



INTERNATIONAL STANDARD ISO/IEC 9075-4:1999
TECHNICAL CORRIGENDUM 2

Published 2003-06-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Database languages — SQL —

Part 4:

Persistent Stored Modules (SQL/PSM)

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Langages de base de données — SQL —
Partie 4: Modules stockés persistants (SQL/PSM)

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to ISO/IEC 9075-4:1999 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*. ISO/IEC 9075-4:1999/Cor. 2:2003 cancels and replaces ISO/IEC 9075-4:1999/Cor. 1:2000.

Statement of purpose for rationale:

A statement indicating the rationale for each change to ISO/IEC 9075 is included. This is to inform the users of that standard as to the reason why it was judged necessary to change the original wording. In many cases the reason is editorial or to clarify the wording; in some cases it is to correct an error or an omission in the original wording.

Notes on numbering:

Where this Corrigendum introduces new Syntax, Access, General and Conformance Rules, the new rules have been numbered as follows:

Rules inserted between, for example, Rules 7) and 8) are numbered 7.1), 7.2), etc. [or 7) a.1), 7) a.2), etc.]. Those inserted before Rule 1) are numbered 0.1), 0.2), etc.

Where this Corrigendum introduces new Subclauses, the new subclauses have been numbered as follows:

Subclauses inserted between, for example, Subclause 4.3.2 and 4.3.3 are numbered 4.3.2a, 4.3.2b, etc.

Those inserted before, for example, 4.3.1 are numbered 4.3.0, 4.3.0a, etc.

Contents

	Page
3.3.1.1 Exceptions	3
3.3.1.2 Other terms	3
3.3.2.1 Clause, Subclause, and Table relationships	3
4.2 SQL-invoked routines	4
4.7 Diagnostics areas	5
4.8 Cursors	5
4.9 Condition handling	5
4.10 SQL-statements	6
4.10.1 SQL-statements classified by function	7
4.10.7 SQL-statement atomicity	7
4.11 SQL-sessions	8
5.1 <token> and <separator>	8
6.2 <identifier chain>	8
8.1 <routine invocation>	9
9.12 <drop collation statement>	9
9.13 <drop transliteration statement>	10
9.18 <SQL-server module definition>	10
9.19 <drop module statement>	10
10.2 <revoke statement>	10
11.1 Calls to an <externally-invoked procedure>	11
11.2 <SQL procedure statement>	11
11.1 Calls to an externally-invoked procedure	12
12.2 <fetch statement>	12
13.1 <compound statement>	13
13.2 <handler declaration>	14
13.5 <assignment statement>	16
13.7 <if statement>	16
13.13 <for statement>	17
15.1 <embedded SQL host program>	18
16.1 <get diagnostics statement>	18
16.2 <signal statement>	19
16.3 <resignal statement>	20
17.2 MODULE_PRIVILEGES view	21
17.4 MODULES view	21
19.1 SQLSTATE	22
Annex A SQL Conformance Summary	23
Annex E Incompatibilities with ISO/IEC 9075:1992	23
Annex F SQL Feature Taxonomy	23

Information technology — Database languages — SQL —

Part 4:

Persistent Stored Modules (SQL/PSM)

TECHNICAL CORRIGENDUM 2

3.3.1.1 Exceptions

1. *Rationale: Delete definition of an unused term.*

Delete Subclause 3.3.1.1.

3.3.1.2 Other terms

1. *Rationale: Delete what is moved to Part 2.*

Replace the 1st, 2nd and 3rd paragraphs with:

Insert this paragraph An SQL-statement *S1* may be said to be executed as a direct result of executing an <SQL control statement> *S2* if *S2* contains *S1*.

3.3.2.1 Clause, Subclause, and Table relationships

1. *Rationale: Correct the classification of SQL-statements.*

Insert the following row into Table 1, "Clause, Subclause, and Table relationships":

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 4.10.2a "Preparable and immediately executable SQL-statements"	Subclause 4.6.5 Preparable and immediately executable SQL-statements	ISO/IEC 9075-5

2. *Rationale: Correct cross reference.*

Replace the following rows in Table 1, "Clause, Subclause, and Table relationships":

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 4.10.7 "SQL-statement atomicity"	(none)	(none)

with:

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
--	---	--------------------------------

Subclause 4.10.7 “SQL-statement atomicity”	Subclause 4.30.4 “SQL-statement atomicity”	ISO/IEC 9075-2
--	--	----------------

3. *Rationale: Address requirement for multiple diagnostics areas*

Insert the following row into Table 1, "Clause, Subclause, and Table relationships":

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 4.11, “SQL-sessions”	Subclause 4.34, “SQL-sessions”	ISO/IEC 9075-2

4. *Rationale: Correct cross reference.*

Replace the following rows in Table 1, "Clause, Subclause, and Table relationships":

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 4.11.1, “Privileges”	Subclause 4.31, “Basic security model”	ISO/IEC 9075-2
Clause 11, “SQL-client module”	Subclause 13.1, “<SQL-client module definition>”	ISO/IEC 9075-2

with:

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 4.11.1, “Privileges”	Subclause 4.31.2, “Privileges”	ISO/IEC 9075-2
Clause 11, “SQL-client module”	Subclause 13, “SQL-client modules”	ISO/IEC 9075-2

4.2 SQL-invoked routines

1. *Rationale: Correct the Part Merge instructions*

Replace the 2nd paragraph with:

Replace 32nd paragraph — An SQL-invoked routine has a *routine SQL-path*, which is inherited from its containing SQL-server module or schema, the current SQL-session, or the containing SQL-client module.

Replace the 3rd paragraph with:

Insert in the 34th paragraph — If the SQL-invoked routine is not a schema-level routine, then the <SQL-server module name> of the SQL-server module that includes the SQL-invoked routine and the <schema name> of the schema that includes the SQL-server module.

4.7 Diagnostics areas

1. *Rationale: Address requirement for multiple diagnostics areas*

Replace the entire Subclause with:

Replace paragraph 8 The <get diagnostics statement> is used to obtain information from an occupied condition area, referenced by its ordinal position within the specified diagnostics area. Normally, only the first (i.e., current) area may be specified. However, if a handler is active, it is also possible to reference the second (i.e., most recently stacked) area, in order to obtain information about the condition that caused the handler to become active.

Insert this paragraph Information about a completion or exception condition that causes a handler to be activated is placed into one or more condition areas of the first diagnostics area before any handler is activated. The diagnostics area stack is then pushed so that the handler can access that information even while its own execution is causing the first diagnostics area to be modified.

Insert this paragraph The first diagnostics area is emptied during the execution of a <signal statement>. Information is added to the first diagnostics area during the execution of a <resignal statement>.

4.8 Cursors

1. *Rationale: Fix bug in processing results sets in PSM.*

Replace the 1st paragraph with:

Insert this paragraph For every <declare cursor> in a <compound statement>, a cursor is effectively created each time the <compound statement> is executed, and destroyed when that execution completes, unless the cursor is an open result set cursor.

4.9 Condition handling

1. *Rationale: Address requirement for multiple diagnostics areas*

Replace the 5th paragraph with:

A condition represents an error or informational state caused by execution of an <SQL procedure statement>. Conditions are raised to provide information in a diagnostics area about the execution of an <SQL procedure statement>.

2. *Rationale: Address requirement for multiple diagnostics areas. Simplify and clarify where execution resumes.*

Replace the 12th, 13th and 14th paragraphs, including NOTE 2, with:

If a handler type specifies CONTINUE, then, when the handler is activated, it will:

- Push the diagnostics area stack.
- Execute the handler action.

- Pop the diagnostics area stack.
- Cause the SQL-session to continue as it would have done if execution of the innermost executing statement that raised the condition had completed.

If a handler type specifies EXIT, then, when the handler is activated, it will:

- Push the diagnostics area stack.
- Execute the handler action.
- Pop the diagnostics area stack.
- Implicitly LEAVE the <compound statement> for which the handler was created, with no active exception condition.

If a handler type specifies UNDO, then, when the handler is activated, it will:

- Push the diagnostics area stack.
- Roll back all of the changes to SQL-data or to schemas by the execution of every SQL-statement contained in the SQL-statement list of the <compound statement> at the scope of the handler and cancel any <SQL procedure statement>s triggered by the execution of such statements.
- Execute the handler action.
- Pop the diagnostics area stack.
- Cause the SQL-session to continue as it would have done if execution of the <compound statement> for which the handler was created had completed.

4.10 SQL-statements

1. *Rationale: Correct the classification of SQL-statements.*

Insert the following Subclause after Subclause 4.10.2, “Embeddable SQL-statements”:

4.10.2a Preparable and immediately executable SQL-statements

Insert this paragraph

Consequently, the following SQL-control statements are not preparable:

- <compound statement>
- <case statement>
- <if statement>
- <iterate statement>
- <leave statement>

- <loop statement>
- <while statement>
- <repeat statement>
- <for statement>
- <assignment statement>

Insert this paragraph Consequently, the following SQL-control declarations are not preparable:

- <condition declaration>
- <handler declaration>
- <SQL variable declaration>

4.10.1 SQL-statements classified by function

1. *Rationale: Correct the classification of SQL-statements.*

Delete the 2nd bullet from the 2nd paragraph.

4.10.7 SQL-statement atomicity

1. *Rationale: Not all SQL-statements are atomic.*

Insert the following paragraph:

Insert this paragraph The following are non-atomic SQL-statements:

- <assignment statement>
- <compound statement>, unless BEGIN ATOMIC is specified
- <case statement>
- <if statement>
- <loop statement>
- <while statement>
- <repeat statement>
- <for statement>