ISO/IEC JTC 1/SC 32 N **1289**

ISO/IEC JTC 1/SC 32

Data Management and Interchange

Secretariat: United States of America (ANSI)
Administered by Farance Inc. on behalf of ANSI

DOCUMENT TYPE	Other Document (Open)
TITLE	USA Comments on CD 9075-1, -2, -3, -4, -9, -10, -11, and -13
SOURCE	Jim Melton (USA)
PROJECT NUMBER	01.32.03.06.01.00
STATUS	USA National Body Position
REFERENCES	
ACTION ID.	FYI
REQUESTED ACTION	
DUE DATE	
Number of Pages	15
LANGUAGE USED	English
DISTRIBUTION	P & L Members
	SC Chair
	WG Conveners and Secretaries

File: H2-2005-084R1-USA-comments-CD2004-sql2007.doc

Printed: Monday, 14 March, 2005 at 15:25

INCITS H2-2005-084R1
ISO/IEC JTC 1 / SC 32 / WG3:TXL-040
10 March, 2005 — 2005-03-10

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: 01.32.03.06 (ISO) and 1234D (ANSI)

Title: USA Comments on CD 9075-1, -2, -3, -4, -9, -10, -11, and -13

Status: USA National Body position

Author: Jim Melton (Ed.)

References:

1) WG3:TXL-002 = H2-2004-004 = 32N1198, *ISO/IEC CD 9075-1, Information technology — Database languages SQL – Part 1: Framework (SQL/Framework)*

2) WG3:TXL-003 = H2-2004-005 = 32N1199, *ISO/IEC CD 9075-2, Information technology — Database languages — SQL – Part 2: Foundation (SQL/Foundation)*

3) WG3:TXL-004 = H2-2004-006 = 32N1201, *ISO/IEC CD 9075-3, Information technology — Database languages — SQL – Part 3: Call-Level Interface (SQL/CLI)*

4) WG3:TXL-005 = H2-2004-007 = 32N1202, *ISO/IEC CD 9075-4, Information technology — Database languages — SQL – Part 4: Persistent Stored Modules (SQL/PSM)*

5) WG3:TXL-006 = H2-2004-008 = 32N1203, *ISO/IEC CD 9075-9, Information technology — Database languages — SQL – Part 9: Management of External Data (SQL/MED)*

6) WG3:TXL-007 = H2-2004-009 = 32N1204, *ISO/IEC CD 9075-10, Information technology — Database languages — SQL – Part 10: Object language bindings (SQL/OLB)*

7) WG3:TXL-008 = H2-2004-010 = 32N1205, *ISO/IEC CD 9075-11, Information technology — Database languages — SQL – Part 11: Information and Definition Schemas (SQL/Schemata)*

8) WG3:TXL-009 = H2-2004-011 = 32N1206, *ISO/IEC CD 9075-13, Information technology — Database languages — SQL – Part 13: SQL Routines and Types for the Java Programming Language (SQL/JRT)*

Discussion

This summary is provided to assist INCITS H2 to be maximally effective in establishing the USA position and comments submitted on the referenced ballots. The columns in the document have the following meanings:
- **SEQ #** — A sequence number indicating the total number of comments
- **Cmnt ID** — The comment identifier assigned by the organization submitting the comment
- **See Also** — Pointers to (comment identifiers of) other comments that appear to be closely related
- **Severity** — The severity of the problem as stated by the organization submitting the comment
- **Reference** — The location in the balloted document that the organization identified as most pertaining to the comment
- **Description** — The description provided by the organization with the comment; this includes any solution provided by the organization
- **Addressed By** — "Comment" if a solution was provided by the organization; a paper number if a known proposal addresses the comment; blank otherwise.

Comment Breakdown

The proposed ballot comments are distributed as follows.

SQL/Framework						
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	1	0	0	0	**1**	N†
Total	**1**	**0**	**0**	**0**	**1**	

* DNV → Did Not Vote

† N → No (every No vote must be accompanied by comments)

‡ Y → Yes without comments

¤ YWC → Yes with comments

+ A → Abstain

SQL/Foundation						
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	16	0	6	5	**27**	N†
Total	**16**	**0**	**6**	**5**	**27**	

SQL/CLI						
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	1	0	1	0	**2**	N†
Total	**1**	**0**	**1**	**0**	**2**	

SQL/PSM						
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	2	0	1	0	**3**	N†
Total	**2**	**0**	**1**	**0**	**3**	

SQL/MED					
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	1	0	0	0	1	N†
Total	**1**	**0**	**0**	**0**	**1**	

SQL/OLB					
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	1	0	0	0	1	N†
Total	**1**	**0**	**0**	**0**	**1**	

SQL/Schemata					
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	1	0	0	0	1	N†
Total	**1**	**0**	**0**	**0**	**1**	

SQL/JRT					
Participant	**Major Technical**	**Minor Technical**	**Major Editorial**	**Minor Editorial**	**Number of Comments**	**Vote**
USA	1	0	0	0	1	N†
Total	**1**	**0**	**0**	**0**	**1**	

Consolidated Ballot Comments

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					SQL/Framework	
	USA-P01-999		1-Major Technical	*P01-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved. **Solution** None provided with comment.	
					SQL/Foundation	
	USA-P02-001		3-Major Editorial	*P02-00.00, Foreword*	The 6th paragraph seems to be out of date. **Solution** Modify the 6th paragraph as shown here: This **sixth** ~~fifth~~ edition cancels and replaces the **fifth** ~~fourth~~ edition (ISO/IEC 9075:**2003** ~~1999~~). This problem might (probably does) extend to other parts as well. A better solution might be to move (not replicate) the edition information in a single list in Framework.	See comment
	USA-P02-010		4-Minor Editorial	*P02-04.18.03, Known functional dependencies in a base table*	In the 5th paragraph, replace "parameteric" with "parametric" **Solution** Provided with comment.	See comment
	USA-P02-020		1-Major Technical	*P02-05.04, Names and identifiers*	SQL restricts a <descriptor name> to <simple value specification> (<literal>, <host parameter names>, <SQL parameter references>, and <embedded variable name>). We believe an <identifier> should also be a valid choice for a <descriptor name>, just as an <identifier> is a valid choice for a <statement name> and a <cursor name>. **Solution** None provided with comment.	
	USA-P02-030		1-Major Technical	*P02-07.13, <query expression>*	Sometimes it is necessary/sufficient for an application to only retrieve the first n rows of a query (where $n > 0$). Therefore an option on <query expression> is necessary to indicate that only a fixed number of rows should be returned. We suggest syntax such as "FETCH FIRST n ROWS ONLY", "LIMIT TO n ROWS", or "TOP n" to accomplish this. The order of the rows that are the result of a <query expression> is implementation-dependent and therefore two executions of the same <query	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By

					expression>s including this new functionality would be non-deterministic. Thus, another option is needed to order the result before the first *n* rows are retrieved. This option would be the <order by clause> as it is currently defined for a <cursor specification>. Of course, specifying either one or both of the new options makes the <query expression> read-only (*i.e.*, not updatable).
					
					Many times only the very first row is important. For this case, additional syntactic sugar would be appropriate.
					
					This could be accomplished by modifying the syntax of <query expression> as follows:
					`<query expression> ::=`
					`    [ <with clause> ] <query expression body>`
					`    [ [ <order by clause> ] <row counting clause> ]`
					
					`<row counting clause> ::= Syntax to be determined`
					 Solution None provided with comment.	

	USA-P02-035		1-Major Technical	*P02-08.06, <similar predicate>*	The <similar predicate> was proposed in 1988 in a paper known as CPH-37a. Over the next few years, the predicate was refined into what we see today, but the basis for its design was always the regular expression syntax specified in Unix and in the Posix standard that was then popular. Unfortunately, in later years, the Posix standard has been viewed as not very successful, and its regular expression design has not stood the test of time. Most modern standards and other specifications (notably the W3C's XML Schema and, by extension, XQuery) have chosen to use a different regular expression "standard": that of the Perl language (version 5 seems most popular). The differences between Posix regular expressions and Perl regular expressions are not huge, but they are important differences. It is undoubtedly true that SQL implementations have faithfully implemented the <similar predicate> as specified, using Posix-style regular expressions. However, there is visible market demand for Perl-style regular expressions. Either the definition of the <similar predicate> should be replaced by a definition that uses Perl-style regular expressions (or a corresponding subset thereof), or an additional predicate (or even more than one predicate) should be defined to provide Perl-style regular expressions, or the syntax of the <similar predicate> should be enhanced to allow an application program to specify which of the two styles of regular expressions it wishes to use.	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					Solution	
					None provided with comment.	
	USA-P02-040		1-Major Technical	*P02-11.12, <alter column definition>*	While it is possible to modify the properties of identity columns, similar functionality is missing for generated columns. Of course, consideration would have to be given to possible side effects, such as what would happen to values of generated columns stored in existing rows in tables.	
					Solution	
					None provided with comment.	
	USA-P02-050		1-Major Technical	*P02-11.12, <alter column definition>*	It should be possible to modify the data type of a column under certain conditions. One criterion should be that the existing and new data types are compatible. Another one should be that the size, precision and scale, where applicable, of the new data type are at least as large as the ones of the existing data type. For example, it should be possible to increase the size of a VARCHAR column (on the other hand, decreasing the size should not be allowed). Other possible restrictions need to be considered as well.	
					Solution	
					None provided with comment.	
	USA-P02-060		1-Major Technical	*P02-11.39, <trigger definition>*	Not all views are updatable. However every view could be made updatable if an INSTEAD OF trigger were defined for it. Similar to the existing BEFORE and AFTER triggers on base tables, an INSTEAD OF trigger could be defined for views for INSERT, UPDATE, and DELETE operations. However, for INSTEAD OF triggers it is sufficient to be on a per row basis and no more than one such trigger for each action of insert, update, and delete.	
					Solution	
					None provided with comment.	
	USA-P02-070		4-Minor Editorial	*P02-13.04, Calls to an <externally-invoked procedure>*	Modify General Rule 3) e) as shown here: e) If *DT* identifies INT, DEC, or REAL and the caller language of *EP* is M, then a reference to *PN* that assigns some value *SV* to *PN* implicitly assigns the value CAST (*SV* AS CHARACTER VARYING(*ML*)) to *PI*, where *ML* is the implementation-defined maximum length of variable-length ~~of~~ character strings.	See comment
					Solution	
					Provided with comment.	
	USA-P02-080		1-Major Technical	*P02-14.09, <merge statement>*	Currently, the syntax of MERGE statement allows either updating the existing rows in the target table or inserting new rows into the target table, by branching to either <merge when matched clause> or <merge when not matched clause> based on the truth value of a <search condition>. We believe it would be useful to allow multiple insert/update operations based on the truth value of additional	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<search condition> in both <merge when matched clause> and <merge when not matched clause>. **Solution** None provided with comment.	
	USA-P02-090		3-Major Editorial	*P02-14.11, <update statement: searched>*	Modify Conformance Rule 1) as shown here: 1) Without Feature F781, "Self-referencing operations", conforming SQL language shall not contain an <update statement: **searched** ~~positioned~~> in which a leaf generally underlying table of *T* is an underlying table of any <query expression> generally contained in the <search condition>. **Solution** Provided with comment.	See comment
	USA-P02-100		3-Major Editorial	*P02-19.06, <prepare statement>*	General Rule 6) contains the phrase: the implementation-defined maximum ~~value of <length> for the CHARACTER VARYING data type~~ The standard usually says: implementation-defined maximum **length of variable-length character strings** **Solution** Provided with comment.	See comment
	USA-P02-110		3-Major Editorial	*P02-19.06, <prepare statement>*	There seems to be a major problem with GR 6) a) xv) that starts off "If *DP* is the <cast operand> …" and GR 6) a) xvi) that starts off "The General Rules of Subclause 14.22 …", perhaps caused by an editorial mishap somewhere along the way. Our research indicates that GR 6) a) xvi) was actually part of GR 6) a) xv) until 5WD-02-Foundation-2002-12R1.pdf, but got split off into a separate rule (with the addition of some possibly spurious text) by the time 5WD-02-Foundation-2003-12R1.pdf came along. Could the Editor please research this bug further and restore the correct rule? **Solution** None provided with comment.	
	USA-P02-150		3-Major Editorial	*P02-22.01, <get diagnostics statement>*	Modify the lead-in of GR 6) a) ii) 1) as shown here (see also SIA-025r1): 1) Let *S* be … **Solution** Provided with comment.	See comment
	USA-P02-120		4-Minor Editorial	*P02-22.01, <get diagnostics statement>*	Modify Syntax Rule 2) as shown here: 2) The declared type of <all info target> shall be a character **string** type. **Solution** Provided with comment.	See comment
	USA-P02-130		4-Minor	*P02-22.01, <get*	General Rule 6) contains the phrase:	See comment

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
			Editorial	*diagnostics statement>*	implementation-defined maximum ~~value for the <length> contained in a <data type>~~ The standard usually says: implementation-defined maximum **length of variable-length character strings** **Solution** Provided with comment.	
	USA-P02-140		4-Minor Editorial	*P02-22.01, <get diagnostics statement>*	Delete one of the two "where"s in GR 6) a) i). **Solution** Provided with comment.	See comment
	USA-P01-999		1-Major Technical	*P02-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved. **Solution** None provided with comment.	
	USA-P02-180		1-Major Technical	*P02-No specific location*	SQL should provide a way to insert/update/delete rows from a table and to retrieve the rows that were inserted/updated/deleted as a single operation. See also Language Opportunity FND-849, which asks for a similar functionality. **Solution** None provided with comment.	
	USA-P02-190		1-Major Technical	*P02-No specific location*	Applications have a need to temporarily disable (table) constraints (across transaction boundaries). For example, a user may want to disable constraints because he is going to be reloading data from multiple sites and he does not want to go through the work of making sure that the data across N tables is loaded in the appropriate order. Though the standard has the ability to defer constraint checking until the end of a transaction this would not be a complete solution as most users would not want to perform the entire set of loads all under one gigantic transaction and in the case of loads from multiple data sources, it is likely impractical if not impossible. So the standard should provide syntax to let the user specify whether a given constraint is enforced or not enforced. **Solution** None provided with comment.	
	USA-P02-200		1-Major Technical	*P02-No specific location*	Many applications have a need to deal with binary data. Not always are BINARY LARGE OBJECTs (BLOBs) appropriate for these applications since BLOBs have certain restrictions, such as they are not allowed in <general set function>, <group by clause>, and <order by clause>, etc. For such applications, it would be beneficial if the standard supports a "regular" binary string type that	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					does not have the restrictions associated with BLOB type. Just as the standard has three kinds of character string types: CHAR, VARCHAR, and CLOB, we believe the standard should support three kinds of binary types: BINARY, VARBINARY, and BLOB (all measure in lengths of octets). Note that this new data type needs to avoid the problems that led to the removal of the BIT and BIT VARYING data types. **Solution** None provided with comment.	
	USA-P02-210		1-Major Technical	*P02-No specific location*	Exact numeric types are used when exact results are desired for arithmetic operations, but they suffer from the fact that the range of values supported by these types is much smaller than the range supported by approximate numeric types. The upcoming revision of ANSI/IEEE standard 754, "IEEE Standard for Binary Floating-Point Arithmetic", includes a new kind of exact numeric type called "Decimal Floating Point" that offers a much bigger range of values for a given precision while providing exact results for arithmetic operations. Many popular programming languages are in the process of adding this new type to their type systems. We believe SQL should also add this new type to the list of predefined types. **Solution** None provided with comment.	
	USA-P02-220		1-Major Technical	*P02-No specific location*	When a user is granted one or more roles, he may expect to be allowed to access not only those objects on which he has been granted privileges, but also the privileges granted to every role with which he is associated. Unfortunately, this is not currently the case in SQL. When a user starts an SQL-session, he is not allowed to access any of the objects on which a role he has been granted has privileges (unless, of course, he has the privilege directly). If he wants to access objects on which a role he has been granted has privileges, he has to first perform a SET ROLE statement. It would be useful to have the ability to specify that a ROLE is enabled by default. This would prevent a user from having to execute a SET ROLE statement to get the an initial ROLE. **Solution** None provided with comment.	
	USA-P02-225		1-Major Technical	*P02-No specific location*	The current definition of ROLEs supports the ability to use a SET ROLE statement for only one role at a time. There are situations where it would be useful to allow a user to enable multiple roles concurrently. **Solution** None provided with comment.	
	USA-P02-235		1-Major Technical	*P02-No specific location*	Currently it is possible to grant a user both a "Teller" role and an "Auditor" role even though the two roles would be considered mutually exclusive in most	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					organizations. Similarly, it is possible to grant a user an "Auditor" role even if that user already has the "Teller" role. We believe that SQL should provide a mechanism to prevent a user from acquiring or setting such mutually exclusive roles, similar to what the ANSI standard on Role Based Access Control, ANSI INCITS 359:2004 refers to as "static separation of duty relations". **Solution** None provided with comment.	
	USA-P02-240		1-Major Technical	*P02-No specific location*	The current security model of SQL is based on granting/revoking privileges to users to manipulate entire database objects. However, SQL does not provide any sort of mandatory access control—that is, there is no way in the current security model to restrict access to subsets of data in database objects such as tables, based on labels such as "Top Secret", "Board of Directors only", etc. only to those users that possess appropriate authorizations. Such an extended security model is a requirement in many organizations. We believe the security model of SQL should be extended to offer such a capability. **Solution** None provided with comment.	

	USA-P02-160		3-Major Editorial	*P02-No specific location*	Subclause 9.8, "Determination of identical values", General Rule 2) reads (with emphasis added by the comment author):
					2) Case:
					  a) If *V1* and *V2* **are both null**, then *V1* is identical to *V2*.
					  b) If *V1* **is null** and *V2* **is not null**, or if *V1* is not null and *V2* is null, then *V1* is not identical to *V2*.
					  c) …
					This GR and other rules in Foundation and other parts use the terms "is/are null", "is not null", *etc.* loosely. What is preferable is to use the terms "is the null value", *etc.* instead.
					One needs to carefully examine Foundation and possibly all other parts to replace all these offending phrases.
					Solution
					Provided with comment.	

SQL/CLI						

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
	USA-P03-010	USA-P04-010	3-Major Editorial	*P03-02.01, JTC1 standards*	The references for [Framework], [Foundation], and [Schemata] are out-of-date. **Solution** None provided with comment.	
	USA-P03-999		1-Major Technical	*P03-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved. **Solution**	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					None provided with comment.	
			=...		**SQL/PSM**	
	USA-P04-010	USA-P03-010	3-Major Editorial	*P04-02.01, JTC1 standards*	The references for [Framework], [Foundation], and [Schemata] are out-of-date. **Solution** None provided with comment.	

	USA-P04-020		1-Major Technical	*P04-13.13. <for statement>*	Modify Syntax Rule 7) as shown here (see also STX-041):
					7) Let *COMMON_CODE* be:
					`    ...`
					`    BEGIN NOT ATOMIC`
					`    OPEN CN;`
					`BL:``LOOP`
					`        OPEN CN;`
					`        BEGIN NOT ATOMIC`
					`    ...`
					Solution
					Provided with comment.
						See comment

	USA-P04-999		1-Major Technical	*P04-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved. **Solution** None provided with comment.	
					SQL/MED	
	USA-P09-999		1-Major Technical	*P09-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved. **Solution** None provided with comment.	
					SQL/OLB	
	USA-P10-999		1-Major Technical	*P10-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved. **Solution** None provided with comment.	
					SQL/Schemata	
	USA-P11-999		1-Major Technical	*P10-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved.	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					Solution	
					None provided with comment.	
				SQL/JRT		
	USA-P13-999		1-Major Technical	*P10-No specific location*	All Possible Problems and Editor's Notes must be satisfactorily resolved and all problems discovered during the course of the ballot resolution process must be satisfactorily resolved.	
					Solution	
					None provided with comment.	

Cross Reference by CmntID

Cmnt ID	SEQ #

===========================
End of paper
===========================