

ITALY

Lack of Experts

National Body CAN Comments — 2005-02-03

32N1198, ISO/IEC CD 9075-01 Information technology - Database Languages - SQL - Part 1: Framework (SQL/Framework)

32N1199, ISO/IEC CD 9075-02 Information technology - Database Languages - SQL - Part 2: Foundation (SQL/Foundation)

32N1201, ISO/IEC CD 9075-03 Information technology - Database Languages - SQL - Part 3: Call-Level Interface (SQL/CLI)

32N1202, ISO/IEC CD 9075-04 Information technology - Database Languages - SQL - Part 4: Persistent Stored Modules (SQL/PSM)

32N1203, ISO/IEC CD 9075-09 Information technology - Database Languages - SQL - Part 9: Management of External Data (SQL/MED)

32N1204, ISO/IEC CD 9075-10 Information technology - Database Languages - SQL - Part 10: Object language bindings (SQL/OLB)

32N1205, ISO/IEC CD 9075-11 Information technology - Database Languages - SQL - Part 11: Information and Definition Schemas (SQL/Schemata)

32N1206, ISO/IEC CD 9075-13 Information technology - Database Languages - SQL - Part 13: SQL Routines and Types Using the Java™ Programming Language (SQL/JRT)

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
CD SQL/Framework						
	CAN-P01-001		1-Major Technical	<i>P01-No specific location</i>	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.	
CD SQL/Foundation						
	CAN-P02-001		1-Major Technical	<i>P02-No specific location</i>	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.	
CD SQL/CLI						
	CAN-P03-001		1-Major Technical	<i>P03-No specific location</i>	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.	
CD SQL/PSM						
	CAN-P04-001		1-Major Technical	<i>P04-No specific location</i>	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
					None provided with comment. Solution	
CD SQL/MED						
	CAN-P09-001		1-Major Technical	<i>P09-No specific location</i>	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.	
CD SQL/OLB						
	CAN-P10-001		1-Major Technical	<i>P10-No specific location</i>	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.	
CD SQL/Schemata						
	CAN-P11-001		1-Major Technical	<i>P11-No specific location</i>	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.	
CD SQL/JRT						
	CAN-P13-001		1-Major Technical	<i>P13-No specific location</i>	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.	

ISO/IEC JTC1/SC32/WG3:TXL-034 – DIN NI-32 N 0642

Authoritative Version: Adobe Acrobat Portable Document Format (PDF)



ISO

International Organization for Standardization

DIN

Deutsches Institut für Normung

Din NI-32

Database

ISO/IEC JTC 1/SC 32

Data Management and Interchange

WG 3

Database Languages

Title: German Comments on SC32 N 1156: ISO/IEC CD 9075-1, 2, 3, 4, 9, 10, 11, 13:200x(E)

Status: Consolidated comments to assist with resolution of CD ballot comments

Author: Jörn Bartels (Germany)

National Body DEU Comments — 2005-03-08

SEQ #	Cmn t ID	See Als o	Severity	Reference	Description	Addressed By
ISO/IEC FCD 9075-01:200x(E) SQL/Framework						
1	DEU-P01-010		1-Major Technical	<i>P01-No specific location</i>	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.	
ISO/IEC FCD 9075-02:200x(E) SQL/Foundation						
2	DEU-P02-010		1-Major Technical	<i>P02-No specific location</i>	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.	
3	DEU-P02-020		1-Major Technical	<i>P02-07.6 Table reference</i>	Queries of the form SELECT ... FROM <joined table> Do not seem to be supported anymore. This is due to changes proposed in DRS-077. Solution None provided with comment.	
ISO/IEC FCD 9075-03:200x(E) SQL/CLI						
4	DEU-P03-010		1-Major Technical	<i>P03-No specific location</i>	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.	
ISO/IEC FCD 9075-04:200x(E) SQL/PSM						
5	DEU-P04-010		1-Major Technical	<i>P04-No specific location</i>	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.	
6	DEU-P11-020	DEU-P11-030	1-Major Technical	<i>P04-18.2 "MODULE_PRIVILEGES" Table</i>	The table MODULE_PRIVILEGES stores the privileges granted on a specific module. The same information could be stored in the table USAGE_PRIVILEGES. This would simplify the definition schema and standardise the way, how privileges are stored. Solution None provided with comment.	
7	DEU-P11-030	DEU-P11-020	1-Major Technical	<i>P04-18.2 "MODULE_PRIVILEGES" Table</i>	The constraint MODULE_PRIVILEGE_GRANTOR_CHECK and MODULE_PRIVILEGE GRANTEE_CHECK reference still the tables ROLES and USERS. They are gone!	

SEQ #	Cmn t ID	See Also	Severity	Reference	Description	Addressed By
					<p align="center">Solution</p> <p>Replace the check constraints with a foreign key on AUTHORIZATIONS.</p>	
ISO/IEC FCD 9075-09:200x(E) SQL/MED						
8	DEU-P09-010		1-Major Technical	<i>P09-No specific location</i>	<p>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.</p> <p align="center">Solution</p> <p>None provided with comment.</p>	
9	DEU-P09-020		2-Minor Technical	<i>P09-25 Definition Schema</i>	<p>Some tables which are introduced by this clause have for most columns no NOT NULL constraint, where it should be defined. This are at least 25.4 FOREIGN_DATA_WRAPPERS base table, 25.8 FOREIGN_TABLES base table and 25.10 ROUTINE_MAPPINGS base table</p> <p align="center">Solution</p> <p>None provided with comment.</p>	
10	DEU-P09-030		3-Major Editorial	<i>P09-25.2 DATA_TYPE_DESCRIPTOR" table</i>	<p>The constraint DATA_TYPE_DESCRIPTOR_DATA_TYPE_CHECK_COMBINATIONS is entirely replaced. This leads to problems of desynchronisation with SQL/Schemata. It does also not allow modifications from other parts (like SQL/XML) of the standard.</p> <p>The constraint does also currently not check the NULL applicability of the columns, as described in Description 2), which are inserted by this constraint.</p> <p align="center">Solution</p> <p>None provided with comment.</p>	
11	DEU-P09-040		3-Major Editorial	<i>P09-25.2 DATA_TYPE_DESCRIPTOR" table</i>	<p>The Descriptions 2) and 3) are in conflict with each other. They describe both the NULLability of the newly introduced columns. There is a conflict if both come to different results.</p> <p align="center">Solution</p> <p>The Descriptions 2) and 3) should be merged.</p>	
12	DEU-P09-050		2-Minor Technical	<i>P09-25.4 "FOREIGN_DATA_WRAPPERS" table</i>	<p>There is no constraint, which verifies the existence of the catalog and the authorization Identifier, which is used.</p> <p align="center">Solution</p> <p>None provided with comment.</p>	
13	DEU-P09-060		2-Minor Technical	<i>P09-25.6 "FOREIGN_SERVERS" table</i>	<p>There is no constraint, which verifies the existence of the catalog and the authorization Identifier, which is used.</p> <p align="center">Solution</p> <p>None provided with comment.</p>	
14	DEU-P09-070		2-Minor Technical	<i>P09-25.12 "TABLES" table</i>	<p>There is no constraint, which verifies that for a FOREIGN table there is also an entry in the table FOREIGN_TABLES. This could be done as it is done already in constraint TABLES_CHECK_NOT_VIEW of the table TABLES.</p>	

SEQ #	Cmn t ID	See Also	Severity	Reference	Description	Addressed By
					Solution None provided with comment.	
15	DEU-P09-080		2-Minor Technical	P09-25.13 "USAGE_PRIVILEGES" table	<p>There is no constraint added, which checks the values of OBJECT_CATALOG and OBJECT_NAME as described in Description 1)</p> <p>There is also no modification of the constraint USAGE_PRIVILEGES_CHECK_REFERENCES_OBJECT which allows OBJECT_SCHEMA to be the empty string.</p> <p>It is currently also not allowed, that the OBJECT_TYPE is anything except 'DOMAIN', 'CHARACTER SET', 'COLLATION', 'TRANSLATION', 'SEQUENCE'. The use for a foreign-data wrapper or a foreign server requires a modification of constraint USAGE_PRIVILEGES_OBJECT_TYPE_CHECK.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
16	DEU-P09-090		2-Minor Technical	P09-25.15 "USER_MAPPINGS" table	<p>There is no foreign key check for the column AUTHORIZATION_IDENTIFIER.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
ISO/IEC FCD 9075-10:200x(E) SQL/OLB						
17	DEU-P10-010		1-Major Technical	P10-No specific location	<p>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.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
ISO/IEC FCD 9075-11:200x(E) SQL/Schemata						
18	DEU-P11-010		1-Major Technical	P11-No specific location	<p>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.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
19	DEU-P11-020		1-Major Technical	P11-No specific location	<p>It is not clear, which tables should have a direct or indirect relationship to the table SCHEMATA. For some tables is a foreign key defined, for some is a check constraint defined, which checks the foreign key relationship only when there are schemas in the same catalog.</p> <p>There should be an explanation for this distinction and all relationships need to be checked for correctness.</p> <p>An example for a dubious relationship is the constraint TRIGGERS_REFERENCES_TABLES. Is it really possible, to define a trigger on a table of another catalog. This is especially strange, as the table</p>	

SEQ #	Cmn t ID	See Also	Severity	Reference	Description	Addressed By
					TRIGGERED_UPDATE_COLUMNS has a direct foreign key to COLUMNS. Solution None provided with comment.	
20	DEU-P11-030		3-Major Editorial	P11-5.54 Short name views	The View definitions in 5.54 Short name views should be sorted according to the order of the base views (i.E. position of CONSTR_COL_USAGE). Solution Order them according to base view order.	
21	DEU-P11-040		2-Minor Technical	P11-5.78 "SQL_LANGUAGES" View	The View SQL_LANGUAGES is deprecated. In Note 6 is a reference to SQL/Framework Subclause 6.4. There is a backwards reference, in Note 9, which says "The equivalent information is available to the SQL user in the Information Schema." When we delete the view, it is not clear if the Note 9 should also be deleted. Solution None provided with comment.	
22	DEU-P11-050		2-Minor Technical	P11-6.9 "CHARACTER_ENCODING_FORMS" Table	The NOT NULL Constraints are not needed, as all columns are part of the primary key. Solution Delete the NOT NULL constraints.	
23	DEU-P11-060		2-Minor Technical	P11-6.10 "CHARACTER_REPERTOIRES" Table	The NOT NULL Constraints on the column CHARACTER_REPERTOIRE_NAME is not needed, as the column is part of the primary key. Solution Delete the NOT NULL constraint.	
24	DEU-P11-070		4-Minor Editorial	P11-6.11 "CHARACTER_SETS" Table	The column NUMBER_OF_CHARACTERS is in the last Edition of the Standard deprecated and should now be deleted. Solution Delete the column. Do the same in the View Definition 5.12 CHARACTER_SETS view and in 5.78 Short name views in the view CHARACTER_SETS_S. Delete the corresponding List Elements 1) and 2) in Annex C.	
25	DEU-P11-080		2-Minor Technical	P11-6.16 "COLLATIONS" Table	There is no constraint for the column CHARACTER_REPERTOIRE_NAME defined. It needs to reference the Table CHARACTER_REPERTOIRES. Solution Add the constraint COLLATIONS_FOREIGN_KEY_CHARACTER_REPERTOIRES FOREIGN KEY (CHARACTER_REPERTOIRES) REFERENCES CHARACTER_REPERTOIRES.	
26	DEU-P11-090		4-Minor Editorial	P11-6.16 "COLLATIONS" Table	The columns COLLATION_TYPE, COLLATION_DICTIONARY, and COLLATION_DEFINITION are in the last edition of the Standard deprecated and should now be deleted. Solution Delete the columns. Do the same in the View Definition 5.15 COLLATIONS view and in 5.78 Short name views in the view COLLATIONS_S. Delete the corresponding List Elements 3) and 4) in Annex C.	

SEQ #	Cmn t ID	See Also	Severity	Reference	Description	Addressed By
27	DEU-P11-100		2-Minor Technical	P11-6.16 "COLLATIONS" Table	The column PAD_ATTRIBUTE has no NOT NULL check constraint, even that there is in the description no explanation of the meaning of a possible NULL value. Solution None provided with comment.	
28	DEU-P11-110		2-Minor Technical	P11-6.20 "COLUMNS" Table	For the columns IS_GENERATED and IDENTITY_GENERATION are no check constraints specified, but in the description are Lists of allowed values. Solution Add to the column IS_GENERATED the following column level constraint: CONSTRAINT COLUMNS_IS_GENERATED_CHECK CHECK (IS_GENERATED in ('NEVER', 'ALWAYS')) Add to the column IDENTITY_GENERATION the following column level constraint: CONSTRAINT COLUMNS_IDENTITY_GENERATION_CHECK CHECK (IDENTITY_GENERATION IN ('ALWAYS', 'BY DEFAULT'))	
29	DEU-P11-120		2-Minor Technical	P11-6.21 "DATA_TYPE_DESCRIPTOR" Table	The constraint DATA_TYPE_DESCRIPTOR_FOREIGN_KEY_SCHEMATA assures that the values of USER_DEFINED_TYPE_CATALOG and USER_DEFINED_TYPE_SCHEMA have corresponding rows in the table SCHEMATA. The constraint DATA_TYPE_DESCRIPTOR_CHECK_REFERENCES_UDT allows that the value for the column USER_DEFINED_TYPE_CATALOG has no corresponding row in SCHEMATA. As this is not possible according to the first constraint, we could rewrite this constraint as a foreign key. It is not clear, if this is intended. Solution None provided with comment.	
30	DEU-P11-130		2-Minor Technical	P11-6.21 "DATA_TYPE_DESCRIPTOR" Table	The columns SCOPE_CATALOG, SCOPE_SCHEMA, and SCOPE_NAME are not checked against the possible values in the table TABLES. Solution None provided with comment.	
31	DEU-P11-140		4-Minor Editorial	P11-6.21 "DATA_TYPE_DESCRIPTOR" Table	The constraint DATA_TYPE_DESCRIPTOR_CHECK_OBJECT_TYPE should be a column constraint, as it references only the column OBJECT_TYPE. Solution Remove the preceding comma.	
32	DEU-P11-150		4-Minor Editorial	P11-6.21 "DATA_TYPE_DESCRIPTOR" Table	In the constraint DATA_TYPE_DESCRIPTOR_CHECK_REFERENCES_COLLATION_CHARACTER_SET_APPLICABILITY should be a comma "," at the end of the 10 th line of the constraint. Solution Add the missing comma.	
33	DEU-P11-		2-Minor Technical	P11-6.24 "DOMAIN_CONSTRAINTS" Table	Should there be a NOT NULL check constraint for the columns IS_DEFERRABLE and INITIALLY_DEFERRED?	

SEQ #	Cmn t ID	See Also	Severity	Reference	Description	Addressed By
	160				In the description is no explanation of the meaning of a possible null value. Solution None provided with comment.	
34	DEU-P11-170		2-Minor Technical	P11-6.31 "PARAMETERS" Table	The foreign key constraint PARAMETERS_FOREIGN_KEY_SCHEMATA does not check the name of the routine. This constraint should be removed and instead there should be a foreign key to ROUTINES be defined. Solution None provided with comment.	
35	DEU-P11-180		2-Minor Technical	P11-6.31 "PARAMETERS" Table	There needs to be a unique constraint defined, which guarantees the uniqueness of a parameter name for a routine. Solution Add the constraint: CONSTRAINT PARAMETERS_UNIQUE_CHECK (UNIQUE SPECIFIC_CATALOG, SPECIFIC_SCHEMA, SPECIFIC_NAME, PARAMETER_NAME)	
36	DEU-P11-190		2-Minor Technical	P11-6.31 "PARAMETERS" Table	For the columns FROM_SQL_SPECIFIC_CATALOG, FROM_SQL_SPECIFIC_SCHEMA, and FROM_SQL_SPECIFIC_NAME and TO_SQL_SPECIFIC_CATALOG, TO_SQL_SPECIFIC_SCHEMA, and TO_SQL_SPECIFIC_NAME is no foreign key check defined. Solution None provided with comment.	
37	DEU-P11-200		1-Major Technical	P11-6.36 "ROUTINE_PRIVILEGES" Table	The table ROUTINE_PRIVILEGES stores the privileges granted on a specific routine. The same information could be stored in the table USAGE_PRIVILEGES. This would simplify the definition schema and standardise the way, how privileges are stored. Solution None provided with comment.	
38	DEU-P11-210		2-Minor Technical	P11-6.41 "SCHEMATA" Table	For the columns DEFAULT_CHARACTER_SET_CATALOG, DEFAULT_CHARACTER_SET_SCHEMA and DEFAULT_CHARACTER_SET_NAME is a foreign key referencing the table CHARACTER_SETS missing. Solution Add the missing Foreign Key constraint: CONSTRAINT SCHEMATA_FOREIGN_KEY_CHARACTER_SETS FOREIGN KEY (DEFAULT_CHARACTER_SET_CATALOG, DEFAULT_CHARACTER_SET_SCHEMA, DEFAULT_CHARACTER_SET_NAME) REFERENCES CHARACTER_SETS	
39	DEU-P11-		4-Minor Editorial	P11-6.48 "TABLE_CONSTRAINTS" Table	The constraint TABLE_CONSTRAINTS_UNIQUE_CHECK is not needed, as the uniqueness of the constraint name is already checked by the assertion UNIQUE_CONSTRAINT_NAME in subclause 6.4	

SEQ #	Cmn ID	See Also	Severity	Reference	Description	Addressed By
	220				<p>Solution</p> <p>A possible solution is to remove the superfluous constraint.</p>	
40	DEU-P11-230		4-Minor Editorial	P11-6.50 "TABLE_PRIVILEGES" Table	<p>In the constraint TABLE_PRIVILEGES_TYPE_CHECK is the last element of the inlist misspelled. It should be REFERENCES instead of EFERENCES</p> <p>Solution</p> <p>Fix the typo.</p>	
41	DEU-P11-240		4-Minor Editorial	P11-6.54 "TRIGGERED_UPDATE_COLUMNS" Table	<p>The constraint TRIGGERED_UPDATE_COLUMNS_FOREIGN_KEY_TRIGGERS is not needed, as a more restrictive relationship is already guaranteed by constraint TRIGGERED_UPDATE_COLUMNS_EVENT_MANIPULATION_CHECK.</p> <p>Solution</p> <p>A possible solution is to remove the superfluous constraint.</p>	
42	DEU-P11-250		2-Minor Technical	P11-6.55 "TRIGGER_COLUMN_USAGE" Table	<p>The table TRIGGER_COLUMN_USAGE should have a foreign Key to the table TRIGGER_TABLE_USAGE, and not to TRIGGERS.</p> <p>Solution</p> <p>Add the following constraint: TRIGGER_COLUMN_USAGE_FOREIGN_KEY_TRIGGER_TABLE_USAGE FOREIGN KEY (TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME) REFERENCES TRIGGER_TABLE_USAGE It might be possible to remove the constraint TRIGGER_COLUMN_USAGE_FOREIGN_KEY_TRIGGERS.</p>	
43	DEU-P11-260		2-Minor Technical	P11-6.62 "USER_DEFINED_TYPES" Table	<p>In the last query of the constraint USER_DEFINED_TYPES_CHECK_SOURCE_TYPE is the column OBJECT_TYPE not in the reference List of the IN clause.</p> <p>Solution</p> <p>None provided with comment.</p>	
44	DEU-P11-270		2-Minor Technical	P11-Appendix C 6)	<p>The columns FEATURE_ID and FEATURE_NAME of the view SQL_PACKAGES are in the last Edition of the Standard deprecated and should now be deleted. But without these columns does the view not provide any usefull information. Should the entire View be deleted?</p> <p>Solution</p> <p>None provided with comment.</p>	
ISO/IEC FCD 9075-13:200x(E) SQL/JRT						
45	DEU-P13-010		1-Major Technical	P13-No specific location	<p>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.</p> <p>Solution</p> <p>None provided with comment.</p>	

END OF PAPER

3 March, 2005



ISO

International Organization for Standardization

ISO/IEC JTC 1/SC 32
Data Management and Interchange
WG 3
Database Languages

Title: Japan Ballot Comments on CD 9075:2007(E)

Status: Document to accompany ballot response

Author: Masashi Tsuchida ,Takaaki Shiratori, Takashi Kotera

Abstract: We present the comments of Japan on the CD ballot documents, to accompany our ballot response on that document.

References: [1] WG3:TXL-002 = 32N1198, ISO/IEC CD 9075-1, Information technology . Database languages SQL . Part 1: Framework (SQL/Framework)]
[2] WG3:TXL-003 = 32N1199, ISO/IEC CD 9075-2, Information technology . Database languages . SQL . Part 2: Foundation (SQL/Foundation)
[3] WG3:TXL-004 = 32N1201, ISO/IEC CD 9075-3, Information technology . Database languages . SQL . Part 3: Call-Level Interface (SQL/CLI)
[4] WG3:TXL-005 = 32N1202, ISO/IEC CD 9075-4, Information technology . Database languages . SQL . Part 4: Persistent Stored Modules (SQL/PSM)
[5] WG3:TXL-006 = 32N1203, ISO/IEC CD 9075-9, Information technology . Database languages . SQL . Part 9: Management of

ISO/IEC JTC1/SC32/WG3 TXL-*nnn*

External Data (SQL/MED)

[6] WG3:TXL-007 = 32N1204, ISO/IEC CD 9075-10, Information technology . Database languages . SQL . Part 10: Object language bindings (SQL/OLB)

[7] WG3:TXL-008 = 32N1205, ISO/IEC CD 9075-11, Information technology . Database languages . SQL . Part 11: Information and Definition Schemas (SQL/Schemata)

[8] WG3:TXL-009 = 32N1206, ISO/IEC CD 9075-11, Information technology . Database languages . SQL . Part 13: Information and Definition Schemas (SQL/JRT)

ISO/IEC JTC1/SC32/WG3 TXL-*nnn*

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
SQL/Framework						
001	JPN-P01-001		1-Major Technical	<i>P01-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	
SQL/Foundation						
002	JPN-P02-002		1-Major Technical	<i>P02-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	
003	JPN-P02-003		1-Major Technical	<i>P02-11.3, <table definition></i>	It is allowed that <table contents source> which is <as subquery clause> with WITH DATA is specified for a temporary table. But a temporary table can not be materialized at table definition. Solution None provided with comment.	
004	JPN-P02-004		1-Major Technical	<i>P02-14.8, <insert statement></i>	An INSERT statement has no different effects on identity columns specified GENERATED ALWAYS and that specified GENERATED BY DEFAULT. Solution None provided with comment.	
SQL/CLI						
005	JPN-P03-001		1-Major Technical	<i>P03-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	

ISO/IEC JTC1/SC32/WG3 TXL-nnn

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
SQL/PSM						
006	JPN-P04-001		1-Major Technical	<i>P04-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	
SQL/MED						
007	JPN-P05-001		1-Major Technical	<i>P05-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	
SQL/OLB						
008	JPN-P10-001		1-Major Technical	<i>P10-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	
SQL/Schema						
009	JPN-P11-001		1-Major Technical	<i>P11-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	
SQL/JRT						
009	JPN-P13-001		1-Major Technical	<i>P13-No specific location</i>	There are quite a few features to discuss a emerging next standard. Japan thinks that we should take enough time to add new features. Solution None provided with comment.	

ISO/IEC JTC1/SC32/WG3 TXL-*nnn*

ISO
International Organization for Standardization
ISO/IEC JTC 1/SC 32
Data Management and Interchange
WG 3
Database Languages

Project: ISO: 1.32.3.5

Title: Ballot Comment on ISO/IEC CD 9075-1, -2, -3, -4, -9, -10, -11, and -13

Status: Netherlands National Body Comments

Author: Stephen Cannan (Editor)

References:

- [1] SC32 N01198, CD 9075-1 Information Technology - Database Language SQL - Part 1: Framework (SQL/Framework) Jim Melton (Editor), December, 2004.
- [2] SC32 N01199, CD 9075-2 Information Technology - Database Language SQL - Part 2: Foundation (SQL/Foundation) Jim Melton (Editor), December, 2004.
- [3] SC32 N01201, CD 9075-3 Information Technology - Database Language SQL - Part 3: Call Level Interface (SQL/CLI) Jim Melton (Editor), December, 2004.
- [4] SC32 N01202, CD 9075-4 Information Technology - Database Language SQL - Part 4: Persistent Stored Modules (SQL/PSM) Jim Melton (Editor), December, 2004.
- [5] SC32 N01203, CD 9075-9 Information Technology - Database Language SQL - Part 9: Management of External Data (SQL/MED) Jim Melton (Editor), December, 2004.
- [6] SC32 N01204, CD 9075-10 Information Technology - Database Language SQL - Part 10: Object Language Bindings (SQL/OLB) Jim Melton (Editor), December, 2004.
- [7] SC32 N01205, CD 9075-11 Information Technology - Database Language SQL - Part 11: Schemata (SQL/Schemata) Jim Melton (Editor), December, 2004.
- [8] SC32 N01206, CD 9075-13 Information Technology - Database Language SQL - Part 13: Java Routines and Types (SQL/JRT) Jim Melton (Editor), December, 2004.

The Netherlands vote is:

SQL/Framework	No with comments
SQL/Foundation	No with comments
SQL/CLI	No with comments
SQL/PSM	No with comments
SQL/MED	No with comments
SQL/OLB	No with comments
SQL/Schemata	No with comments
SQL/JRT	Yes with comments

If all problems and technical errors, i.e. those identified in this ballot, and those identified during the editing meeting(s), are resolved to our satisfaction, then the Netherlands will change its NO votes to YES votes.

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
SQL/Framework						
	NLD-P01-001		3-Major Editorial	<i>P01-04.04, SQL data types</i>	FRM-002 The following Language Opportunity has been noted: Source: DBL:BBN-167/X3H2-98-386 Language Opportunity: Section needs a better organization There should be a section called SQL Data Types. Then a short definition of what is meant by an SQL data type. Then the list of the five types of data types (predefined, row type, user-defined type, collection type, and reference type). Then there should be a definition for each. Solution None provided with comment.	
	NLD-P01-002		2-Minor Technical	<i>P01-06.03.03.03, Rule evaluation order</i>	WG3-P01-001 The referenced subclause includes the following text: In general, if some syntactic element contains more than one other syntactic element, then the General Rules for contained elements that appear earlier in the production for the containing syntactic element are applied before the General Rules for contained elements that appear later. For example, in the production: <A> ::= <C> the Syntax Rules and Access Rules for <A>, , and <C> are effectively applied simultaneously. The General Rules for are applied before the General Rules for <C>, and the General Rules for <A> are applied after the General Rules for both and <C>. In SQL/Foundation, Subclause 13.5, "<SQL procedure statement>", is a clear exception to this general rule for General Rules, for the GRs of the particular contained statement (e.g., an <insert statement>) are clearly intended to be invoked only when a GR in Subclause 13.5 explicitly states that the contained statement to be executed. Now, it might be that the introductory words, "In general", can be taken to imply that there are some exceptions, but in that case shouldn't the exceptions be explicitly mentioned? Solution None provided with comment.	
SQL/Foundation						
	NLD-P02-001		1-Major Technical	<i>P02-04.32.01, General description of cursors</i>	FND-975 The following Possible Problem has been noted: Source: WG3:SIA-030 = H2-2004-??? Possible Problem: Subclause 4.32.1, "General description of cursors", contains: For every <declare cursor> in [emphasis added] an SQL-client module, a cursor	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>is effectively created when an SQL-transaction (see Subclause 4.35, “SQL-transactions”) referencing the SQL-client module is initiated.</p> <p>For every <dynamic declare cursor> in an <SQL-client module definition>, a cursor is effectively created when an SQL-transaction (see Subclause 4.35, “SQL-transactions”) referencing the <SQLclient module definition> is initiated. An extended dynamic cursor is also [emphasis added] effectively created when an <allocate cursor statement> is executed within an SQL-session and destroyed when that SQL-session is terminated.</p> <p>This text suffers from several problems, all of which probably need to be addressed at the same time:</p> <ol style="list-style-type: none"> 1) The first paragraph entertains the notion of a piece of SQL syntax appearing inside something that is not a piece of SQL syntax. It seems that either "<declare cursor>" should be replaced by "cursor", or "SQL-client module" should be replaced by "<SQL-client module definition>". In either case there would be knock-on effects on the remaining text. Note that the second paragraph prefers to talk about syntactic containment exclusively, but its text is too suspect for it to be used as a guideline for correcting the first paragraph. 2) The first paragraph entertains the notion of an SQL-transaction referencing an SQL-client module. Regardless of whether this should be SQL-client module or <SQL-client module definition>, it is not clear exactly what it means for an SQL-transaction that is the process of being initiated to “reference” that thing. Text elsewhere in SQL/Foundation (for example, in Subclause 16.7, “<commit statement>”), strongly suggests that several distinct SQL-client modules can be “associated with” the same current SQL-transaction. Can they be associated with the SQL-transaction without also being referenced by it? For that matter, can they be referenced by it without also being associated with it? If “referenced by” and “associated with” are synonymous, then how can all the SQL-client modules referenced by SQL-transaction <i>T</i> be known when <i>T</i> is initiated? 3) The second paragraph entertains the notion of creation of a cursor, and yet we have not been able to find any mention of this concept in any General Rule. Subclause 19.8, “<deallocate prepared statement>”, GR3) does require destruction of certain cursors, and this is corroborated (redundantly?) by Subclause 19.15, “<allocate cursor statement>”, GR3)d). However, neither Subclause 19.6, “<prepare statement>”, nor Subclause 19.15, “<allocate cursor statement>”, has any GR requiring a cursor to be created. 4) As already noted, a cursor, having been created in somewhat mysterious circumstances, is never destroyed (unless it happens to be an allocated cursor, or a cursor declared in a <compound statement>, see SQL/PSM, Subclause 13.1, “<compound statement>”, GR3)c)ii)2) and GR5)). It seems, then, that if <i>n</i> SQL-transactions in the same SQL-session “reference” the same SQL-client module, 	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>then each <declare cursor> contained in the corresponding <SQL-client module definition> causes the creation of <i>n</i> distinct cursors. And yet Subclause 14.2, “<open statement>”, SR1), says “Let <i>CR</i> be the cursor specified by <i>DC</i>”, where <i>DC</i> is a <declare cursor>. There are two problems with this:</p> <ul style="list-style-type: none"> — It is not clear which of those <i>n</i> cursors is the one specified by <i>DC</i>. Of course, if the standard clearly specified that all but one of these had been destroyed by this time, then there would be no ambiguity. — The cited text in Subclause 4.32.1, “General description of cursors”, makes it clear that a cursor comes into existence at run-time and therefore, not being a schema object, should not be referred to in a syntax rule. Since the SQL-session context already includes cursor positions, perhaps it should also be defined to include cursors. <p>5) The final sentence contains the word “also”, which could be understood as suggesting that some way of creating an extended dynamic cursor has already been mentioned. Moreover, it ignores the possibility of such a cursor being destroyed as a consequence of its prepared statement being deallocated before SQL-session termination.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-002		2-Minor Technical	<i>P02-03.01, Definitions</i>	<p>FND-953 The following Language Opportunity has been noted: Language Opportunity: There has been a discussion about Unicode 4.0 on the ISODBL list. [Ake has] found out that Note 7 in SQL/Foundation will be affected, because it contains explicit code points. U+180E and U+205F have been added to the "Zs" class in Unicode 4.0. Note that U+200B currently is of class "Zs", although it should not be treated as white-space. The Unicode Technical Committee will probably change the class for U+200B (ZERO-WIDTH SPACE) to "Cf" in the near future.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-003		2-Minor Technical	<i>P02-04.10, Collection types</i>	<p>FND-845 The following Language Opportunity has been noted: Source: WG3:YYJ-016 (CAN-P02-001, USA-P02-005) Language Opportunity: The next edition of the SQL standard should standardize the syntax and semantics of one or more additional collection types.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-004		1-Major Technical	<i>P02-04.14.02, Types of tables</i>	<p>FND-944 The following Possible Problem has been noted: Source: WG3:HBA-042 = H2-2003-____ Possible Problem: This Subclause, with paragraph numbers added for expository purposes, says of created temporary tables:</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>1) A global temporary table is a named table defined by a <table definition> that specifies GLOBAL TEMPORARY. A created local temporary table is a named table defined by a <table definition> that specifies LOCAL TEMPORARY. Global and created local temporary tables are effectively materialized only when referenced in an SQL-session. Every SQL-client module in every SQL-session that references a created local temporary table causes a distinct instance of that created local temporary table to be materialized. That is, the contents of a global temporary table or a created local temporary table cannot be shared between SQL-sessions.</p> <p>2) In addition, the contents of a created local temporary table cannot be shared between SQL-client modules of a single SQL-session. The definition of a global temporary table or a created local temporary table appears in a schema. In SQL language, the name and the scope of the name of a global temporary table or a created local temporary table are indistinguishable from those of a persistent base table. However, because global temporary table contents are distinct within SQL-sessions, and created local temporary tables are distinct within SQL-client modules within SQL-sessions, the <i>effective</i> <schema name> of the schema in which the global temporary table or the created local temporary table is instantiated is an implementation-dependent <schema name> that may be thought of as having been effectively derived from the <schema name> of the schema in which the global temporary table or created local temporary table is defined and the implementation-dependent SQLsession identifier associated with the SQL-session.</p> <p>3) In addition, the <i>effective</i> <schema name> of the schema in which the created local temporary table is instantiated may be thought of as being further qualified by a unique implementation-dependent name associated with the SQL-client module in which the created local temporary table is referenced.</p> <p>4) A declared local temporary table is a module local temporary table. A module local temporary table is a named table defined by a <temporary table declaration> in an SQL-client module. A module local temporary table is effectively materialized the first time it is referenced in an SQL-session, and it persists for that SQL-session.</p> <p>Neither the first sentence of paragraph 1 nor the General Rules of Subclause 11.3, "<table definition>", make it clear that a <table definition> creates a persistent (temporary) table descriptor.</p> <p><i>Materialised</i> is not defined and the meaning added by the qualifier <i>effectively</i> is unclear; the use of <i>instantiated</i> in paragraph 3 suggests a distinction that is probably unintended.</p> <p>In paragraph 2, "<i>effective</i> <schema name> ... may be thought of as ..." [emphasis in original] doesn't tell us what the purpose of this thinking is, nor how the <i>effective</i> <schema name> differs from any possible <i>actual</i> one. It also misleads us into imagining that a local temporary table created in the same schema as a</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>global one might have the same <table name>, when on reflection it clearly can't. This is not, be it noted, analogous to extended names, as explicitly intended by DBL:LON-156.</p> <p>Furthermore, "unique implementation-dependent name associated with the SQL-client module in which the created local temporary table is referenced" is unclear. Is this name persistent? One interpretation is that every occurrence of the <table name> must be contained in the same <SQL-client module definition>.</p> <p>But perhaps it is intended to mean that there is no restriction on where the <table name> can occur; but only occurrences in externally-invoked procedures in the same module refer to the same thing (i.e. those of other modules refer to their own "local" temporary table).</p> <p>In paragraph 2, it is not clear whether a distinction is intended between "global temporary table contents" and "created local temporary tables", but presumably not.</p> <p>In paragraph 2, the meaning of "distinct within SQL-sessions" is unclear, because there is only one SQLsession active at any one time (even though there may be dormant ones).</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-005		1-Major Technical	P02-04.14.02, <i>Types of tables</i>	<p>FND-945 The following Possible Problem has been noted: Source: WG3:HBA-042 = H2-2003-____ Possible Problem: In this Subclause, as modified by [PSM-WD], Subclause 4.3.1, "Types of tables", with paragraph numbers added for expository purposes, says of declared temporary tables:</p> <ol style="list-style-type: none"> 1) A declared local temporary table is a module local temporary table. A module local temporary table is a named table defined by a <temporary table declaration> in an SQL-client module. A module local temporary table is effectively materialized the first time it is referenced in an SQL-session, and it persists for that SQL-session. 2) A declared local temporary table may be declared in an SQL-client module. 3) Inserted by SQL/PSM A declared local temporary table may be declared in an SQL-server module. 4) A declared local temporary table that is declared in an SQL-client module is a named table defined by a <temporary table declaration> that is effectively materialized the first time any <externallyinvoked procedure> in the <SQL-client module definition> that contains the <temporary table declaration> is executed. A declared local temporary table is accessible only by <externally-invoked procedure>s in the <SQL-client module definition> that contains the <temporary table declaration>. The effective <schema name> of the <schema qualified name> of the declared local temporary table may be thought of as the 	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>implementation-dependent SQL-session identifier associated with the SQL-session and a unique implementation-dependent name associated with the <SQL-client module definition> that contains the <temporary table declaration>. 5) Inserted by SQL/PSM A declared local temporary table that is declared in an SQL-server module is a named table defined by a <temporary table declaration> that is effectively materialized the first time any <module routine> in the <SQL-server module definition> that contains the <temporary table declaration> is executed. A declared local temporary table is accessible only by <module routine>s in the <SQL-server module definition> that contains the <temporary table declaration>. The effective <schema name> of the <schema qualified name> of the declared local temporary table may be thought of as the implementation-dependent SQL-session identifier associated with the SQL-session and the name of the <SQL-server module definition> that contains the <temporary table declaration>.</p> <p>The second sentence of paragraph 1 is no longer true when paragraphs 3 and 5 have been inserted by PSM. Moreover, whatever truth is expressed by paragraph 1 is repeated by paragraphs 2 and 4, which are specific to SQL-client modules. Evidently paragraphs 2 and 4 were inserted to correspond to paragraphs 3 and 5, so making paragraph 1 redundant, which should have been deleted at the same time.</p> <p>To say, in paragraphs 4 and 5, that a declared local temporary table has an <i>effective</i> <schema name> is misleading, since its name must be prefixed by MODULE.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-006		2-Minor Technical	P02-04.14.02, <i>Types of tables</i>	<p>FND-969 The following Language Opportunity has been noted: Source: WG3:SIA-018 = H2-2004-429 Language Opportunity: Every view component is an underlying table. The reason that underlying table terminology was not used was that the hierarchy of underlying tables does not follow the hierarchy of syntactic containment, owing to the distinctive treatment accorded the tables and derived tables in the FROM clause of a <query specification> compared with other derived tables found in a <query specification>. If this issue can be overcome, it may be possible to eliminate the notion of view component and just use underlying tables.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-007		1-Major Technical	P02-04.17, <i>Integrity constraints</i>	<p>FND-703 The following Possible Problem has been noted: Source: WG3:BBN-139/X3H2-98-363 Possible Problem: It seems that SQL3's specification of deferrable constraints is ill-specified. Referential constraints are based on the notion of marking rows for deletion</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>before the rows are effectively deleted at the end of the SQL statement. This is necessary because updates cascaded by referential constraints need to be “propagated” through rows marked for deletion in order to avoid anomalies (non deterministic behavior). If a referential constraint is deferred, then rows that need to be kept around for the execution of referential constraints will not be present at the end of the transaction (or when the referential constraint is turned to immediate). These rows will be deleted at the end of the SQL statements. So, it is unclear how referential constraints are checked in these cases (e.g., are we supposed to maintain multiple versions of the database and check the constraints against those versions? If so, how do the updates are “propagated” to the current version of the database?).</p> <p>Another problem with deferrable constraints is that stored procedures and triggers can never rely on the existence of a consistent database during their execution because the application that caused the invocation of the stored procedure and/or trigger could have deferred the checking of certain constraints prior to the invocation of the procedure or trigger. (Please note that this has also a major impact to the implementation of such concepts because plans generated by optimizers (e.g., the exploitation of a unique index) can be invalidated by deferring such constraints.)</p> <p>Also it is not clear to me that deferrable constraints and triggers work smoothly. First, BEFORE triggers execute BEFORE the SQL statement that activates them. However, the BEFORE execution cannot be guaranteed if referential constraints are deferred because the execution of the BEFORE trigger needs to be deferred as well. Second, if the BEFORE trigger is modifying the values of transition variables such that they can be inserted/updated with correct values in the database, what will happen with such values if the BEFORE trigger executes after the database has been updated? Third, triggers are executed in a well defined order. This is important to guarantee that changes to the database are done in a deterministic manner. If constraints are deferred, then one may end up deferring the execution of several instances of the same trigger for which there is no well defined order of execution. This will lead to non-deterministic behavior in the database.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-008		2-Minor Technical	<i>P02-04.27, SQL-invoked routines</i>	<p>FND-725 The following Language Opportunity has been noted: Source: WG3:FRA-122/X3H2-98-688) Language Opportunity: Subclause 4.27, “SQL-invoked routines”, does not adequately describe the concepts of dynamic binding and subject function selection.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-009		2-Minor	<i>P02-04.32,</i>	<p>FND-607 The following Language Opportunity has been noted:</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
			Technical	<i>Cursors</i>	<p>Source: DBL:LGW-146/X3H2-97-349</p> <p>Language Opportunity: The ability to hold a cursor through rollback will be extremely useful to applications. Yet the second bullet of this Subclause says "a holdable-cursor is closed no matter what its state if the SQL-transaction is terminated with a rollback operation." This provision is not always necessary according to Jim Gray and Andrew Reuter "Transaction Processing: Concepts and Techniques".</p> <p>Solution None provided with comment.</p>	
	NLD-P02-010		2-Minor Technical	<i>P02-04.32.01, General description of cursors</i>	<p>FND-929 The following Possible Problem has been noted: Source: WG3:HBA-040</p> <p>Possible Problem: Although the second paragraph of this subclause defines terms to denote both varieties of dynamic cursors, it does not provide a way of referring to a cursor that is <i>not</i> dynamic.</p> <p>Solution None provided with comment.</p>	
	NLD-P02-011		1-Major Technical	<i>P02-04.33.04, SQL-statements and transaction states</i>	<p>FND-923 The following Possible Problem has been noted: Source: WG3:HBA-029</p> <p>Possible Problem: Subclause 4.33.4, "SQL-statements and transaction states", includes: If the initiation of an SQL-transaction occurs in an atomic execution context, and an SQL-transaction has already completed in this context, then an exception condition is raised: <i>invalid transaction termination</i>. At first sight it doesn't seem possible for transaction termination to be followed by transaction initiation "in" the same atomic execution context. In general, transaction initiation is caused by execution of an SQL-statement of the transaction-initiating kind and transaction termination is caused by executing an SQL-statement of a different kind (COMMIT or ROLLBACK). Note that Subclause 13.5, "<SQL procedure statement>", GR2, specifies that a new statement execution context is created whenever an <SQL procedure statement> is executed. Note also that in Subclause 4.33.5, "SQL-statement atomicity and statement execution contexts", we are told (last paragraph) that an SQL-transaction cannot be explicitly terminated within an atomic execution context. We conclude that the cited paragraph is relevant only when execution of a transaction-initiating statement (a) actually causes a transaction to be initiated, and (b) causes an exception to be raised of the special transaction rollback kind (this being the only way of implicitly terminating a transaction). In the light of this observation, we perceive the following problems: 1) There is no General Rule in, for example, Subclause 13.5, "<SQL procedure statement>", to confirm the cited text.</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>2) If execution of an SQL-statement causes an exception to be raised, then all changes to SQL-data and schemas are cancelled anyway. As the failing statement is also the one that initiated the transaction, the effect seems to be the same as that of a successful rollback, so what's the point in raising an additional exception expressing the fact that the transaction cannot be terminated? After all, the user executing the statement in question wasn't even trying to terminate the current transaction!</p> <p>We wonder if the rule was intended to cater for some eventuality other than the only one we can find.</p> <p>Even if the foregoing analysis proves to be refutable, it might be a good idea to add an explanation to Subclause 4.33.4, "SQL-statements and transaction states".</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-012		1-Major Technical	<i>P02-04.33.05, SQL-statement atomicity and statement execution contexts</i>	<p>FND-924 The following Possible Problem has been noted: Source: WG3:HBA-029 Possible Problem: Subclause 4.33.5, "SQL-statement atomicity and statement execution contexts", includes: The statement execution context brought into existence by the execution of an atomic SQL-statement or the evaluation of a <subquery> is an atomic execution context. The inclusion of "or the evaluation of a <subquery>", and the GRs of Subclause 7.15, "<subquery>", that back it up, seem questionable. Isn't expression evaluation always atomic? The question also arises as to whether deletion of the questionable text (and GRs) would make any material difference to the standard. A search of the SQL:2003 Foundation FDIS for the word "atomic" reveals no GRs that are conditional upon the atomicity or non-atomicity of a statement execution context. Instead, there are some special GRs for <subquery> that enforce its atomicity by creating and destroying a savepoint level, and in Subclause 13.5, "<SQL procedure statement>", for undoing any changes to SQLdata or schemas made execution of by an atomic statement that terminates with an exception. It seems, then, that the only effects caused by atomicity are to do with savepoints and database updates. But it appears that database updates are not possible during evaluation of a subquery, being outlawed by Subclause 7.13, "<query expression>", SR23): 1) 23) A <query expression> QE shall not generally contain a <routine invocation> whose subject routine is an SQL-invoked routine that possible modifies SQL-data. Note that the BNF production for <subquery> is <left paren> <query</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>expression> <right paren> and a <query expression> cannot contain an SQL procedure statement. SR23) in combination with the GRs of Subclause 10.4, "<routine invocation>", makes it impossible for an evaluation of an <routine invocation> caused by evaluation of a <subquery> to cause an SQL-data change statement to be executed. Therefore it is impossible for evaluation of a <subquery> to have any effect on SQL-data or schemas (possibly explaining the lack of a GR in Subclause 7.15, "<subquery>", specifying that changes to SQL-data and schemas are to be cancelled). Therefore any savepoints established during evaluation of a <subquery> have to be ineffectual. Therefore there is no point in establishing a new, atomic, statement execution context for the evaluation of a <subquery>.</p> <p>But that's not all! Consider the <query expression> SELECT foo() FROM T, and suppose that there is some flaw in the reasoning that leads to the conclusion that the effect of the invocation of foo() cannot possibly depend on whether the current statement execution context is atomic. In that case SELECT foo() FROM T would not in general be equivalent to SELECT * FROM (SELECT foo() FROM T) T, for the shorter expression does not contain a <subquery>, whereas the longer one does. In general, the consequences of the effect of evaluation of a <query expression> possibly varying according to whether it is enclosed in parentheses are very unclear and would surely raise very awkward problems for optimisers.</p> <p>The foregoing analysis also brings into question the following sentence in Subclause 4.33.4, "SQLstatements and transaction states":</p> <p>1) If an <SQL-control statement> causes the evaluation of a <subquery> and there is no current SQLtransaction, then an SQL-transaction is initiated before evaluation of the <subquery>.</p> <p>Perhaps this is pointless, too. If it proves not to be, the wisdom of starting a transaction in the middle of executing an SQL-statement - - at an indeterminate point in that execution, to boot - - is surely questionable.</p> <p>In any case, we note that the sentence is not borne out by the GRs of Subclause 7.15, "<subquery>".</p> <p style="text-align: center;">Solution</p> <p>Delete "or evaluation of a <subquery>" from the cited sentence of Subclause 4.33.5, "SQL-statement atomicity and statement execution contexts"; possibly delete the cited sentence of Subclause 4.33.4, "SQL-statements and transaction states"; delete GRs 1) ("Let OLDSEC ..."and 4) ("All savepoints ...") of Subclause 7.15, "<subquery>". A search of the SQL:2003 Foundation FDIS for "<subquery>" reveals that a change might also be needed in Subclause 4.33.3, "SQL-statements and SQL-data access indication".</p> <p>Other Parts of SQL:2003 have not been checked to see if they might be affected. But see WG3:HBA-041.</p>	
	NLD-P02-013		1-Major	P02-04.35.02,	FND-972 The following Possible Problem has been noted:	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
			Technical	<i>Savepoints</i>	<p>Source: WG3:SIA-031 = H2-2004-???</p> <p>Possible Problem: Neither here nor anywhere else is there any definition of the term savepoint. Paragraph 5 of this Subclause contains: If a <rollback statement> references a savepoint SS, then all changes made to SQL-data or schema subsequent to the establishment of the savepoint are canceled, all savepoints established since SS was established are destroyed, and the SQL-transaction is restored to its state as it was immediately following the execution of the <savepoint statement>. The state of an SQL-transaction is not defined, nor is it referred to in Subclause 16.8, "<rollback statement>". Presumably what is being referred to is some (or all) of the SQL-session context. Which <savepoint statement> is not specified. Presumably the one that established SS. So perhaps "... following the establishment of SS"; or even "... as it was at that time". Perhaps what is really meant is something to the effect of, A savepoint is a preserved copy of (the values of specified elements) of the SQL-session context at the time a <savepoint statement> was executed, plus sufficient data to enable all subsequent changes to SQL-data or schemas in the current SQL-transaction to be canceled. When a <rollback statement> is executed, that contains a <savepoint specifier> SS, then all changes made to SQL-data or schema subsequent to the establishment of SS are canceled, all savepoints established since SS was established are destroyed, and elements of the SQL-session context are restored to the values that were preserved in SS. Paragraph 6 says: It is implementation-defined whether or not, or how, a <rollback statement> that references a <savepoint specifier> affects diagnostics area contents, the contents of SQL descriptor areas, and the status of prepared statements. This implementation-defined element is not mentioned in Annex B, "Implementation-defined elements" (or in Annex C, "Implementation-dependent elements"). Solution: Specify what happens in terms of the contents of the SQL-session context.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-014		1-Major Technical	<i>P02-04.37, SQL-sessions</i>	<p>FND-954 The following Possible Problem has been noted: Source: WG3:ZSH-037R1/H2-2003-???</p> <p>Possible Problem: WG3:FRA-045r4 proposed no changes to what is now WG3:ZSH-013, Subclause 4.37, "SQLsession". However, according to WG3:FRA-045r4, Section 2.1, "Authorization stack":</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>There is a stack of SQL-session contexts. There is one cell on this stack when the SQLsession begins.</p> <p>An additional SQL-session context is pushed on the stack for each <routine invocation>, and is removed when the <routine invocation> completes execution.</p> <p>There is no reference to this anywhere in this subclause, although there are various statements of the form "An SQL-session has a ...".</p> <p>Moreover, the list of SQL-session contents is incorrect and incomplete. The term "current SQL-session identifier" is listed, where the meaning of "current" is indicated in the following NOTE (55 in WG3:ZSH-013) and evidently used to distinguish the "current" SQL-session from dormant SQL-sessions. It is therefore probably intended to refer to the SQL-session identifier of the currently active (as opposed to dormant) SQL-session. If this surmise is correct, then the "current SQL-session user identifier" is missing.</p> <p>There is no reference to the authorization stack, though the two terms used to refer to the components of the only visible cell of that stack are mentioned.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-015		1-Major Technical	P02-04.37.04, Execution contexts	<p>FND-955 The following Possible Problem has been noted: Source: WG3:ZSH-037R1/H2-2003-???</p> <p>Possible Problem:</p> <p>This subclause contains the statement:</p> <p>There is always a statement execution context, a routine execution context, and zero or more trigger execution contexts.</p> <p>There is a significant and unnecessary inconsistency between the descriptions of routine execution contexts and trigger execution contexts.</p> <p>Consider what happens if an SQL-invoked routine R1 invokes another, R2. Are there now one or more than one routine execution contexts? The answer is clearly there is one in each of two levels of the stack of SQL-session contexts, as is made clear by Subclause 10.4, "<routine invocation>". Whether there is a routine execution context when no routine has been invoked is debatable: it could be (and indeed is) said that there is an empty one; or it could be said that there is none. In which case, it would be true to say that "there are zero or more routine execution contexts", as is said for trigger execution contexts.</p> <p>Consider now how it arises that there is more than one trigger execution context. The only case that springs to mind is that of the triggered action of a trigger T1, causing another trigger T2 to fire. In this case, each trigger will have a trigger execution context. However, it seems fairly clear that the triggered action of T2 cannot access the state changes in the trigger execution context of T1. Therefore, to say that there are, during the execution of T2, two trigger execution contexts, although true in a sense, is unhelpful.</p> <p>Moreover, we seem to be saying that these two trigger execution contexts are in</p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>the same SQLsession context; unless, of course, T1 invokes a routine that causes T2 to fire, in which case a new SQL-session context is created, containing a new routine execution context. However, whether or not it contains, when created, the trigger execution context of T1, we are unable to discover.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-016		1-Major Technical	<i>P02-05.04, Names and identifiers</i>	<p>FND-932 The following Possible Problem has been noted: Source: WG3:HBA-050R1 Possible Problem: SR19) of this subclause is: 19) An <identifier> that is a <correlation name> is associated with a table within a particular scope. The scope of a <correlation name> is either a <select statement: single row>, <subquery>, or <query specification> (see Subclause 7.6, “<table reference>”), or is a <trigger definition> (see Subclause 11.39, “<trigger definition>”). Scopes may be nested. In different scopes, the same <correlation name> may be associated with different tables or with the same table. The inclusion of <subquery> is puzzling. For consider that if such a scope is contained in a <subquery>, then it must also be wholly contained in some <query specification> contained in that <subquery>. Furthermore, a <subquery> that contains more than one <query specification> cannot possibly constitute the scope of any <correlation name>. For example: (SELECT * FROM T1 UNION SELECT * FROM T2) The scope of any correlation name defined with such a <subquery> would be confined to the particular <query specification> in which it is defined. A scalar expression could be added to the <subquery> that includes an outer reference, but the <correlation name> used in that reference would have a wider scope than the <subquery>. The inclusion of <query specification> is also suspect, because the scope of a <correlation name> is not necessarily a whole <query specification>.</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-017		1-Major Technical	<i>P02-05.04, Names and identifiers</i>	<p>FND-946 The following Possible Problem has been noted: Source: WG3:HBA-042 = H2-2003-____ Possible Problem: This subclause says (regarding local temporary tables): Something needs to be said, in either or both of the Syntax Rules and General Rules of Subclause 5.4, “Names and identifiers”, about how a <table name> identifies a created temporary table. As a minimum, a reference to a created local temporary table must be prohibited in any <schema routine> R, because, by the time R is invoked, it cannot be regarded as being syntactically contained in the <SQLclient module definition></p>	

SEQ #	Cmnt ID	See Also	Severity	Reference	Description	Addressed By
					<p>that contained the <externally-invoked procedure> which created it.</p> <p style="text-align: center;">Solution</p> <p>As a minimum, a Syntax Rule should be added, to the effect that: 1) If <table name> identifies a created local temporary table, then <table name> shall not be contained in a <schema routine>. Note: This appears to remove the need for any reference to created local temporary tables in Subclause 10.4, "<routine invocation>", General Rule 5) d) i), which is a problem for PSM.</p>	
	NLD-P02-018		2-Minor Technical	P02-06.01, <data type>	<p>FND-729 The following Language Opportunity has been noted: Source: WG3:YGG-112 (SQL/MM YGG-023), Paul Cotton for WG4, July 6, 1999, and Paul Scarponcini via email on 6 July 1999 Language Opportunity: According to YGG-112: "REF types need to be scoped; i.e., the table(s) they refer to must be explicitly provided. If a column is of type REF type, the scope may be defined at table creation time. If the column is of type UDT which contains REF type attributes, then the scope must be declared when the UDT is created. The SQL/MM Part 3: Spatial standard defines the UDTs for spatial data. The standard is unable to predict in which tables the referenced information will be stored; this is a function of database design. Therefore, column scoping must be expanded to support deeply nested references, i.e., REF types within a UDT or ARRAY. This would allow a user, when creating tables, to define the scope of a UDTs REF type as part of the column definition for a column of type UDT." When a <reference type> is used as the data type of an attribute of a structured type, the <scope clause> must be specified when the encompassing user-defined type is defined. It is a Language Opportunity to be able to specify the <scope clause> of the "nested" <reference type>s when a column is defined on the encompassing user-defined type. Paul Scarponcini added: This applies to ARRAYS as well (e.g., an ARRAY of REF, and ARRAY of UDTs having REF attributes. The resultant syntax may be quite messy, as different REFs within the column may have different scopes. Would it be worth considering reversing the scope specification: when the reference dtable is created, specify that it shall be included in the scope for a particular column, rather than specifying the referenced table when the referencing column is specified?</p> <p style="text-align: center;">Solution</p> <p>None provided with comment.</p>	
	NLD-P02-019		2-Minor Technical	P02-06.01, <data type>	<p>FND-730 The following Language Opportunity has been noted: Source: WG3:YGG-112 (SQL/MM YGG-023) and Paul Cotton for WG4, July 6, 1999</p>	

