

# SQL Standards Status and Directions

September 2011  
Rdb Technical Forum

**Keith W. Hare**  
JCC Consulting, Inc.  
600 Newark Road, P.O. Box 381  
Granville, OH 43023 USA  
+1.740.587.0157  
[Keith@jcc.com](mailto:Keith@jcc.com)

# Abstract

While the SQL Standard has not been at the forefront of the computing press, it is still progressing and expanding.

- What is SQL?
- What distinguishes SQL?
- ISO and ANSI
- Current Status
- Directions

# Who Am I?

- Muskingum College, 1980, BS in Biology and Computer Science
- Senior Consultant with JCC Consulting, Inc. since 1985 – high performance database systems
- Ohio State – Masters in Computer & Information Science, 1985
- SQL Standards committees since 1988
- Vice Chair, INCITS H2 since 2003
- Convenor, ISO/IEC JTC1 SC32 WG3 since 2005

# 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

# What is a Data Base?

- A data base is a way of storing and retrieving data in a coherent manner
- Hierarchical – early 1960s – IBM's IMS
- Network – late 1960s
- Relational – 1970s – System R, Ingres, Oracle
- SQL – 1980s – DB2, Ingres, Oracle, Sybase, Microsoft, MySQL...

# SQL Data Bases

- Data stored in columns and tables
- Relationships represented by data
- Data Manipulation Language
- Data Definition Language
- Transactions

# 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



# 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

# Other Data Base Features

The SQL Standard specifies the logical data definition and manipulation, not the physical data structures.

- Does not specify how to define:
  - Storage Areas/Files/Table Spaces
  - Indexes
  - Query Optimization strategies
  - Transaction Journals/Logs
  - Backup & Restore
  - Etc.

# What Distinguishes SQL?

Several factors distinguish SQL from other query and data storage technologies:

- Persistent data accumulated over long periods
- Complex update and read-only transactions
- Concurrent update and read-only access
- Null values and three-valued logic
- Integration with other technologies

# ISO and ANSI

In the international arena, the SQL Standard is developed by ISO/IEC JTC1 SC32 WG3.

■ Officers:

- Convenor – Keith W. Hare – USA
- Editor – Jim Melton – USA

■ Active participants are:

- Canada – Standards Council of Canada
- China – Chinese Electronics Standardization Institute
- Germany – DIN Deutsches Institut für Normung e. V.
- Great Britain – British Standards Institution
- Japan – SQL working group of JIS (Japan Industrial Standards)
- Netherlands
- USA – ANSI INCITS

# Translation

- ISO/IEC
  - ISO – International Organization for Standardization
  - IEC – International Electrotechnical Commission
- JTC1 – Joint Technical Committee 1
  - Information Technology
- SC32 – SubCommittee 32
- WG3 – Working Group 3 – the committee responsible for data base language standards

# USA – ANSI INCITS DM32.2

## ■ Officers

- Chair: Don Deutsch – Oracle
- Vice Chair: Keith Hare – JCC Consulting, Inc.
- Secretary – Mike Gorman – Whitemarsh
- International Representative – Krishna Kulkarni – IBM
- Editor – Jim Melton – Oracle

- DM32.2 (originally H2) organized in 1978 to standardize database languages.

## ■ Current Participants

- Bentley Systems
- Computer Associates
- HP
- IBM
- Ingres
- Intersystems
- JCC Consulting, Inc.
- Microsoft
- Oracle Corporation
- Software Workshop, Inc
- Sybase
- Teradata
- Whitemarsh Information Systems

# Translation

- ANSI – American National Standards Institute
- INCITS – InterNational Committee for Information Technology Standards
- DM32.2 – the committee responsible for database standards

# Current Status

- Brief History
- SQL 2003
- SQL 2008
- SQL 2011
- Validation Test
- SQL/MM



# Brief History

The following is a brief history of the SQL Standard's major revisions:

- SQL-86 (ANSI) and SQL-87 (ISO)
- SQL-89
- SQL-92
- SQL:1999
- SQL:2003
- SQL:2008
- SQL:2011

# SQL:2003 & SQL:2008

With these changes, the complete list for SQL 2003 & SQL 2008 are:

- 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

SQL:2011 is in process for a final ballot (FDIS)

- If ballot is started by October 1, will be SQL:2011
- If ballot is later, will be SQL:2012

Consists of five 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)

# Page Count Comparison

<b>Part</b>	<b>SQL 1992</b>	<b>SQL 1999</b>	<b>SQL 2008</b>	<b>SQL 2011</b>
Part 1 – SQL/Framework		85	93	88
Part 2 – SQL/Foundation	628	1,147	1,361	1,470
Part 3 – SQL/CLI (1995)	236	421	403	
Part 4 – SQL/PSM (1996)	256	170	190	198
Part 5 – SQL/Bindings		261		
Part 9 – SQL/MED			485	
Part 10 – SQL/OLB			414	
Part 11 – SQL/Schemata			298	314
Part 13 – SQL/JRT			208	
Part 14 – SQL/XML			444	457
<b>Total</b>	<b>1,120</b>	<b>2,084</b>	<b>3,896</b>	<b>2,527</b>

# SQL/Foundation:2011 New Features

- The following slides briefly discuss some of the new features in SQL/Foundation:2011
  - System-Versioned Tables – Transaction Time
  - Application-Time Period Tables – Valid Time
  - Other small features this talk will ignore

# System Versioned Tables

```
CREATE TABLE employees
(emp_name VARCHAR(50) NOT NULL
,dept_id VARCHAR(10)
,system_start TIMESTAMP(6) GENERATED ALWAYS AS ROW START
,system_end TIMESTAMP(6) GENERATED ALWAYS AS ROW END
,PERIOD FOR SYSTEM_TIME (system_start, system_end)
,PRIMARY KEY (emp_name)
,FOREIGN KEY (dept_id)
    REFERENCES departments (dept_id)
) WITH SYSTEM VERSIONING;
```

(Example thanks to Krishna Kulkarni)

# System Versioned Tables

- Essentially an audit trail of changes
- System\_Time values set by implementation
- Simple Select returns the current version:
  - Select Emp\_Name, Dept\_id  
from Employees
- Syntax for executing a query as of a point in time
  - Select Emp\_Name, Dept\_id  
from Employees  
**for system\_time as of** '2011-08-01 08:00'

# Application Time Period Tables

```
CREATE TABLE employees
(
  emp_name VARCHAR(50) NOT NULL PRIMARY KEY
  ,dept_id VARCHAR(10)
  ,start_timestamp timestamp NOT NULL
  ,end_timestamp timestamp NOT NULL
  ,PERIOD FOR emp_period (start_timestamp, end_timestamp)
  ,PRIMARY KEY (emp_name, emp_period WITHOUT OVERLAPS)
  ,FOREIGN KEY (dept_id, PERIOD emp_period)
    REFERENCES
      departments (dept_id, PERIOD dept_period)
);
```

(Example thanks to Krishna Kulkarni)



# Application Time Period Tables

- Start and End Timestamps provided by application
- Start Timestamp < End Timestamp
- For current row, End Timestamp is implementation-defined maximum value for the datatype
- **Without Overlaps** enforced by SQL implementation
- Upward compatible with existing data

# System-Versioned & Application-Time Period Tables

Expansion of SQL Standard to deal with issues of time

- Transaction time – What was the data when it was recorded
- Valid Time – What should the data have been at a particular point in time
- Roots in SQL/Temporal (Part 7) work in late 1990's
  - Two competing academic viewpoints
    - Richard T. Snodgrass, Christian S. Jensen, “Developing Time-Oriented Database Applications in SQL”, Morgan Kaufmann, 1999. This book is out of print but can be downloaded from Rick’s web site at <http://www.cs.arizona.edu/people/rts/>
    - C. J. Date, Hugh Darwen, & Nikos Lorentzos, “Temporal Data & the Relational Model”, Morgan Kaufmann, 2002.
- Current approach is evolutionary – supports adding temporal information to existing tables

# Directions

Over the decades, the SQL Standards process has incorporated new technology directions:

- Object oriented capabilities.
- Using Java with SQL.
- Integrating SQL and XML.
- System-Versioned & Application-Time Period Tables
- Near-term Work
  - Row Pattern Recognition
- Possible future work
  - NoSQL Database interface

# Row Pattern Recognition

New proposal for additional capabilities to support streaming data queries

- Patterns defined using Regular Expressions
- Regular Expression variables span sub-sequences of rows
- Defined using conditions on individual rows and their aggregates

# NoSQL Distinguishing Characteristics

- Large data volumes
  - Google's "big data"
- Scalable replication and distribution
  - Potentially thousands of machines
  - Potentially distributed around the world
- Queries need to return answers quickly
- Mostly query, few updates
- Asynchronous Inserts & Updates
- Schema-less
- ACID transaction properties are not needed – BASE
- CAP Theorem
- Open source development

# NoSQL: Storing and Modifying Data

- Syntax varies
  - HTML
  - Java Script
  - Etc.
- Asynchronous – Inserts and updates do not wait for confirmation
- Versioned
- Optimistic Concurrency

# NoSQL: Retrieving Data

- Syntax Varies
  - No set-based query language
  - Procedural program languages such as Java, C, etc.
- Application specifies retrieval path
- No query optimizer
- Quick answer is important
- May not be a single “right” answer

# NoSQL: Open Source

- Small upfront software costs
- Suitable for large scale distribution on commodity hardware



# NoSQL Summary

- NoSQL databases reject:
  - Overhead of ACID transactions
  - “Complexity” of SQL
  - Burden of up-front schema design
  - Declarative query expression
  - Yesterday’s technology
- Programmer responsible for
  - Step-by-step procedural language
  - Navigating access path

# Getting Copies of the Standards

Because of ANSI and ISO copyright restrictions, copies of the SQL standards specifications are available only for purchase.

- Electronic (PDF) of printed copies of the SQL standards are available from several sources.
- ISO – <http://www.iso.ch>, click on "ISO STORE", and search for 9075. Prices are in Swiss Francs. A CD of all 9 parts is CHF 356.00
- ANSI – <http://www.ansi.org/>, click on "eStandards Store" and search for SQL.
  - Make sure you choose the 2008 (soon 2011) versions
  - ISO/IEC 9075-\*:2008

# Summary

- SQL Standards development is ongoing
- Visibility of SQL Standards development is not high
- New standards development is following new technologies
  - Temporal data this year
  - Row Pattern Recognition soon
  - Standard NoSQL language?

# Questions?

