

ISO/IEC JTC 1/SC 32 N 1977

Date: 2010-02-22

REPLACES: 32N1924

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	Final Text Submitted for COR Publication
TITLE	ISO/IEC 9075-4:2008 Cor 1:2010(E) Information technology - Database languages - SQL - Part 4: Persistent Stored Modules (SQL/PSM)
SOURCE	WG3 - Stephen Cannan - corrigendum editor
PROJECT NUMBER	1.32.03.06.04.00
STATUS	sent to ITTF for COR publication as approved at Jeju 2009-06-26; Disposition of Comments on DCOR 32N1924 may be found in 32N1982
REFERENCES	
ACTION ID.	ITTF
REQUESTED ACTION	
DUE DATE	--
Number of Pages	9
LANGUAGE USED	English
DISTRIBUTION	P & L Members SC Chair WG Conveners and Secretaries

Dr. Timothy Schoechle, Secretary, ISO/IEC JTC 1/SC 32
Farance Inc *, 3066 Sixth Street, Boulder, CO, United States of America
Telephone: +1 303-443-5490; E-mail: Timothy@Schoechle.org
available from the JTC 1/SC 32 WebSite <http://www.jtc1sc32.org/>
*Farance Inc. administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI

ISO/IEC JTC 1/SC 32

Date: 2010-02-14

ISO/IEC 9075-4:2008/Cor.1: 2010(E)

ISO/IEC JTC 1/SC 32/WG 3

The United States of America (ANSI)

Information technology — Database languages — SQL —

**Part 4:
Persistent Stored Modules (SQL/PSM)**

TECHNICAL CORRIGENDUM 1

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

RECTIFICATIE TECHNIQUE 1

Document type: Corridenga
Document subtype: Technical Corrigendum (COR)
Document stage: (5) IS Publication
Document language: English

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 Subclause 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

6	Scalar expressions	1
6.1	<value specification> and <target specification>	1
8	Additional common elements	1
8.1	<routine invocation>.....	1
9	Schema definition and manipulation	2
9.9	<drop table constraint definition>.....	2
10	Access control	2
10.3	<revoke statement>.....	2
12	Data manipulation	3
12.3	<fetch statement>.....	3
14	Control statements	3
14.6	<case statement>.....	3
15	Dynamic SQL	4
15.1	<prepare statement>.....	4

6 Scalar expressions

6.1 <value specification> and <target specification>

1. *Rationale: Use the correct syntax element.*

Replace General Rule 2) with:

- 2) Replace GR 15) A <simple target specification> specifies a target that is a host parameter, an output SQL parameter, a column of a new transition variable, a host variable, or an SQL variable, according to whether the <simple target specification> is a <host parameter name>, an <SQL parameter reference>, a <column reference>, or an <embedded variable name>, or an <SQL variable reference>, respectively.

NOTE 5 — A <simple target specification> can never be assigned the null value.

8 Additional common elements

8.1 <routine invocation>

1. *Rationale: Local temporary tables instantiated by an SQL-server module should not be visible to an external routine.*

Add the following General Rule:

- 1.1) Insert after GR 5)d)ii)1) Remove from RSC the identities of all instances of created local temporary tables that are referenced in <SQL-server module definition>s and all instances of declared local temporary tables that are defined by <temporary table definition>s that are contained in <SQL-server module definition>s.

9 Schema definition and manipulation

9.9 <drop table constraint definition>

1. *Rationale: Fix problems with contradictions between some symbol definitions and their usage and the use of undefined symbols.*

Replace General Rule 1) with:

- 1) Replace GR 2) Let *R* be any SQL-invoked routine whose routine descriptor contains the <constraint name> of *TC* in the <SQL routine body> or any SQL-invoked routine that is dependent on *TC*.

Case:

- a) If *R* is included in an SQL-server module *M*, then let *MN* be the <SQL-server module name> of *M*. The following <drop module statement> is effectively executed without further Access Rule checking once for every *M*:

```
DROP MODULE MN CASCADE
```

- b) Otherwise, let *SN* be the specific name of *R*. The following <drop routine statement> is effectively executed without further Access Rule checking for every *R*:

```
DROP SPECIFIC ROUTINE SN CASCADE
```

10 Access control

10.3 <revoke statement>

1. *Rationale: Handle dependencies of triggers and generated columns on SQL-server modules.*

Add the following General Rules:

- 3.1) Insert after GR 19)g) EXECUTE privilege on every SQL-server module that includes one or more SQL-invoked routines that are among the subject routines of any <routine invocation>, <method invocation>, <static method invocation>, or <method reference> that is contained in any <search condition> of *TR*.
- 3.2) Insert after GR 19)h) EXECUTE privilege on every SQL-server module that includes one or more SQL-invoked routines that are among the subject routines of any <routine invocation>, <method invocation>, <static method invocation>, or <method reference> that is contained in the <triggered SQL statement> of *TR*.
- 4.1) Insert after GR 22)c) *CD* has a generation expression *GE* and the revoke destruction action would result in *AI* no longer having in its applicable privileges EXECUTE privilege on any SQL-server

module that includes one or more SQL-invoked routines that are among the subject routines of any <routine invocation>, <method invocation>, <static method invocation> or <method reference> that is contained in *GE*.

12 Data manipulation

12.3 <fetch statement>

1. *Rationale: Correct the invocation of Subclause 9.2, “Store assignment”.*

Replace Syntax Rule 2) with:

- 2) Replace SR 9)a)i) If *TS* is an <SQL variable reference> or an <SQL parameter reference>, then the Syntax Rules of Subclause 9.2, “Store assignment”, in [ISO9075-2] apply to *TS* as *TARGET* and an arbitrary value of the row type of *T* as *VALUE*.

2. *Rationale: Correct the invocation of Subclause 9.2, “Store assignment”.*

Replace Syntax Rule 3) with:

- 3) Replace the lead text of SR 9)b)ii) For each <target specification> *TSI_i*, $1 \text{ (one)} \leq i, \leq \text{NTS}$, that is either an <SQL variable name>, an <SQL parameter reference>, or a <target array element specification>,

14 Control statements

14.6 <case statement>

1. *Rationale: Conformance Rules need to be applied after the syntactic transformation.*

Insert the following Syntax Rule:

- 1) ...
 - i.1) The Conformance Rules of Clause 8, “Predicates” in [ISO9075-2] are applied to the result of this syntactic transformation.

15 Dynamic SQL

15.1 <prepare statement>

1. *Rationale: Clarify the data type of dynamic parameters in <case statement>.*

Replace General Rule 5) with:

- 5) Insert after GR 5)a)xxxi) If *DP* is a <case operand> or <when operand> simply contained in a <simple case statement> *CS*, or if *DP* represents the value of a subfield *SF* of such a <case operand> or <when operand>, then:
 - a) let *RT* be the result of applying the Syntax Rules of Subclause 9.3, “Result of data type combinations”, in [ISO9075-2] to the set union of the following sets of declared types:
 - i) The set consisting of the declared type of the <case operand>.
 - ii) The set consisting of the declared type of any <when operand> of *CE* that is a <row value predicand>.
 - iii) The set consisting of the declared types of any <row value predicand> simply contained in any <comparison predicate part 2>, <between predicate part 2>, <character like predicate part 2>, <octet like predicate part 2>, <similar predicate part 2>, <regex like predicate part 2>, <overlaps predicate part 2>, <distinct predicate part 2>, or <member predicate part 2> that is simply contained in *CE*.
 - iv) The set consisting of the declared row type of a <table subquery> simply contained in an <in predicate part 2> or <quantified comparison predicate part 2> simply contained in *CE*.
 - v) The set consisting of the declared types of the <row value expression>s simply contained in an <in value list> simply contained in an <in predicate part 2> simply contained in *CE*.

NOTE — The following “part 2” predicates do not have any value expressions with declared type information: <null predicate part 2>, <normalized predicate part 2>, <set predicate part 2>, <type predicate part 2>.
 - b) Case:
 - i) If *DP* is a <case operand> or <when operand> simply contained in *CS*, then *DT* is *RT*.
 - ii) Otherwise, *DT* is the declared type of the subfield of *RT* that corresponds to *SF*.