



Information technology — Database languages — SQL —

Part 14: XML-Related Specifications (SQL/XML)

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Langages de base de données — SQL —

Partie 14: Spécifications relatives au XML (SQL/XML)

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 9075-14:2008 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

Statement of purpose for rationale

A statement indicating the rationale for each change to ISO/IEC 9075-14:2008 is included. This is to inform the users of ISO/IEC 9075-14:2008 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 Technical 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 Technical Corrigendum introduces new subclauses, the new subclauses have been numbered as follows:

Subclauses inserted between, for example, 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
4 Concepts.....	1
4.2 XML.....	1
4.2.3 Characteristics of XML values.....	1
4.2.5 Operations invoking XML values.....	1
10 Additional common rules.....	2
10.7 Determination of equivalent XML values.....	2
10.13 Construction of an XML element.....	2
10.16 Parsing a string as an XML value.....	3
11 Additional common elements.....	3
11.1 <routine invocation>.....	3
18 Embedded SQL.....	4
18.3 <embedded SQL C program>.....	4
18.4 <embedded SQL COBOL program>.....	5
18.5 <embedded SQL Fortran program>.....	5
18.6 <embedded SQL MUMPS program>.....	5
18.7 <embedded SQL Pascal program>.....	6
18.8 <embedded SQL PL/I program>.....	6

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 9075-14:2008/Cor 1:2010

Tables

Table	Page
4 XQuery node properties.	2

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 9075-14:2008/Cor 1:2010

(Blank page)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 9075-14:2008/Cor 1:2010

Information technology — Database languages — SQL —

Part 14:

XML-Related Specifications (SQL/XML)

TECHNICAL CORRIGENDUM 1

4 Concepts

4.2 XML

4.2.3 Characteristics of XML values

1. *Rationale: Editorial correction.*

Replace the 4th paragraph with:

Every XML value *XV* that is either

- the null value.
- a non-null value of type XML(CONTENT(ANY)) that is an XQuery document node *D* such that both of the following are true:
 - for every XQuery element node that is contained in the XQuery tree *T* rooted in *D*, the **type-name** property is **xs:untyped** and the **nilled** property is **false**, and
 - for every XQuery attribute node that is contained in *T*, the **type-name** property is **xs:untypedAtomic**

is a value of type XML(CONTENT(UNTYPED)).

4.2.5 Operations invoking XML values

1. *Rationale: Editorial correction.*

Replace the 6th paragraph with:

<XML PI> is an operator that returns an XML value, given an <identifier> and an optional character string *CS*. The XML value consists of an XQuery processing instruction node whose **target** property is the <identifier>, and whose **content** property is *CS*, trimmed of leading blanks and mapped to Unicode. This XQuery processing instruction node may optionally be placed as the sole child of an XQuery document node.

10 Additional common rules

10.7 Determination of equivalent XML values

1. *Rationale: Editorial correction.*

Replace “Table 4 — XQuery node properties” with:

Table 4 — XQuery node properties

XQuery Node Type	Significant Properties	Insignificant Properties
document	children unparsed entities	base-uri document-uri
element	node-name parent type-name children attributes nilled	namespaces base-uri
attribute	node-name string-value parent type	
processing instruction	target content parent	base-uri
text	content parent	
comment	content parent	
namespace	uri parent	prefix

10.13 Construction of an XML element

1. *Rationale: Editorial correction.*

Replace GR 4) c) iii) 1) B) III) with:

- 4) ...
- c) ...
 - iii) ...
 - 1) ...
 - B) ...
 - III) The **type-name** property is **xs:untypedAtomic**.

10.16 Parsing a string as an XML value

1. *Rationale: Editorial correction.*

Replace General Rule 5) with:

- 5) A string is called a *textual XML 1.1 content* if any of the following is true:
 - a) The character string is a textual XML 1.1 document.
 - b) The character string conforms to the definition of a well-formed external parsed entity as defined in [XML 1.1], as modified by [Namespaces 1.1].

11 Additional common elements

11.1 <routine invocation>

1. *Rationale: Correct the argument and the result in application of Subclause 9.2, “Store assignment” in [ISO9075-2].*

Replace General Rule 1) with:

- 1) Replace GR 3)a) If P_i is an input SQL parameter or both an input SQL parameter and an output SQL parameter, then apply the General Rules of Subclause 9.2, “Store assignment”, with V_i as *VALUE*, a temporary site *ST* whose declared type is T_i as *TARGET*, and, if T_i is an XML type, the <XML passing mechanism> of P_i as *PASSING*. Let CPV_i be the value of *ST*.

2. *Rationale: Correct the argument and the result in application of Subclause 9.2, “Store assignment” in [ISO9075-2].*

Replace General Rule 6) with:

- 6) Replace GR 9)a)iii) Apply the General Rules of Subclause 9.2, “Store assignment”, with *RV* as *VALUE* and some site *ST* whose declared type is *ERDT* as *TARGET*. Let the result of the <routine invocation> be the value of *ST*. If P_i is an input SQL parameter or both an input SQL parameter and an output SQL parameter, then apply the General Rules of Subclause 9.2, “Store assignment”, with RV_i as *VALUE*, some site *ST* whose declared type is *ERDT* as *TARGET*, and, if *ERDT* is an XML type, the <XML passing mechanism> of the <returns clause> of *R* as *PASSING*. Let the result of the <routine invocation> be the value of *ST*.

18 Embedded SQL

18.3 <embedded SQL C program>

1. *Rationale: Delete the incorrect reference to a character set in the description of an XML BLOB host variable.*

Replace Syntax Rule 2) with:

- 2) Insert after SR 5)f) The syntax

```
SQL TYPE IS XML XO XWO AS BLOB(L)
```

for a given <C host identifier> *hvn* shall be replaced by:

```
struct {
    long          hvn_reserved;
    unsigned long hvn_length;
    char          hvn_data[L];
} hvn
```

in any <C XML BLOB variable>, where:

- a) *L* is the numeric value of <large object length> as specified in Subclause 5.1, “<token> and <separator>”.
- b) *XO* is either <document or content> as specified in Subclause 6.8, “<string value function>”, or the zero-length string.
- c) *XWO* is either <XML whitespace option> as specified in Subclause 6.16, “<XML parse>”, or the zero-length string.

hvn is an *XML BLOB host variable*. *L* is the *length of XML BLOB host variable*. If *XO* is the zero-length string, then it is implementation-defined whether DOCUMENT or CONTENT is the XML option of XML BLOB host variable; otherwise *XO* is the *XML option of XML BLOB host variable*. If *XWO* is the zero-length string, then it is implementation-defined whether STRIP WHITESPACE or PRESERVE WHITESPACE is the *XML whitespace option of the XML BLOB host variable*; otherwise *XWO* is the *XML whitespace option of the XML BLOB host variable*.