Standardizing Graph Database Functionality

An Invitation to Collaborate

Jan Michels, Keith Hare, Jim Melton
ISO/IEC JTC 1/SC 32/WG 3 Members

February 9, 2017

Safe Harbor Statement

The following is intended to outline the general direction. It is intended for information purposes only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making decisions. The development, release, and timing of any features, functionality, or standards described remains at the sole discretion of whoever develops or releases such features, functionality, or standards.

Agenda

1. Intro
2. Standards 101 – Why, Where, When, Who
3. WG 3 – SQL and other database query languages
4. Questions/discussion

Who we are

- Jan Michels
  - Expert member of ISO/IEC JTC 1/SC 32/WG 3
  - Oracle representative in ANSI INCITS DM32.2
  - jan.michels@oracle.com
- Keith Hare
  - Convenor of ISO/IEC JTC 1/SC 32/WG 3
  - Vice Chair of ANSI INCITS DM32.2
  - keith@jcc.com
- Jim Melton
  - Chair of ISO/IEC JTC 1/SC 32
  - Editor of ISO/IEC 9075 (SQL standard)
  - jim.melton@oracle.com

Agenda

1. Intro
2. Standards 101 – Why, Where, When, Who
3. WG 3 – SQL and other database query languages
4. Questions/discussion

What Is Standardization?

- Formulation, publication, and implementation
- of guidelines, rules, and specifications
- for common and repeated use,
- aimed at achieving optimum degree of order or uniformity
- in a given context, discipline, or field.*

* http://www.businessdictionary.com/definition/standardization.html

Why Standardize?

- Standardization can help
  - Maximize compatibility, interoperability, safety, repeatability, or quality.*
  - Facilitate commoditization of formerly custom processes.*
- Effects:*
  - Shift competition from integrated systems to individual components
  - Increase compatibility and interoperability between products
  - Reduced uncertainty
  - Increased adoption of a new technology
  - Increased flexibility, rapid introduction of new products
  - Shift competition from features to price

* https://en.wikipedia.org/wiki/Standardization

Where Does Standardization Happen?

- Locally — Corporate standards, popular specs with community of users
- *De facto* — Formal or informal organizations develop specs that become popular because of excellence, utility, or lack of alternatives; consortia are often sources of *de facto* standards
- *De jure* — Formal organizations, with national or international recognition, following processes designed to be open, develop specs for broad use
- Treaty — Formal organizations develop specs that become part of treaties, required for international interoperability

When Does Standardization Happen?

- Sufficient perceived need for benefits
- Sufficient community of implementers and users
- Government requirements

- In advance of implementation (“inventiveness”)
- Following wide implementation (“document”)
- Just-in-time (“collaboration”)

Who Does Standardization?

- Implementers — promote their ideas, products
- Users — encourage required capabilities
- Academia — apply theories, encourage research
- Government — maximize choice, minimize cost

- Engineers — Do the heavy lifting, provide expertise
- Marketers — Help guide direction of standardization
- Businesspeople — Fund, provide expectations
- Academics — Get PhD degrees ☺

De Jure Standards Organizations

- International:
  - ISO — International Organization for Standardization
  - IEC — International Electrotechnical Commission
  - ITU — International Telecommunications Union
  - …

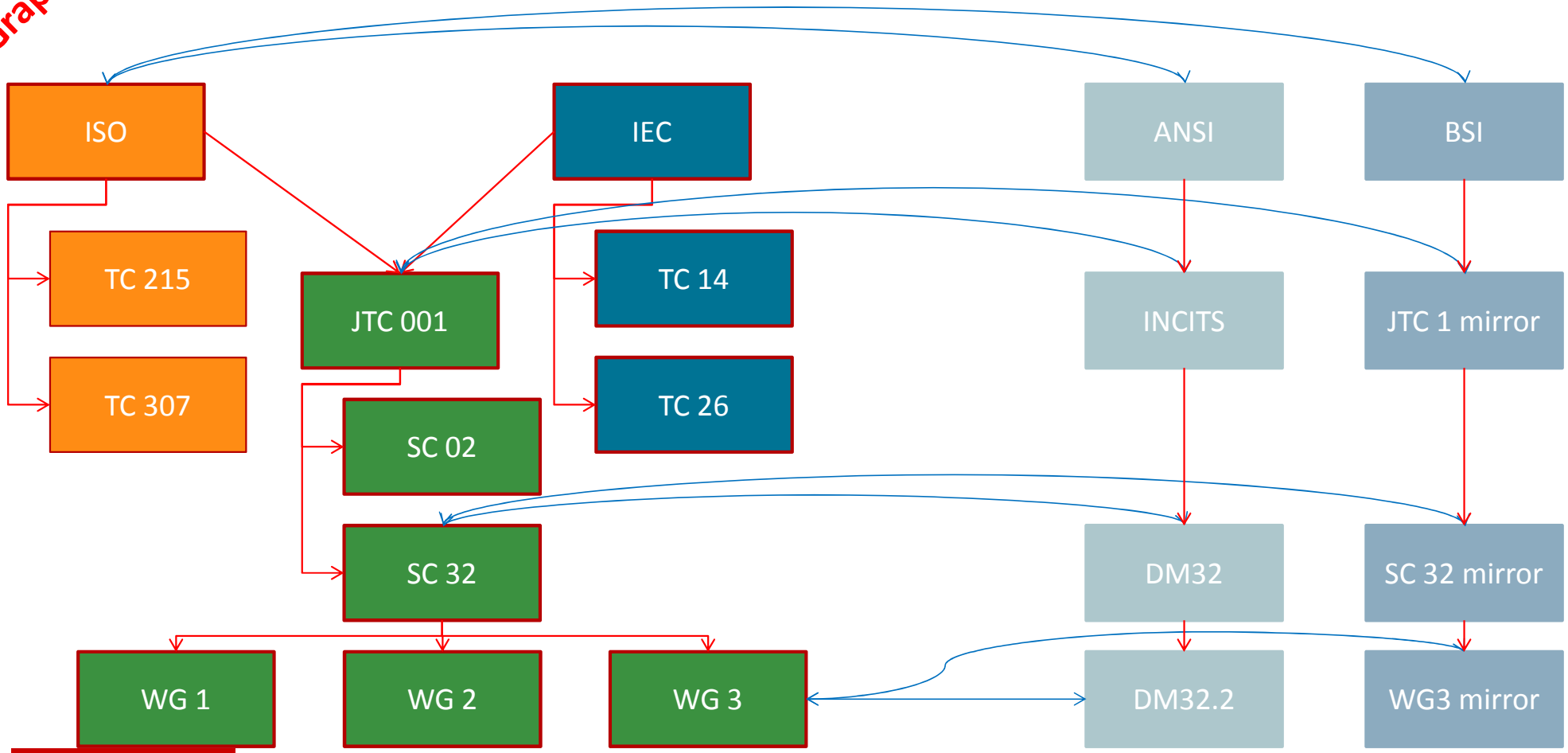
- National:
  - ANSI — American National Standards Institute
    - INCITS — International Committee for Information Technology Standardization
    - IEEE — Institute of Electrical and Electronics Engineers
    - …
  - BSI — British Standards Institute
  - DIN — Deutsches Institut für Normung
  - AFNOR — Association Française de Normalisation
  - …

Agenda

1. Intro
2. Standards 101 – Why, Where, When, Who
3. WG 3 – SQL and other database query languages
4. Questions/discussion

Graph Query

SQL Standardization — Organizations



ORACLE

Standardization – Organizations (cheat sheet)

- ISO: International Organization for Standardization
  - www.iso.org
- IEC : International Electrotechnical Commission
  - www.iec.ch
- JTC 1: Joint Technical Committee 1 – Information Technology
  - www.iso.org/iso/jtc1_home.html
- SC 32: Subcommittee 32 – Data management and interchange
  - http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_committee.htm?commid=45342
- WG 3: Working Group 3 – Database languages

What has been happening in WG 3?

- SQL:2016 was just published in December of 2016
  - Incorporates all of SQL-87, SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2008, SQL:2011, and newly added support for:
  - Java Script Object Notation (JSON)
    - Query, store, retrieve, construct, and convert JSON values
  - Row Pattern Recognition (RPR)
    - Use regular expressions to find patterns across sequences of rows
  - Polymorphic Table Functions (PTFs)
    - Parameters and function return value can be tables whose shape is not known until run-time
  - Work is currently still on-going on a new part to support multi-dimensional arrays

What is going to happen in WG 3 for SQL:2020?

- WG 3 identified these areas of interest for the next version of the SQL standard:
  - Better support for Big Data applications.
  - **Graph queries.**
  - Approximate queries/aggregates and uncertain data.
  - Integration of statistical packages (i.e./e.g., “R”).
  - MapReduce support.
  - Streaming/continuous queries.
  - Support for blockchains.
  - BASE transactions.
- Any participant can bring forward a proposal for any new functionality

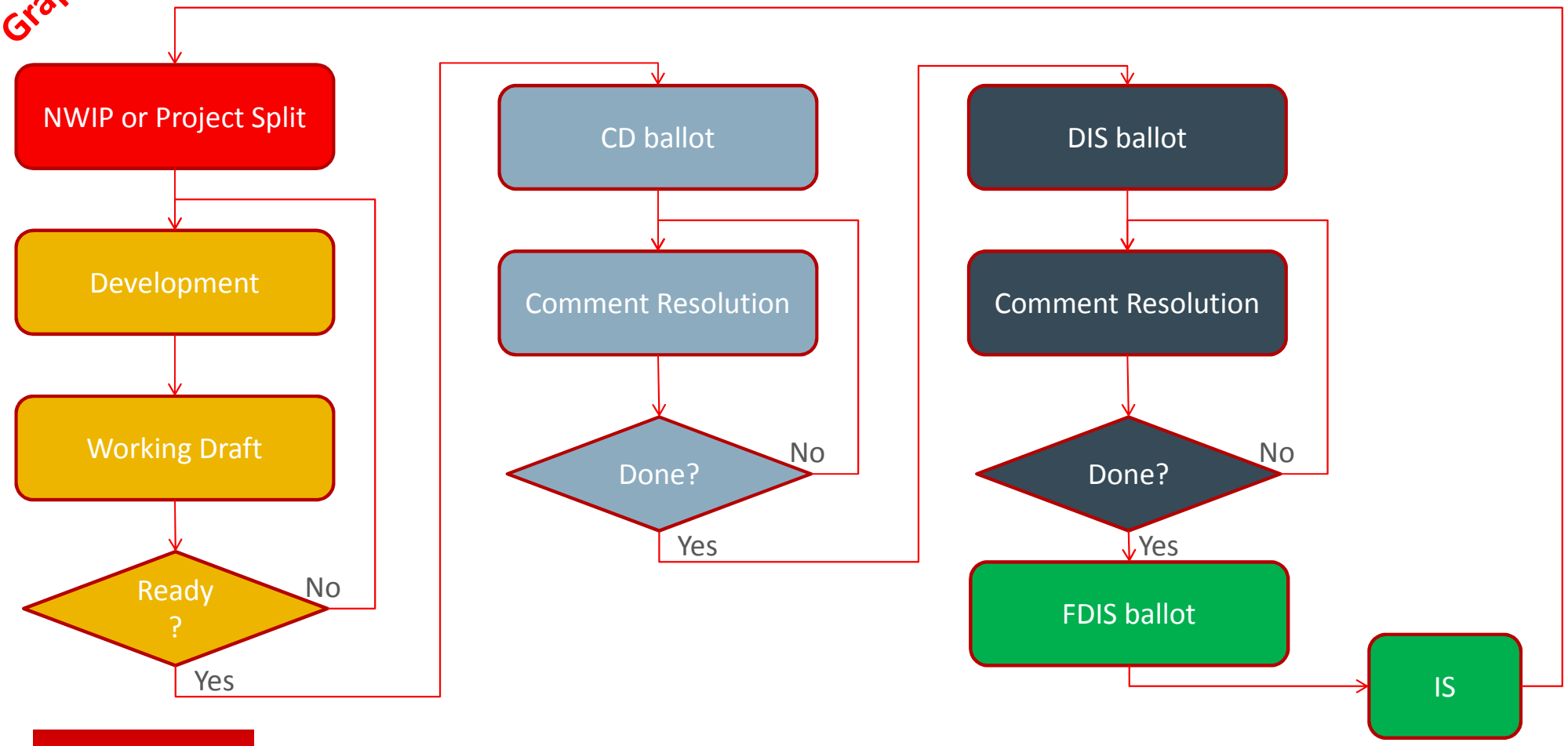
What is going to happen in WG 3 for SQL:2020? *(cont.)*

- At its most recent meeting in January 2017, WG 3 discussed graph databases and query languages:
  - Time is right for formal standards in this area
  - Existing work
  - Technology begins to gel
  - Market demand
- WG 3 is interested in property graph technology
  - **Would like to work with LDBC (Graph QL Task Force) to develop formal standards**
- At the June 2017 SC 32 Plenary, WG 3 will consider:
  - Applying for a project split for SQL/Graph
  - Applying for a new work item/give notice of a preliminary new work item for a "Property Graph Query* Language"

* "Query" does not necessarily mean read-only retrieval operations only, but can include general DML/DDL operations.

Graph Query

SQL Standardization — Process



Standardization – Process (cheat sheet)

- NWIP: New Work Item Proposal
  - *E.g.*, proposal for a (stand-alone) graph query language.
- Project Split
  - New part of an existing standard. *E.g.*, SQL/Graph.
- CD: Committee Draft
- DIS: Draft International Standard
- FDIS: Final Draft International Standard
- IS: International Standard

- Timeline:
  - ISO/IEC JTC1 projects are 24 to 48 months
  - 24 months is very aggressive
  - Ask for 48 months, shoot for 36 months
    - Need a good base of work for a start

How can you get involved/contribute?

- Join your national standards organization/WG3 mirror committee to become a credentialed expert for WG 3
  - E.g., ANSI/INCITS/DM32.2 in the US, DIN in Germany, *etc.*
- Establish a Liaison between LDBC (Graph QL Task Force) and SC 32 (/WG 3)
  - Jim and Keith are happy to work with you on this.
- Informally work with any WG 3 expert
- **Any and all contributions in whichever form are welcome**

Agenda

1. Intro
2. Standards 101 – Why, Where, When, Who
3. WG 3 – SQL and other database query languages
4. Questions/discussion

Backup slides

What is SQL?

- SQL is a language for defining databases and manipulating the data in those databases
- SQL Standard uses SQL as a name, not an acronym
  - 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

Who Develops the SQL Standards?

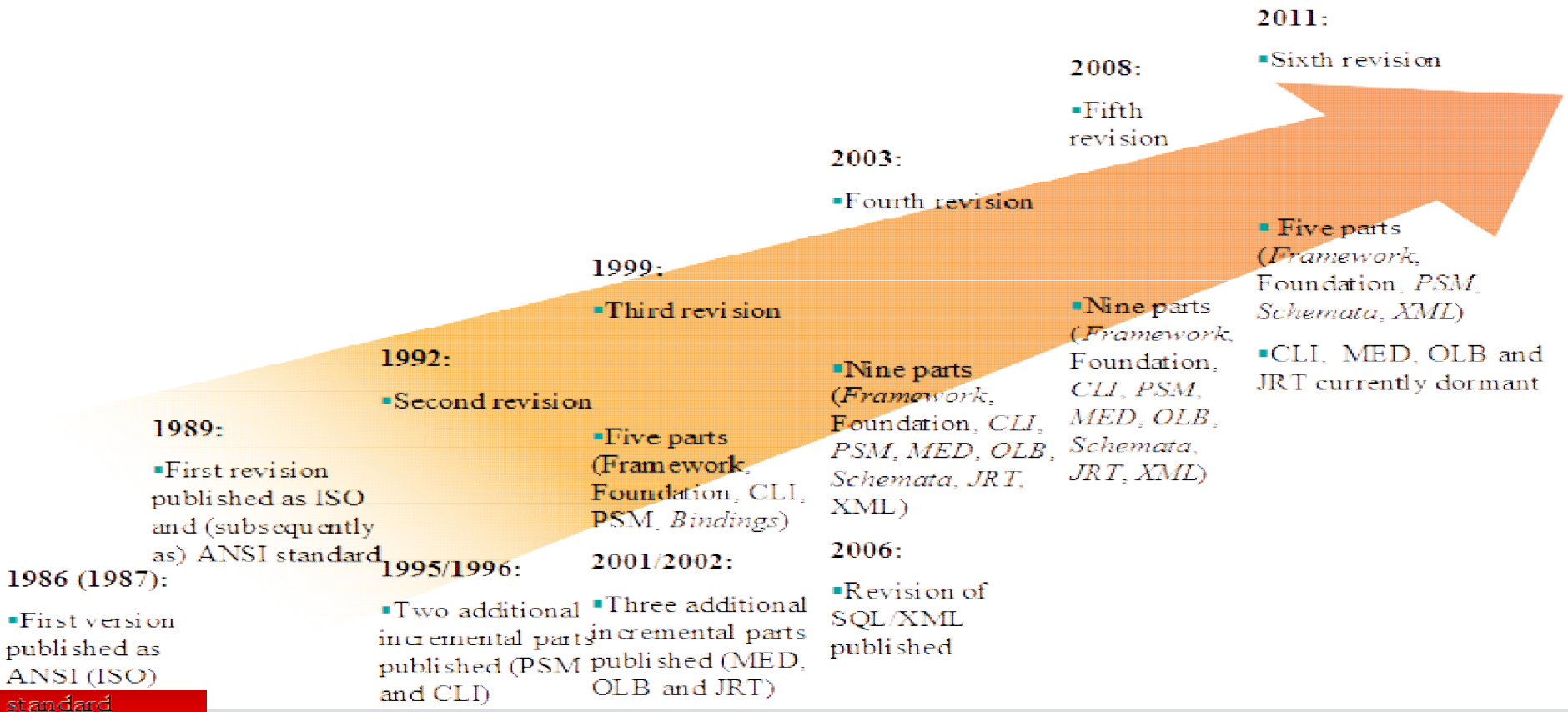
In the international arena, the SQL Standard is developed by ISO/IEC JTC1/SC32/WG3.

- Officers:
  - Convenor – Keith W. Hare – USA
  - Editor – Jim Melton – USA
- Active participants are:
  - Canada: SCC – Standards Council of Canada
  - China: CESI – Chinese Electronics Standardization Institute
  - Germany: DIN – Deutsches Institut für Normung e. V.
  - Great Britain: BSI – British Standards Institution
  - Japan: SQL working group of JIS – Japan Industrial Standards
  - Netherlands: NEN – Netherlands Standardization Institute
  - USA: INCITS – InterNational Committee for Information Technology Standards

SQL:2016 Parts

Document number	Document title
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)
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 programming language (SQL/JRT)
ISO/IEC 9075-14	Information technology -- Database languages -- SQL -- Part 14: XML-Related Specifications (SQL/XML)

SQL Standard (ANSI/ISO/IEC 9075) – A Brief History



SQL Standards – a brief history

- ISO/IEC 9075 Database Language SQL
  - SQL-87
  - SQL-89
  - SQL-92
  - SQL:1999
  - SQL:2003
  - SQL:2008
  - SQL:2011
  - SQL:2016
- Long history of support and expansion of the standard

SQL Standard – Parts Overview

- Part 1: SQL/Framework
  - Structure of the standard and relationship between various parts
  - Common definitions and concepts
  - Conformance requirements statement
- Part 2: SQL/Foundation
  - Specifies the "core" language - all of SQL:1999/Foundation plus several extensions
    - DDL for creating, altering, and dropping various persistent objects including tables, views, user-defined types, and SQL-invoked routines.
    - Predefined data types + type constructors
    - DML for retrieving and updating persistent data
      - Including temporal support
    - Scalar and table expressions
    - Predicates
    - Host language bindings, dynamic SQL, and direct SQL

SQL Standard – Parts Overview *(cont.)*

- Part 3: SQL/CLI
  - A Call-Level Interface for invoking SQL from applications
  - Consists of over 60 routine specifications
    - Control connections to SQL-servers
    - Allocate and deallocate resources
    - Execute SQL statements
    - Control transaction termination
    - Obtain information about the implementation
  - Provided for vendors of truly portable "shrink wrapped" software
    - CLI does not require pre-compilation of the application program
    - Application program can be delivered in "shrink wrapped", object-code form

SQL Standard – Parts Overview *(cont.)*

- Part 4: SQL/PSM
  - Procedural language constructs (similar to those found in block-structured languages)
    - Improve performance in centralized and client/server environments
      - Multiple SQL statements in a single EXEC SQL
      - Multi-statement procedures, functions, and methods
  - Gives great power to DBMS
    - Several control statements (procedural language extension)
      - begin/end block, assignment, call, case, if, loop, for, signal/resignal, variables, exception handling
  - SQL-only implementation of complex functions
    - Without worrying about security ("firewall")
    - Without worrying about performance ("local call")

SQL Standard – Parts Overview *(cont.)*

- Part 9: SQL/MED
  - Datalinks
  - Foreign Data/Tables
- Part 10: SQL/OLB
  - Embedding of SQL statements in Java programs
  - Many differences from the traditional host language bindings:
    - specification in terms of JDBC, but static compilation
    - provides typed cursors and better exception handling
    - platform independence (binary portability)
  - precursor: SQLJ Part 0

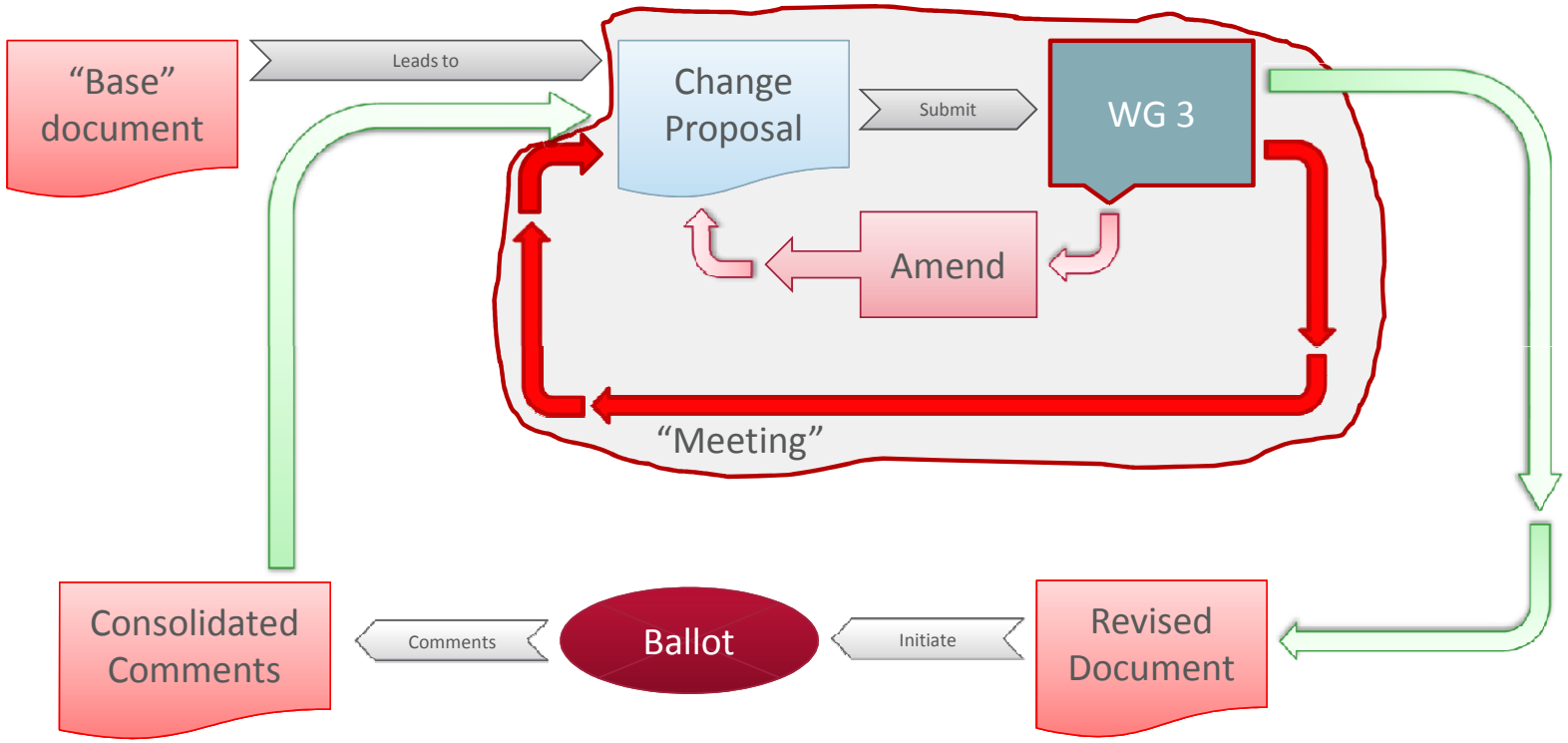
SQL Standard – Parts Overview *(cont.)*

- Part 11: SQL/Schemata
  - Specification of over 65 views that describe the metadata in the SQL-environment:
    - TABLES
    - COLUMNS
    - USER_DEFINED_TYPES, etc.
- Part 13: SQL/JRT
  - "SQL Routines and Types using the Java™ Programming Language"
  - SQL extensions that allow creation of:
    - SQL-invoked routines corresponding to Java static methods
    - SQL user-defined structured types corresponding to Java classes
  - precursor: SQLJ Parts 1 & 2

SQL Standard – Parts Overview *(cont.)*

- Part 14: SQL/XML
  - Major goals:
    - "Publish" SQL query results as XML documents
    - Ability to store and retrieve XML documents
    - Query XML data
  - Rules for mapping SQL types, SQL identifiers and SQL data values to and from corresponding XML concepts
  - A new built-in type XML
    - Based on the XQuery data model
    - Can be an XML document or more complex (i.e., a sequence in the XQuery data model)
  - A number of built-in operators that produce values of type XML
  - A number of built-in expressions to query/manipulate XML values:
    - XMLTable/XMLQuery/XMLExists/XMLCast

SQL Standardization — Step by Step



SQL Standardization Process

- A new project starts with an initial Working Draft (may be blank) – content is built-up by a series of change proposals.
- Project proposal lists a broad list of features that might become part of Working Draft, but the actual content is determined by the proposals submitted by members.
- Proposals are discussed and approved at meetings
  - Majority of work is done prior to the meetings as part of writing proposals
- Editor merges accepted proposals into Working Draft
- Public reviews and ballots at predefined stages

SQL Standardization — Change Proposals

- Structure:
  - Title page — Title, author(s), paper#(s), reference(s), *etc.*
  - Discussion — Motivation, rationale, comments addressed, feature description, *etc.*
  - Detailed proposal — In «document», make the following changes to «section»:
    - Insert this «kind of» rule in «this location»: «rule text»
    - Replace «kind of» rule «number»: «rule text»
    - Modify «kind of» rule «number»: «**modified** ~~rule~~ text»
    - Delete «kind of» rule «number»: «partial text»
  - Checklist
- Very formal change proposal:
  - Allows to review and discuss exactly what is proposed
  - Provides an extensive history of how we arrived at the current state

Pop Quiz

- True or False?
  - There is no CREATE DATABASE statement in the SQL standard. T/F?
  - Other than in its official title, the SQL standard does not define/use the word “database.” T/F?
  - There is a CREATE INDEX statement in the SQL standard. T/F?
  - There is a TRUNCATE TABLE statement in the SQL standard. T/F?
  - SQL:1992 had 628 pages / SQL:2011 has 2011 pages. T/F?
  - Every conforming implementation has to implement each and every feature of SQL:2011. T/F?
  - There exists (at least) one relational DBMS that implements every feature of SQL:2011 . T/F?

Pop Quiz (w/answers)

- True or False?
  - There is no CREATE DATABASE statement in the SQL standard. -> **TRUE**
  - Other than in its official title, the SQL standard does not define/use the word “database.” -> **TRUE**
  - There is a CREATE INDEX statement in the SQL standard. -> **FALSE**
  - There is a TRUNCATE TABLE statement in the SQL standard. -> **TRUE**
  - SQL:1992 had 628 pages / SQL:2011 has 2011 pages. -> **FALSE**, SQL:2011 has 4063 pages
  - Every conforming implementation has to implement each and every feature of SQL:2011. **FALSE**, there is a “Core SQL” conformance level and additional feature IDs for advanced features.
  - There exists (at least) one relational DBMS that implements every feature of SQL:2011. -> **FALSE**