ISO/IEC JTC 1/SC 32 N 0435

Date: 2000-01-27
REPLACES: --

ISO/IEC JTC 1/SC 32

Data Management and Interchange

Secretariat: United States of America (ANSI)
Administered by Pacific Northwest National Laboratory on behalf of ANSI

DOCUMENT TYPE	National Body Contribution
TITLE	ISO Project Proposal for Replication
SOURCE	Herb Sutter (USA)
PROJECT NUMBER	
STATUS	Proposal for USA recommendation to ISO/IEC JTC1/SC32 WG3
REFERENCES	
ACTION ID.	FYI
REQUESTED ACTION	
DUE DATE	
Number of Pages	8
LANGUAGE USED	English
DISTRIBUTION	P & L Members SC Chair WG Conveners and Secretaries

Douglas Mann, Secretariat, ISO/IEC JTC 1/SC 32

Pacific Northwest National Laboratory *, 901 D Street, SW., Suite 900, Washington, DC, 20024-2115, United States of America

Telephone: +1 703 575 2114; Facsimile; +1 703 681 9180; E-mail: MannD@battelle.org
available from the JTC 1/SC 32 WebSite http://bwonotes5.wdc.pnl.gov/SC32/JTC1SC32.nsf

*Pacific Northwest National Laboratory (PNL) administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI

ISO/IEC JTC 1/SC 32 N00435

X3H2-99-475

WG3:SAF-045

File: 99-475.doc

1999-12-23 — 23 December, 1999

Printed: Thursday, 23 December, 1999 at 18:44

Authoritative Version: Adobe Acrobat Portable Document Format (PDF)

ISO

International Organization for Standardization

ANSI

American National Standards Institute

ANSI TC X3H2

Database

ISO/IEC JTC 1/SC 32

Data Management and Interchange

WG 3

Database Languages

Project: ANSI: 1234D — ISO: 1.32.3.4

Title: ISO Project Proposal for Replication

Status: Proposal for USA recommendation to ISO/IEC JTC1/SC32 WG3

Author: Herb Sutter (USA)

Abstract: Urgent market needs for portable database replication facilities may best be met by progression of a standard for such facilities as a new part to SQL:1999.

References:

1) [FrameFDIS] ANSI X3H2-99-074 = WG3:YGJ-006, *(Final Committee Draft) Framework (SQL/Framework)*, March, 1999

2) [FoundFDIS] ANSI X3H2-99-075 = WG3:YGJ-007, *(Final Committee Draft) Foundation (SQL/Foundation)*, March, 1999

3) [CLI-FDIS] ANSI X3H2-99-___ = WG3:YGJ-022, *(Final Committee Draft) Call-Level Interface (SQL/CLI)*, March, 1999

4) [REPL] ANSI X3H2-99-354, *Distributed Database Standardization*, October 1999

1. Discussion

1.1. Overview

The marketplace has been moving to database replication (a.k.a., propagation, synchronization) as a means to deploy multiple local databases where changes made to one local database are asynchronously propagated to the other affected local databases. This paper will use the term “replication” as just described in order to conform to common usage, although this was not the original meaning of the term “replication.”

Database replication capability is a growing requirement in the marketplace and will remain so for the foreseeable future, has become well-understood with mature implementations available, but lacks a useful standard that would help the user community and promote adoption for all vendors. This paper proposes that H2 and/or WG3 initiate work on a standard for database replication, which would in entirely or in part take the form of an incremental part of SQL:1999.

Both database replication (asynchronous) and N-phase commit (synchronous) approaches concern themselves with the ability to deploy multiple local database instances (“local databases”) such that changes performed at one local database are reflected in the other local databases. Different local databases may use the same or different schemas, and may store the same information or subsets of information.

1.2. Historical Context: N-Phase Commit

The salient difference between database replication and N-phase commit is that the former propagates changes asynchronously to other local databases and allows greater autonomy and concurrency, whereas the latter performs changes synchronously at all affected local databases and requires complex elements like distributed lock management.

N-phase commit (including two-phase commit, or 2PC) technology keeps the local databases in ‘lock step,’ whereby a request to commit a change at one local database initiates a distributed transaction such that the local commit does not complete until it is known that the change will successfully commit in all affected local databases. This approach has been a well-understood technology with mature implementations for about 10 years. Despite this, it has not been widely adopted in the marketplace. This appears to be because this approach: reduces autonomy instead of promoting it (the multiple databases are tightly interdependent); often reduces application performance instead of enhancing it, particularly for updates (updates always require distributed transactions); does not scale easily to beyond a few dozen database instances; and can be complex to administer and maintain.

This paper does not propose any work in N-phase commit technologies, which are already mature and well-understood.

1.3. Database Replication

The marketplace has been moving to database replication (a.k.a., propagation, synchronization) as a means to deploy multiple local databases where changes made to one local database are asynchronously propagated to the other affected local databases. For many years, the user community has been producing “homegrown” software because commercial products have only recently become mature. Our committees should consider the following aspects of this part of the market:

2. Needs

2.1. Marketplace Demand for a Replication Standard

To date, there are more commercial database applications developers (independent software vendors, or ISVs) who have created their own in-house replication software than all ISVs using all commercial replication products combined. All of the homegrown and many of the commercial products are incompatible, hindering total market growth for all vendors.

The major reason why ISVs continue to create their own custom replication software is that most ISVs support three to five different database vendor platforms, each of which provides different replication functionality or none at all, and are not able to bear the training and administrative expense of learning and supporting three to five different replication systems. ISVs continue to do this custom software development in spite of research that shows each such replication project effort costs well in excess of US$1 million and that most of such efforts fail.

2.2. Demand Tied To Long-term Growth Trends

The marketplace demand will continue to grow for the foreseeable future because it is being driven by long-term trends. In particular, the following trends each require multiple database instances and means to effectively synchronize changes made in those databases:

2.2.1. PDAs

The rapid adoption of personal digital assistants (PDAs) is being accompanied by the rising demand to use them to store and manipulate enterprise business information locally.

2.2.2. Embedded Systems

The emergence and coming proliferation of embedded systems (for example, parking meters, soda machines, network routers) including local relational storage.

2.2.3. Warehousing, Marts, and Application Integration

There is a strong existing need to maintain multiple databases with different schemas, meaning not just for data warehousing and data marts but also for enterprise application integration and integration of different companies’ or departments’ system, such as after mergers and acquisitions.

2.2.4. Decentralization of Offices

The increase of decentralized offices is being accompanied by a rise in multiple local regional or office servers storing overlapping relational data.

2.2.5. Mobile and Remote Work Forces

The continued proliferation of mobile and remote not-well-connected work forces requires the ability to work with local instances of business applications, including but not limited to the classic case of sales-force automation systems.

2.2.6. E-commerce and Other Web-based Systems

The industry is experiencing a rapid deployment of e-commerce systems with hundreds of thousands of users per day, or more, for which a single central database is often not tolerable as either a scalability bottleneck or as a point of failure. Traditional centralized clustering does not resolve the problem because another common requirement is to maintain web back-ends in multiple geographical locations (specifically internationally) both for load distribution and to provide faster access to users in particular geographies.

2.3. Absence of a Suitable Standard

No existing or proposed standard addresses these requirements completely. For example, SQL/MED includes some features addressing multiple local databases, but is not a complete solution that meets all of the requirements listed in §2.2.

3. Existing Practice

In early 1999, replication capabilities are available in several forms. Various SQL database vendors have implemented a variety of replication-oriented features into their products to provide support for specific parts of the overall problem; without a standard to guide them, the vendors' choices of features are incompatibly different between products with the predictable result of confused and frustrated customers who are increasingly "locked into" specific products. A number of after-market tools and layered products have been marketed in an attempt to provide more common and heterogeneous replication capabilities; however, these tools and products have also suffered from the lack of a guiding standard and are at least as varied as the SQL database products' efforts.

In the vacuum left by lack of a standard response to the replication market needs, no vendor's solutions have emerged as a *de facto* standard, although there remains the possibility that some dominant database vendor will stumble into an opportunity to promulgate such a standard without the participation of other vendors or the customer base.

4. Expected Stability

In the last couple of years, the market requirements for replication capabilities and the technical solutions responding to those requirements have finally gelled. The requirements are much better understood today than they were only five years ago, and the database implementation facilities have improved enough to make responses to the requirements entirely feasible.

Therefore, it is expected that a replication standard published in the very early years of the next decade will be widely implemented by SQL product vendors and that those implementations will continue for at least another decade (this prediction being based on the longevity of existing custom-written replication systems implemented as long as a decade ago). While it should be expected that additional replication-related facilities will be discovered and specified during the coming decade, they will be compatible with the facilities specified in the proposed standard.

5. Program of Work

Specify an incremental part to ISO 9075, or if appropriate multiple parts, to define additional facilities—including concepts, syntax, and semantics—that respond to market requirements for SQL database support for replication capabilities, with the intent of publishing this part in 2001.

5.1. Included Scope (Note: Facilities will be standardized in appropriate SC 32 documents)

Such facilities may include, but are not necessarily limited to, features such as:

- *Autonomy.* Local databases should be loosely connected and capable of operating independently even when temporarily disconnected from each other.
- *Conflict prevention and handling.* In an asynchronous distributed environment, competing changes could made to the same data in two local databases and should result in a conflict during propagation. Conflict issues include accounting for various categories and types of conflicts, and specifying suitable conflict resolution facilities.
- *Data subset definition and management.* Most of the applications listed in §2.2 require the ability to store subsets of the entire database at some locations (for example, on constrained PDA or embedded platforms), rather than having the entire database at all locations.
- *Asynchronous distributed transaction model.* Transaction requirements should be defined, and transaction facilities provided, in a way that provides useful guarantees even in a replicated environment with propagation delays.
- *Protocol.* A suitable communications data protocol for performing replication between local databases supplied by various conforming vendors.
- *Transformation and mapping.* Some of the applications listed in §2.2 can require the existence of dissimilar schemas in different databases within the same distributed system (for example, in warehousing and application integration situations). This creates a requirement for facilities to express business and mapping rules across schema implementations.
- *SQL language.* A key design goal is transparency to user applications, and so it is anticipated that there will be no impact on user query and update operations on the individual local databases. On the other hand, the DDL should be extended to allow specification of additional metadata describing subsetting and transformation rules (for example), which can be done according to application business rules that are to be expressed in terms of tables, columns, primary keys, foreign keys, and existing SQL functions, thus avoiding product-specific engineering details.

5.2. Excluded Scope

The following areas would be explicitly excluded and out of scope of the proposed work:

- *Synchronous approaches.* N-phase commit and similar approaches are already well-understood and mature and do not require standardization. Further, they do not solve the problems intended to be addressed by this work.
- *Distributed queries.* Facilities for allowing one client query to be performed across multiple databases are already provided by other standards, including SQL/MED.

6. Justification for Subproject Request

Various criteria have been specified for the approval of program extensions (that is, subdivisions and minor enhancements) of existing projects. Those criteria relative to this effort are satisfied as follows:

1. The rationale for the SQL/Replication extension of Database Language SQL is given in the “Needs” and “Existing Practice” paragraphs above. This proposed work is within the scope of already-authorized work in the SQL project description.
2. Consensus on the need to do this work is evident from activity in the industry and the voting on this paper.
3. NCITS TC H2 and ISO/IEC SC32/WG3 are uniquely qualified to address and resolve issues specific to SQL database implementations.
4. The SQL editor, Mr. Jim Melton, is willing to be the subproject editor for Database Language SQL — Replication (SQL/Replication).

– End of paper –