and Retrieve JSON objects
- Polymorphic Table Functions
  - both parameters and function return value can be tables whose shape is not known until runtime.

SQL:201x Parts

- Part 1: Framework (SQL/Framework)
- Part 2: Foundation (SQL/Foundation)
- Part 3: Call-Level Interface (SQL/CLI)
- Part 4: Persistent Stored Modules (SQL/PSM)
- Part 9: Management of External Data (SQL/MED)
- Part 10: Object Language Bindings (SQL/OLB)
- Part 11: Information and Definition Schemas (SQL/Schemata)
- Part 13: SQL Routines and Types Using the Java™ Programming Language (SQL/JRT)
- Part 14: XML-Related Specifications (SQL/XML)

SQL Technical Reports

- SQL Standards committees have accumulated a great deal of descriptive material
- useful but does not belong in the actual standard.
- have started creating Technical Reports from this material
- First was published in 2011.
- Three more were published in July 2015.
- Additional Technical Reports are in process and will be published in conjunction with the next version of the SQL standards.
- The current list of Technical Reports is:

SQL Technical Reports

Reference	Document title	Publication Date
ISO/IEC TR 19075-1:2011	Information technology -- Database languages -- SQL Technical Reports -- Part 1: XQuery Regular Expression Support in SQL	2011-07-06
ISO/IEC TR 19075-2	Information technology -- Database languages -- SQL Technical Reports -- Part 2: SQL Support for Time-Related Information	2015-07-01
ISO/IEC TR 19075-3	Information technology -- Database languages -- SQL Technical Reports -- Part 3: SQL Embedded in Programs using the Java TM programming language	2015-07-01
ISO/IEC TR 19075-4	Information technology -- Database languages -- SQL Technical Reports -- Part 4: SQL with Routines and types using the JavaTM programming language	2015-07-01
ISO/IEC TR 19075-5	Information technology -- Database languages -- SQL Technical Reports -- Part 5: Row Pattern Recognition in SQL	2016
ISO/IEC TR 19075-6	Information technology -- Database languages -- SQL Technical Reports -- Part 6: SQL support for JSON	2016
19075-7	SQL Technical Reports - Part 7: SQL Support for Polymorphic Table Functions	2017
19075-8	SQL Technical Reports - Part 8: SQL Support for Multi-Dimensional Arrays	2017

SQL Futures

The SQL standards committees are following the database industry and have a lot of overlap with the Big Data standardization efforts.

- Some work is already in progress to support Big Data style applications.
- **SQL/MDA – Multi-Dimensional Arrays**
  - Support for complex operations on multi-dimensional arrays within the context of an SQL database
  - Probably 2017
- **Support for Streaming data**
  - In initial discussion of requirements and specifications
  - General idea is to map an incoming stream of data and process it using all of the existing SQL capabilities, including Row Pattern Recognition
  - Potentially 2018

SQL Futures – Big Data

- 2014 JTC1 Study Group on Big Data
- 2015 Big Data Standards Groups
- Standards Gaps Identified by SGBD
- Standards Gaps in SC32’s Space

JTC1 Study Group on Big Data

- Completed in September 2014
- Identified 16 Standards Gaps
- Recommendations
  - Assign standards gaps to existing JTC1 Sub Committees
  - Creation of a new Working Group
- Report:

  http://www.iso.org/iso/big_data_report-jtc1.pdf

Current Big Data Standards Groups

Efforts to define and standardize aspects of Big Data

1. ISO/IEC JTC1 Working Group on Big Data (WG9)
   - Group recommended by SGBD
   - Convenor – Wo Chang
2. INCITS Big Data Technical Committee
   - USA Mirror to JTC1 WG9
   - Chair – David Boyd
   - Vice Chair – Keith Hare
3. NIST Big Data Public Working Group
   - Industry group in the USA
   - Chair – Wo Chang

Three groups overlap, communicate, and cooperate

Standards Gaps Identified by SGBD

1. *Big Data use cases, definitions, vocabulary and reference architectures (e.g. system, data, platforms, online/offline, etc.)*
2. *Specifications and standardization of metadata including data provenance*
3. *Application models (e.g. batch, streaming, etc.)*
4. *Query languages including non-relational queries to support diverse data types (XML, RDF, JSON, multimedia, etc.) and Big Data operations (e.g. matrix operations)*
5. *Domain-specific languages*
6. *Semantics of eventual consistency*
7. *Advanced network protocols for efficient data transfer*
8. *General and domain specific ontologies and taxonomies for describing data semantics including interoperation between ontologies*

Standards Gaps Identified by SGBD

9. *Big Data security and privacy access controls.*
10. *Remote, distributed, and federated analytics (taking the analytics to the data) including data and processing resource discovery and data mining*
11. *Data sharing and exchange*
12. *Data storage, e.g. memory storage system, distributed file system, data warehouse, etc.*
13. *Human consumption of the results of big data analysis (e.g., visualization)*
14. *Energy measurement for Big Data*
15. *Interface between relational (SQL) and non-relational (NoSQL) datastores*
16. *Big Data Quality and Veracity description and management*

Standards Gaps in SC32's Space

- Definition of standard interfaces (e.g., language, API) to support non-relational datastores (4)
- Definition of SQL extension to support exchange and integration between SQL and non-SQL datastores (11, 15)
- Metadata and provenance standards (2,9)
- SQL and NoSQL standards for data mining (10)
- Support for large complex data structures in SQL and/or SQL/MM (4,11)
- Support for operations on complex data structures and defined operations on such structures (e.g. add, multiply union) (4,5)
- Standards for eventual consistency and acceptable consistency (6)

SC32 Challenges

- How does all of this fit into the SQL Eco-system?
- Lots of existing work to support bits of the big data standards gaps
- Existing standards need review and enhancement
- New efforts already in progress
- Some completely new work needed

Standard interfaces for non-relational datastores (4)

4. Query languages including non-relational queries to support diverse data types (XML, RDF, JSON, multimedia, etc.) and Big Data operations (e.g. matrix operations)

- Diverse data types
  - XML – Supported since 9075:2003
  - JSON – In next edition of 9075 (2016 or 2017)
  - RDF – graph data structures
  - Multimedia – SQL/MM standard
  - Etc.
- Big Data operations
  - Matrix operations – new 9075 part, SQL Multi Dimensional Arrays
  - Support for streaming data – under discussion
  - Etc.

SQL extension for exchange and integration (11, 15)

11. Data sharing and exchange

15. Interface between relational (SQL) and non-relational (NoSQL) datastores

- APIs for data access
  - SQL/CLI
  - JDBC
  - Others?
- Support for registering data sources
- Support for identifying potentially useful data sources
- Identify requirements for data exchange

Metadata and provenance standards (2,9)

2. Specifications and standardization of metadata including data provenance

9. Big Data security and privacy access controls.

- ISO/IEC 11179
- Review provenance requirements
- Data Source registry using 11179
- Security & Privacy
  - Similar topic space with differences
  - Difficult problems

SQL and NoSQL standards for data mining (10)

10. Remote, distributed, and federated analytics (taking the analytics to the data) including data and processing resource discovery and data mining

- SQLMM Data Mining
- Support for registering data sources
- Support for identifying potentially useful data sources

Support for large complex data structures in SQL and/or SQL/MM (4,11)

4. Query languages including non-relational queries to support diverse data types (XML, RDF, JSON, multimedia, etc.) and Big Data operations (e.g. matrix operations)

11. Data sharing and exchange

- Operations on Images, video, sound, etc.

Operations on complex data structures (e.g. add, multiply, union) (4,5)

4. Query languages including non-relational queries to support diverse data types (XML, RDF, JSON, multimedia, etc.) and Big Data operations (e.g. matrix operations)

5. Domain-specific languages

- External technologies
  - R Statistical Computing Platform
  - W3C specifies SPARQL Query Language for RDF
- SC32 WG3 Technologies
  - Support for matrices under development – SQL/Multi-Dimensional Arrays
  - SQL support for graph data structures has been discussed, but no concrete change proposals

Standards for eventual consistency and acceptable consistency (6)

6. Semantics of eventual consistency

- ISO/IEC 9075 currently specifies details for ACID transactions
  - Atomic, Concurrent, Isolated, Durable
  - Defined in terms of phenomena
  - Supports specification of transaction consistency
    - Serializable, Repeatable Read, Read Committed, Read Uncommitted
- Need to investigate
  - BASE Transactions
    - Basically Available, Soft state, Eventual consistency
  - Brewer’s CAP theorem
    - Consistency, Availability and Partition Tolerance – pick two
- ACID transaction consistency not the same as BASE transaction consistency

Eventual Consistency

- Mechanism for finding out what types of transaction a database supports
- Specify syntax for choosing two of Consistency, Availability and Partition Tolerance

Data Source Registry Requirements

- Ability to register a database in a metadata registry
  - Create a new metadata model
  - Tie source data to the registry entries
  - Match/map to an existing metadata model – might be manual
- Ability to create tables in a database using components from a metadata registry
  - Tie the database metadata back to the source registry
    - Registry URL & Unique identifier/certificate
    - Unique identify for each table and column
    - Timestamp
  - Could be syntax on Create Table statement
- “Semantics is not computable – approximate semantics with syntax”

Why Should You Care?

- Benefits of standards are long term
  - Standards development is long term – 3 to 5 years
  - Implementations could be in advance of, in parallel with or trailing the standards publications
- Benefits for Users
  - Easier to integrate tools
  - Easier to find personnel
  - Easier to integrate multiple data stores
  - Focus on the data and analysis, rather than the tools
- Benefits for Implementers
  - Focus on performance rather than interface
  - Expands the market

Summary

- SQL Standards have a long history
  - Practice integrating new technologies
  - Used for large number of existing applications
  - Unlikely to be replaced in the near future
- Big Data paradigm identifying new requirements
  - Many overlap with existing SQL technology
  - New requirements will be integrated into the SQL eco-system

Acknowledgements

My understanding of Big Data has been influenced and informed by discussions with participants in the JTC1 SC32 Working Groups, the NIST Big Data Public Working Group, JTC1 Study Group on Big Data, as well as numerous articles and books.

All errors, misunderstandings, misleading statements, and idiotic comments are mine and mine alone.

Keith W. Hare
Senior Consultant
JCC Consulting, Inc.
600 Newark Granville Road
Granville, OH 43023 USA
keith@jcc.com
+1.740.587.0157
www.jcc.com

References

- Donald R. Deutsch, "The SQL Standard: How it Happened", IEEE Annals of the History of Computing, vol.35, no. 2, pp. 72-75, April-June 2013, doi:10.1109/MAHC.2013.30
- ISO/IEC 9075-1 Information technology -- Database languages -- SQL -- Part 1: Framework (SQL/Framework)
- ISO/IEC 9075-2 Information technology -- Database languages -- SQL -- Part 2: Foundation (SQL/Foundation)
- ISO/IEC 9075-3 Information technology -- Database languages -- SQL -- Part 3: Call-Level Interface (SQL/CLI)
- ISO/IEC 9075-4 Information technology -- Database languages -- SQL -- Part 4: Persistent Stored Modules (SQL/PSM)

References

- ISO/IEC 9075-9 Information technology -- Database languages -- SQL -- Part 9: Management of External Data (SQL/MED)
- ISO/IEC 9075-10 Information technology -- Database languages -- SQL -- Part 10: Object Language Bindings (SQL/OLB)
- ISO/IEC 9075-11 Information technology -- Database languages -- SQL -- Part 11: Information and Definition Schemas (SQL/Schemata)
- ISO/IEC 9075-13 Information technology -- Database languages -- SQL -- Part 13: SQL Routines and Types Using the Java^{TM} Programming Language(SQL/JRT)
- ISO/IEC 9075-14 Information technology -- Database languages -- SQL -- Part 14: XML-Related Specifications (SQL/XML)

Big Data Standards Groups

Efforts to define and standardize aspects of Big Data

- ISO/IEC JTC1 Working Group on Big Data (WG9)
  - Convenor – Wo Chang
- INCITS Big Data Technical Committee (USA)
  - Chair – David Boyd
  - Vice Chair – Keith Hare
- NIST Big Data Public Working Group (USA)
  - Wo Chang

ISO/IEC JTC1 Working Group on Big Data (WG9)

JTC1 2014 Plenary essentially adopted Study Group recommendations – Resolution 28

- Established Working Group 9 on Big Data reporting directly to JTC 1
  - Appointed Mr. Wo Chang to serve as Convenor
- Cross SC coordination tasks assigned to Special Working Group on Management (SWG-M)

JTC1 WG9 – Big Data Projects

- ISO/IEC 20546 Information technology Big Data -- Overview and Vocabulary
- ISO/IEC ISO/IEC 20547 Big Data - Reference architecture
  - ISO/IEC TR 20547-1 Information technology -- Big Data Reference Architecture -- Part 1: Framework and Application Process
  - ISO/IEC TR 20547-2, Information technology -- Big Data Reference Architecture -- Part 2: Use Cases and Derived Requirements
  - ISO/IEC 20547-3, Information technology -- Big Data Reference Architecture -- Part 3: Reference Architecture
  - ISO/IEC 20547-4, Information technology -- Big Data Reference Architecture -- Part 4: Security and Privacy Fabric
  - ISO/IEC TR 20547-5, Information technology -- Big Data Reference Architecture -- Part 5: Standards Roadmap
- Target standards publication in 2018
- Participants identified by national bodies

INCITS Big Data Technical Committee

- USA mirror group to JTC1 WG9 Big Data
  - Chair: David Boyd
  - Vice Chair: Keith Hare
- INCITS participants are registered as participants in JTC1 WG9
- INCITS group provides US input to JTC1 WG9

NIST Big Data Public Working Group

- USA National Institute of Standards and Technology Big Data Public Working group meeting since June, 2013.
- Weekly web conferences
- Web Site: http://bigdatawg.nist.gov/home.php
- NIST BD PWG Output documents were input to the JTC1 Study Group on Big Data
- Future NIST BD PWG documents likely to be input to JTC1 WG9 via the INCITS group

NIST BD PWG Output Documents

NIST Big Data Definitions & Taxonomies Subgroup

- Volume 1: NIST Big Data Definitions
- Volume 2: NIST Big Data Taxonomies

NIST Big Data Use Case & Requirements Subgroup

- Volume 3: NIST Big Data Use Case & Requirements

NIST Big Data Security & Privacy Subgroup

- Volume 4: NIST Big Data Security and Privacy Requirements

NIST Big Data Reference Architecture Subgroup

- Volume 5: NIST Big Data Architectures White Paper Survey
- Volume 6: NIST Big Data Reference Architecture

NIST Big Data Technology Roadmap Subgroup

- Volume 7: NIST Big Data Technology Roadmap