INTERNATIONAL STANDARD

ISO/IEC 9075-14

Sixth edition
2023-06

Information technology — Database languages SQL —

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

Technologies de l'information — Langages de base de données SQL — Partie 14: Spécifications relatives au XML (SQL/XML)

Reference number
ISO/IEC 9075-14:2023(E)

Documents
Document Sharing Hub
© ISO/IEC 2023

Full document details: ISO IEC 09075-14-2023

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 09075-14-2023

Contents

Page

Foreword . . . ix
Introduction . . . xi
1 Scope . . . 1
2 Normative references . . . 2
3 Terms and definitions . . . 4
4 Concepts . . . 14
4.1 Notations and conventions . . . 14
4.1.1 Notations . . . 14
4.1.2 XML-related notations . . . 14
4.2 Data types . . . 15
4.2.1 Naming of predefined types . . . 15
4.2.2 Data type terminology . . . 16
4.3 XML . . . 16
4.3.1 Introduction . . . 16
4.3.2 XML types . . . 16
4.3.3 Characteristics of XML values . . . 17
4.3.4 XML comparison and assignment . . . 18
4.3.5 Operations involving XML values . . . 18
4.3.6 Registered XML schemas . . . 20
4.4 Data conversions . . . 21
4.5 Data analysis operations . . . 22
4.5.1 Aggregate functions . . . 22
4.6 SQL-invoked routines . . . 22
4.6.1 Routine descriptors . . . 22
4.7 SQL-statements . . . 23
4.7.1 SQL-statements classified by function . . . 23
4.7.1.1 SQL-session statements . . . 23
4.8 Basic security model . . . 23
4.8.1 Privileges . . . 23
4.9 SQL-sessions . . . 23
4.9.1 SQL-session properties . . . 23
4.10 XML namespaces . . . 24
4.11 Overview of mappings . . . 24
4.11.1 Introduction to mappings . . . 24
4.11.2 Mapping SQL character sets to Unicode . . . 25
4.11.3 Mapping Unicode to SQL character sets . . . 25
4.11.4 Mapping SQL <identifier>s to XML . . . 25
4.11.5 Mapping XML names to SQL . . . 25

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

4.11.6 Mapping SQL data types to XML. . . . 26
4.11.7 Mapping values of SQL data types to XML. . . . 27
4.11.8 Mapping XQuery atomic values to SQL values. . . . 28
4.11.9 Visibility of columns, tables, and schemas in mappings from SQL to XML. . . . 29
4.11.10 Mapping an SQL table to XML. . . . 29
4.11.11 Mapping an SQL schema to XML. . . . 30
4.11.12 Mapping an SQL catalog to XML. . . . 30

5 Lexical elements. . . . 32
5.1 <token> and <separator>. . . . 32
5.2 <literal>. . . . 34
5.3 Names and identifiers. . . . 35

6 Scalar expressions. . . . 36
6.1 <data type>. . . . 36
6.2 <field definition>. . . . 39
6.3 <value expression primary>. . . . 40
6.4 <value specification> and <target specification>. . . . 41
6.5 <case expression>. . . . 42
6.6 <cast specification>. . . . 43
6.7 <XML cast specification>. . . . 46
6.8 <value expression>. . . . 54
6.9 <string value function>. . . . 55
6.10 <XML value expression>. . . . 60
6.11 <XML value function>. . . . 61
6.12 <XML comment>. . . . 62
6.13 <XML concatenation>. . . . 64
6.14 <XML document>. . . . 66
6.15 <XML element>. . . . 68
6.16 <XML forest>. . . . 72
6.17 <XML parse>. . . . 75
6.18 <XML PI>. . . . 77
6.19 <XML query>. . . . 80
6.20 <XML text>. . . . 86
6.21 <XML validate>. . . . 88

7 Query expressions. . . . 93
7.1 <table reference>. . . . 93
7.2 <query expression>. . . . 97

8 Predicates. . . . 98
8.1 <predicate>. . . . 98
8.2 <XML content predicate>. . . . 99
8.3 <XML document predicate>. . . . 100
8.4 <XML exists predicate>. . . . 102
8.5 <XML valid predicate>. . . . 103

9 Mappings. . . . 108
9.1 Mapping SQL <identifier>s to XML names. . . . 108
9.2 Mapping a multi-part SQL name to an XML name. . . . 111
9.3 Mapping XML names to SQL <identifier>s. . . . 113

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

9.4 Mapping an SQL data type to an XML name. . . . 115
9.5 Mapping SQL data types to XML schema data types. . . . 120
9.6 Mapping an SQL data type to a named XML schema data type. . . . 139
9.7 Mapping a collection of SQL data types to XML schema data types. . . . 142
9.8 Mapping values of SQL data types to values of XML schema data types. . . . 144
9.9 Mapping an SQL table to XML schema data types. . . . 150
9.10 Mapping an SQL table to an XML element or a sequence of XML elements. . . . 154
9.11 Mapping an SQL table to XML and an XML schema document. . . . 158
9.12 Mapping an SQL schema to XML schema data types. . . . 163
9.13 Mapping an SQL schema to an XML element. . . . 166
9.14 Mapping an SQL schema to an XML document and an XML schema document. . . . 169
9.15 Mapping an SQL catalog to XML schema data types. . . . 174
9.16 Mapping an SQL catalog to an XML element. . . . 176
9.17 Mapping an SQL catalog to an XML document and an XML schema document. . . . 179

10 Additional common rules. . . . 184

10.1 Retrieval assignment. . . . 184
10.2 Store assignment. . . . 186
10.3 Result of data type combinations. . . . 188
10.4 Type precedence list determination. . . . 190
10.5 Type name determination. . . . 191
10.6 Determination of identical values. . . . 192
10.7 Determination of equivalent XML values. . . . 193
10.8 Equality operations. . . . 196
10.9 Grouping operations. . . . 197
10.10 Multiset element grouping operations. . . . 198
10.11 Ordering operations. . . . 199
10.12 Potential sources of non-determinism. . . . 200
10.13 Invoking an SQL-invoked routine. . . . 202
10.14 Determination of namespace URI. . . . 205
10.15 Construction of an XML element. . . . 207
10.16 Concatenation of two XML values. . . . 210
10.17 Serialization of an XML value. . . . 211
10.18 Parsing a string as an XML value. . . . 216
10.19 Removing XQuery document nodes from an XQuery sequence. . . . 220
10.20 Constructing a copy of an XML value. . . . 222
10.21 Constructing an unvalidated XQuery document node. . . . 223
10.22 Creation of an XQuery expression context. . . . 224
10.23 Determination of an XQuery formal type notation. . . . 227
10.24 Validating an XQuery document or element node. . . . 230

11 Additional common elements. . . . 232

11.1 <aggregate function>. . . . 232
11.2 <XML lexically scoped options>. . . . 235
11.3 <XML returning clause>. . . . 237
11.4 <XML passing mechanism>. . . . 238
11.5 <XML valid according to clause>. . . . 239

12 Schema definition and manipulation. . . . 242

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 09075-14-2023

12.1 <column definition>. 242
12.2 <check constraint definition>. 244
12.3 <alter column data type clause>. 245
12.4 <view definition>. 246
12.5 <assertion definition>. 248
12.6 <user-defined type definition>. 249
12.7 <attribute definition>. 250
12.8 <SQL-invoked routine>. 251
12.9 <user-defined cast definition>. 255

13 SQL-client modules. 256
13.1 <externally-invoked procedure>. 256
13.2 <SQL procedure statement>. 258
13.3 Data type correspondences. 259

14 Data manipulation. 261
14.1 <fetch statement>. 261
14.2 <select statement: single row>. 263
14.3 <delete statement: searched>. 265
14.4 <insert statement>. 266
14.5 <merge statement>. 267
14.6 <update statement: positioned>. 268
14.7 <update statement: searched>. 269

15 Control statements. 270
15.1 <compound statement>. 270
15.2 <assignment statement>. 272

16 Session management. 274
16.1 <set XML option statement>. 274

17 Dynamic SQL. 275
17.1 Description of SQL descriptor areas. 275
17.2 <input using clause>. 276
17.3 <output using clause>. 277
17.4 <prepare statement>. 278

18 Embedded SQL. 279
18.1 <embedded SQL host program>. 279
18.2 <embedded SQL Ada program>. 284
18.3 <embedded SQL C program>. 287
18.4 <embedded SQL COBOL program>. 291
18.5 <embedded SQL Fortran program>. 294
18.6 <embedded SQL Pascal program>. 297
18.7 <embedded SQL PL/I program>. 300

19 Call-Level Interface specifications. 304
19.1 SQL/CLI data type correspondences. 304

20 Diagnostics management. 306
20.1 <get diagnostics statement>. 306

21 Information Schema. 307
21.1 Information Schema digital artifact. 307

© ISO/IEC 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

21.2 NCNAME domain. . . . 307
21.3 URI domain. . . . 308
21.4 ATTRIBUTES view. . . . 309
21.5 COLUMNS view. . . . 310
21.6 DOMAINS view. . . . 311
21.7 ELEMENT_TYPES view. . . . 312
21.8 FIELDS view. . . . 313
21.9 METHOD_SPECIFICATION_PARAMETERS view. . . . 314
21.10 METHOD_SPECIFICATIONS view. . . . 315
21.11 PARAMETERS view. . . . 316
21.12 ROUTINES view. . . . 318
21.13 XML_SCHEMA_ELEMENTS view. . . . 320
21.14 XML_SCHEMA_NAMESPACES view. . . . 321
21.15 XML_SCHEMAS view. . . . 322
21.16 Short name views. . . . 323

22 Definition Schema. . . . 333
22.1 Definition Schema digital artifact. . . . 333
22.2 DATA_TYPE_DESCRIPTOR base table. . . . 333
22.3 PARAMETERS base table. . . . 339
22.4 ROUTINES base table. . . . 341
22.5 USAGE_PRIVILEGES base table. . . . 343
22.6 XML_SCHEMA_ELEMENTS base table. . . . 344
22.7 XML_SCHEMA_NAMESPACES base table. . . . 346
22.8 XML_SCHEMAS base table. . . . 347

23 SQL/XML XML schema. . . . 349

24 Status codes. . . . 353
24.1 SQLSTATE. . . . 353

25 Conformance. . . . 355
25.1 Claims of conformance to SQL/XML. . . . 355
25.2 Additional conformance requirements for SQL/XML. . . . 356
25.3 Implied feature relationships of SQL/XML. . . . 357

Annex A (informative) **SQL conformance summary. . . . 365**

Annex B (informative) **Implementation-defined elements. . . . 402**

Annex C (informative) **Implementation-dependent elements. . . . 417**

Annex D (informative) **SQL optional feature taxonomy. . . . 419**

Annex E (informative) **Deprecated features. . . . 425**

Annex F (informative) **Incompatibilities with ISO/IEC 9075:2016. . . . 426**

Annex G (informative) **Defect Reports not addressed in this document. . . . 427**

Bibliography. . . . 428

Index. . . . 429

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 09075-14-2023

Tables

Table

Page

1 Permanently registered XML schemas. 21
2 XML namespace prefixes and their URIs. 24
3 Constraining facets of XML schema integer types. 126
4 XQuery node properties. 194
5 Data type correspondences for Ada. 259
6 Data type correspondences for C. 259
7 Data type correspondences for COBOL. 259
8 Data type correspondences for Fortran. 259
9 Data type correspondences for M. 260
10 Data type correspondences for Pascal. 260
11 Data type correspondences for PL/I. 260
12 Codes used for SQL data types in Dynamic SQL. 275
13 SQL/CLI data type correspondences for Ada. 304
14 SQL/CLI data type correspondences for C. 304
15 SQL/CLI data type correspondences for COBOL. 304
16 SQL/CLI data type correspondences for Fortran. 305
17 SQL/CLI data type correspondences for M. 305
18 SQL/CLI data type correspondences for Pascal. 305
19 SQL/CLI data type correspondences for PL/I. 305
20 SQL-statement codes. 306
21 SQLSTATE class and subclass codes. 353
22 Implied feature relationships of SQL/XML. 357
A.1 Feature definitions outside of Conformance Rules. 365
D.1 Feature taxonomy for optional features. 419

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC have not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

This sixth edition cancels and replaces the fifth edition (ISO/IEC 9075-14:2016), which has been technically revised. It also incorporates the Technical Corrigenda ISO/IEC 9075-14:2016/Cor.1:2019 and ISO/IEC 9075-14:2016/Cor.2:2022.

The main changes are as follows:

— improve the presentation and accuracy of the summaries of implementation-defined and implementation-dependent aspects of this document;

— introduction of several digital artifacts;

— alignment with updated ISO house style and other guidelines for creating standards.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 09075-14-2023

This sixth edition of ISO/IEC 9075-14 is designed to be used in conjunction with the following editions of other parts of the ISO/IEC 9075 series, all published in 2023:

— ISO/IEC 9075-1, sixth edition;

— ISO/IEC 9075-2, sixth edition;

— ISO/IEC 9075-3, sixth edition;

— ISO/IEC 9075-4, seventh edition;

— ISO/IEC 9075-9, fifth edition;

— ISO/IEC 9075-10, fifth edition;

— ISO/IEC 9075-11, fifth edition;

— ISO/IEC 9075-13, fifth edition;

— ISO/IEC 9075-15, second edition;

— ISO/IEC 9075-16, first edition.

A list of all parts in the ISO/IEC 9075 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

Introduction

The organization of this document is as follows:

1) Clause 1, "Scope", specifies the scope of this document.

2) Clause 2, "Normative references", identifies additional standards that, through reference in this document, constitute provisions of this document.

3) Clause 3, "Terms and definitions", defines the terms and definitions used in this document.

4) Clause 4, "Concepts", presents concepts related to this document.

5) Clause 5, "Lexical elements", defines the lexical elements of the language.

6) Clause 6, "Scalar expressions", defines the elements of the language that produce scalar values.

7) Clause 7, "Query expressions", defines the elements of the language that produce rows and tables of data.

8) Clause 8, "Predicates", defines the predicates of the language.

9) Clause 9, "Mappings", defines the ways in which certain SQL information can be mapped into XML and certain XML information can be mapped into SQL.

10) Clause 10, "Additional common rules", specifies the rules for assignments that retrieve data from or store data into SQL-data, and formation rules for set operations.

11) Clause 11, "Additional common elements", defines additional language elements that are used in various parts of the language.

12) Clause 12, "Schema definition and manipulation", defines facilities for creating and managing a schema.

13) Clause 13, "SQL-client modules", defines SQL-client modules and externally-invoked procedures.

14) Clause 14, "Data manipulation", defines the data manipulation statements.

15) Clause 15, "Control statements", defines the SQL-control statements.

16) Clause 16, "Session management", defines the SQL-session management statements.

17) Clause 17, "Dynamic SQL", defines the SQL dynamic statements.

18) Clause 18, "Embedded SQL", defines the host language embeddings.

19) Clause 19, "Call-Level Interface specifications",

20) Clause 20, "Diagnostics management", defines the diagnostics management facilities.

21) Clause 21, "Information Schema", defines viewed tables that contain schema information.

22) Clause 22, "Definition Schema", defines base tables on which the viewed tables containing schema information depend.

23) Clause 23, "SQL/XML XML schema", defines the content of an XML namespace that is used when SQL and XML are utilized together.

24) Clause 24, "Status codes", defines values that identify the status of the execution of SQL-statements and the mechanisms by which those values are returned.

25) Clause 25, "Conformance", specifies the way in which conformance to this document may be claimed.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 09075-14-2023

26) Annex A, "SQL conformance summary", is an informative Annex. It summarizes the conformance requirements of the SQL language.

27) Annex B, "Implementation-defined elements", is an informative Annex. It lists those features for which the body of this document states that the syntax, the meaning, the returned results, the effect on SQL-data and/or schemas, or other aspect is partly or wholly implementation-defined.

28) Annex C, "Implementation-dependent elements", is an informative Annex. It lists those features for which the body of this document states that the syntax, the meaning, the returned results, the effect on SQL-data and/or schemas, or other aspect is partly or wholly implementation-dependent.

29) Annex D, "SQL optional feature taxonomy", is an informative Annex. It identifies the optional features of the SQL language specified in this document by an identifier and a short descriptive name. This taxonomy is used to specify conformance.

30) Annex E, "Deprecated features", is an informative Annex. It lists features that the responsible Technical Committee intends not to include in a future edition of this document.

31) Annex F, "Incompatibilities with ISO/IEC 9075:2016", is an informative Annex. It lists incompatibilities with the previous edition of this document.

32) Annex G, "Defect Reports not addressed in this document", is an informative Annex. It describes the Defect Reports that were known at the time of publication of this document. Each of these problems is a problem carried forward from the previous edition of the ISO/IEC 9075 series. No new problems have been created in the drafting of this edition of this document.

In the text of this document, in Clause 5, "Lexical elements", through Clause 25, "Conformance", Subclauses begin new pages. Any resulting blank space is not significant.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

Information technology — Database language SQL —

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

1 Scope

This document defines ways in which Database Language SQL can be used in conjunction with XML.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies. [1] [2]

ISO/IEC 9075-1, *Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework)*

ISO/IEC 9075-2, *Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)*

ISO/IEC 9075-3, *Information technology — Database languages — SQL — Part 3: Call-Level Interface (SQL/CLI)*

ISO/IEC 9075-4, *Information technology — Database languages — SQL — Part 4: Persistent Stored Modules (SQL/PSM)*

ISO/IEC 9075-11, *Information technology — Database languages — SQL — Part 11: Information and Definition Schemas (SQL/Schemata)*

W3C Canonical XML 1.1 *Canonical XML, W3C Recommendation*. Edited by: Boyer, John; Marcy, Glenn 2 May 2008
Available at: https://www.w3.org/TR/xml-c14n

W3C XML Information Set *XML Information Set, W3C Recommendation.*. Edited by: Cowan, John; Tobin, Richard 4 February 2004
Available at: https://www.w3.org/TR/xml-infoset

W3C XML Namespaces *XML Namespaces is used to reference either Namespaces in XML 1.0 or Namespaces in XML 1.1 when there is no significant difference between the two for the purposes of a given citation*

W3C Namespaces in XML 1.0 *Namespaces in XML 1.0, W3C Recommendation*. Edited by: Bray, Tim, et al. 8 December 2009
Available at: https://www.w3.org/TR/xml-names

W3C Namespaces in XML 1.1 *Namespaces in XML 1.1, W3C Recommendation*. Edited by: Bray, Tim, et al. 16 August 2006
Available at: https://www.w3.org/TR/xml-names11

Internet Engineering Task Force (IETF) RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*. Edited by: Berners-Lee, T., Fielding, R., Masinter, L. January 2005
Available at: https://www.ietf.org/rfc/rfc3986.txt

W3C XML Schema Part 1: Structures *XML Schema Part 1: Structures, W3C Recommendation*. Edited by: Thompson, Henry S et al. 28 October 2004
Available at: https://www.w3.org/TR/xmlschema-1/

W3C XML Schema Part 2: Datatypes *XML Schema Part 2: Datatypes, W3C Recommendation*. Edited by: Biron, Paul V.; Malhotra, Ashok 28 October 2004
Available at: https://www.w3.org/TR/xmlschema-2/

1 XML Namespaces is used to reference either Namespaces in XML 1.0 or Namespaces in XML 1.1 when there is no significant difference between the two for the purposes of a given citation.

2 XML is used to reference either XML 1.0 or XML 1.1 when there is no significant difference between the two for the purposes of a given citation.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

W3C XSLT and XQuery Serialization 3.1 *XSLT and XQuery Serialization 3.1, W3C Recommendation*. Edited by: Coleman Andrew; Sperberg-McQueen, C M. 21 March 2017
Available at: https://www.w3.org/TR/xslt-xquery-serialization-31/

The Unicode Consortium. *The Unicode Standard (Information about the latest version of the Unicode standard can be found by using the "Latest Version" link on the "Enumerated Versions of The Unicode Standard" page.)* [online]. Mountain View, California, USA: The Unicode Consortium, Available at https://www.unicode.org/versions/enumeratedversions.html

W3C Unicode in XML and Other Markup Languages *Unicode in XML and Other Markup Languages, W3C Working Group Note*. Edited by: Phillips, Addison 13 July 2017
Available at: https://www.w3.org/TR/unicode-xml/

XML is used to reference either XML 1.0 or XML 1.1 when there is no significant difference between the two for the purposes of a given citation . [Place of publication unknown]:

W3C XML 1.0 *Extensible Markup Language (XML) Version 1.0, W3C Recommendation*. Edited by: Bray, Tim et al. 26 November 2008, revised 7 February 2013
Available at: https://www.w3.org/TR/xml

W3C XML 1.1 *Extensible Markup Language (XML) Version 1.1, W3C Recommendation*. Edited by: Bray, Tim et al. 16 August 2006, revised 29 September 2006
Available at: https://www.w3.org/TR/xml11

W3C XML Path Language (XPath) 3.1 *XML Path Language (XPath) 3.1, W3C Recommendation*. Edited by: Robie, J; Dyck, M; & Spiegel, J. 21 March 2017
Available at: https://www.w3.org/TR/xpath-31/

W3C XQuery 3.1: an XML Query Language *XQuery 3.1: an XML Query Language, W3C Recommendation*. Edited by: Robie, J; Dyck, M; & Spiegel, J. 21 March 2017
Available at: https://www.w3.org/TR/xquery-31/

W3C XQuery and XPath Data Model 3.1 *XQuery and XPath Data Model 3.1, W3C Recommendation*. Edited by: Walsh, N.; Snelson, J.; & Coleman, A. 21 March 2017
Available at: https://www.w3.org/TR/xpath-datamodel/

W3C XQuery and XPath Functions and Operators 3.1 *XQuery and XPath Functions and Operators 3.1, W3C Recommendation*. Edited by: Malhotra, Ashok et al. 21 March 2017
Available at: https://www.w3.org/TR/xpath-functions/

W3C XQuery 1.0 and XPath 2.0 Formal Semantics *XQuery 1.0 and XPath 2.0 Formal Semantics, W3C Recommendation*. Edited by: Draper, Denise, et al. 14 December 2010, revised 7 September 2015
Available at: https://www.w3.org/TR/xquery-semantics/

W3C XQuery Update Facility 1.0 *XQuery Update Facility 1.0, W3C Recommendation*. Edited by: Robie, Jonathan et al. 17 March 2011
Available at: https://www.w3.org/TR/xquery-update-10/

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 9075-1, ISO/IEC 9075-2, and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at `https://www.iso.org/obp`

— IEC Electropedia: available at `https://www.electropedia.org/`

3.1
DTD
XML 1.0 document type declaration (3.2) or *XML 1.1 document type declaration* (3.3)

3.2
XML 1.0 document type declaration
declaration that contains or points to markup declarations that provide a grammar for a class of documents as defined in XML 1.0

[SOURCE: XML 1.0, 2.8]

3.3
XML 1.1 document type declaration
declaration that contains or points to markup declarations that provide a grammar for a class of documents as defined in XML 1.1

[SOURCE: XML 1.1, 2.8]

3.4
XML document
XML 1.0 document (3.5) or *XML 1.1 document* (3.6)

3.5
XML 1.0 document
well-formed XML document, as defined in XML 1.0

[SOURCE: XML 1.0, 2.1]

3.6
XML 1.1 document
well-formed XML document, as defined in XML 1.1

[SOURCE: XML 1.1, 2.1]

3.7
annotation
annotation schema component, as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 3.13]

3.8
facet
facet, as defined in XML Schema Part 2: Datatypes

[SOURCE: XML Schema Part 2: Datatypes, 2.4]

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.9
wildcard schema component
wildcard schema component, as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 3.10.2]

3.10
XML schema
XML Schema, as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 2.1]

3.11
all group content model
content model of an XML schema complex type described with **`xs:all`** as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 3.8]

3.12
canonical XML schema literal
canonical lexical representation for an XML schema type ***XST***, as defined in XML Schema Part 2: Datatypes, based on Canonical XML 1.1

Note 1 to entry: There is a unique canonical XML schema literal for each value in the value space of ***XST***.

[SOURCE: XML Schema Part 2: Datatypes, 2.3.1]

3.13
empty XQuery sequence
empty sequence, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 2.5]

3.14
global element declaration schema component
element declaration schema component of a global element declaration, as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 3.3.2]

3.15
metadata
data about data

Note 1 to entry: In this document, metadata is included in table descriptors, column descriptors, and so forth, as defined in ISO/IEC 9075-2

3.16
sequence content model
content model of an XML schema complex type described with **`xs:sequence`** as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 3.3.2]

3.17
SQL value space
set of all values for a particular SQL <data type>

3.18
URI
Uniform Resource Identifier as defined in RFC 3986

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.19
valid XML character
if Feature X211, "XML 1.1 support", is supported, then a *valid XML 1.1 character* (3.21); otherwise, a *valid XML 1.0 character* (3.20)

3.20
valid XML 1.0 character
valid character as defined in XML 1.0, rule [2], "Char"

[SOURCE: XML 1.0, 2.2]

3.21
valid XML 1.1 character
valid character as defined in XML 1.1, rule [2], "Char"

[SOURCE: XML 1.1, 2.2]

3.22
XML attribute
XML 1.0 attribute (3.23) or *XML 1.1 attribute* (3.24)

3.23
XML 1.0 attribute
attribute, as defined in XML 1.0

[SOURCE: XML 1.0, 3.3]

3.24
XML 1.1 attribute
attribute, as defined in XML 1.1

[SOURCE: XML 1.1, 3.3]

3.25
XML attribute information item
attribute information item, as defined in XML Information Set

[SOURCE: XML Information Set, 2.3]

3.26
XML character information item
character information item, as defined in XML Information Set

[SOURCE: XML Information Set, 2.6]

3.27
XML declaration
XML 1.0 declaration (3.28) or *XML 1.1 declaration* (3.29)

3.28
XML 1.0 declaration
XMLDecl, as defined in XML 1.0

[SOURCE: XML 1.0, 2.8]

3.29
XML 1.1 declaration
XMLDecl, as defined in XML 1.1

[SOURCE: XML 1.1, 2.8]

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.30
XML document information item
document information item, as defined in XML Information Set

[SOURCE: XML Information Set, 2.1]

3.31
XML element
XML 1.0 element (3.32) or *XML 1.1 element* (3.33)

3.32
XML 1.0 element
element, as defined in XML 1.0

[SOURCE: XML 1.0, 3.2]

3.33
XML 1.1 element
element, as defined in XML 1.1

[SOURCE: XML 1.1, 3.2]

3.34
XML element information item
element information item, as defined in XML Information Set

[SOURCE: XML Information Set, 2.2]

3.35
XML information item
information item, as defined in XML Information Set

[SOURCE: XML Information Set, 2]

3.36
XML 1.0 Name
Name as defined by XML 1.0

[SOURCE: XML 1.0, 2.3]

3.37
XML 1.1 Name
Name as defined by XML 1.1

[SOURCE: XML 1.1, 2.3]

3.38
XML 1.0 NameChar
NameChar as defined by XML 1.0

[SOURCE: XML 1.0, 2.3]

3.39
XML 1.1 NameChar
NameChar as defined by XML 1.1

[SOURCE: XML 1.1, 2.3]

3.40
XML 1.1 NameStartChar
NameStartChar as defined by XML 1.1

[SOURCE: XML 1.1, 2.3]

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.41
XML namespace
XML 1.0 namespace (3.42) or *XML 1.1 namespace* (3.43)

3.42
XML 1.0 namespace
XML namespace, as defined in Namespaces in XML 1.0

[SOURCE: Namespaces in XML 1.0, 2]

3.43
XML 1.1 namespace
XML namespace, as defined in Namespaces in XML 1.1

[SOURCE: Namespaces in XML 1.1, 2]

3.44
XML namespace prefix
XML 1.0 namespace prefix (3.45) or *XML 1.1 namespace prefix* (3.46)

3.45
XML 1.0 namespace prefix
XML namespace prefix, as defined in Namespaces in XML 1.0

[SOURCE: Namespaces in XML 1.0, 3]

3.46
XML 1.1 namespace prefix
XML namespace prefix, as defined in Namespaces in XML 1.1

[SOURCE: Namespaces in XML 1.1, 3]

3.47
XML 1.0 NCName
NCName, as defined by rule [4] in Namespaces in XML 1.0

[SOURCE: Namespaces in XML 1.0, 3]

3.48
XML 1.1 NCName
NCName, as defined by rule [4] in Namespaces in XML 1.1

[SOURCE: Namespaces in XML 1.1, 3]

3.49
XML 1.0 QName
QName, as defined by rule [7] of Namespaces in XML 1.0

[SOURCE: XML 1.0, 4]

3.50
XML 1.1 QName
QName, as defined by rule [7] of Namespaces in XML 1.1

[SOURCE: XML 1.1, 4]

3.51
XML Qname prefix
XML 1.0 QName prefix (3.52) or *XML 1.1 QName prefix* (3.53)

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.52
XML 1.0 QName prefix
Prefix, as defined by rule [10] of Namespaces in XML 1.0

[SOURCE: Namespaces in XML 1.0, 4]

3.53
XML 1.1 QName prefix
Prefix, as defined by rule [10] of Namespaces in XML 1.1

[SOURCE: Namespaces in XML 1.1, 4]

3.54
XML 1.0 QName local part
LocalPart, as defined by rule [11] of Namespaces in XML 1.0

[SOURCE: Namespaces in XML 1.0, 4]

3.55
XML 1.1 QName local part
LocalPart, as defined by rule [11] of Namespaces in XML 1.1

[SOURCE: Namespaces in XML 1.1, 4]

3.56
XML schema built-in data type
built-in datatype, as defined in XML Schema Part 2: Datatypes

[SOURCE: XML Schema Part 2: Datatypes, 3]

3.57
XML schema complex type
complex type defined by a complex type definition, as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 3.4]

3.58
XML schema data type
datatype, as defined in XML Schema Part 2: Datatypes

[SOURCE: XML Schema Part 2: Datatypes, A]

3.59
XML schema document
schema document, as defined in XML Schema Part 1: Structures

[SOURCE: XML Schema Part 1: Structures, 4.2]

3.60
XML schema primitive type
primitive type, as defined in XML Schema Part 2: Datatypes

[SOURCE: XML Schema Part 2: Datatypes, 4.1.1]

3.61
XML schema simple type
simple type defined by a simple type definition, as defined in XML Schema Part 2: Datatypes

[SOURCE: XML Schema Part 2: Datatypes, 4.1.2]

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.62
XML schema type
term used to collectively refer to *XML schema built-in data types* (3.56), *XML schema complex types* (3.57), *XML schema data types* (3.58), and *XML schema simple types* (3.61), when it is not necessary to distinguish among those terms

3.63
XML target namespace
target namespace as defined by XML Schema Part 1: Structures

3.64
XML target namespace URI
URI (3.18) of an *XML target namespace* (3.63)

3.65
XML text
character string that is a substring of a textual XML 1.0 content or a textual XML 1.1 content, as defined in Subclause 10.18, "Parsing a string as an XML value"

3.66
XML value space
value space, as defined by XML Schema Part 2: Datatypes

[SOURCE: XML Schema Part 2: Datatypes, 2.2]

3.67
XQuery accessor
accessor, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 5]

3.68
XQuery atomic type
atomic type, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 2.1]

3.69
XQuery atomic value
atomic value, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 2.7.5]

3.70
XQuery attribute node
attribute node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 3.3.1.1]

3.71
XQuery comment node
comment node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6.6]

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.72
XQuery datetime normalized value
normalized value of a value of XML schema types `xs:dateTime`, `xs:date`, `xs:time`, or any type derived from these types, as this term is used in XQuery and XPath Functions and Operators 3.1, and implicitly defined in XQuery and XPath Data Model 3.1, section 3.3.2 "Dates and Times"

[SOURCE: XQuery and XPath Data Model 3.1, 3.3.2]

3.73
XQuery datetime time zone component
time zone component of a value of XML schema types `xs:dateTime`, `xs:date`, `xs:time`, or any type derived from these types, as this term is used in XQuery and XPath Functions and Operators 3.1 and implicitly defined in XQuery and XPath Data Model 3.1, section 3.3.2, "Dates and Times"

[SOURCE: XQuery and XPath Data Model 3.1, 3.3.2]

3.74
XQuery document node
document node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6.1]

3.75
XQuery dynamic context
dynamic context, as defined in XQuery 3.1: an XML Query Language

[SOURCE: XQuery 3.1: an XML Query Language, 2.1.2]

3.76
XQuery element node
element node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6.2]

3.77
XQuery error
static error, type error or dynamic error, as defined in XQuery 3.1: an XML Query Language

[SOURCE: XQuery 3.1: an XML Query Language, 2.3]

3.78
XQuery evaluation with XML 1.0 lexical rules
process of determining the value of an XQuery expression with XML 1.0 lexical rules, as defined in XQuery 3.1: an XML Query Language

[SOURCE: XQuery 3.1: an XML Query Language, 2.2]

3.79
XQuery evaluation with XML 1.1 lexical rules
process of determining the value of an XQuery expression with XML 1.1 lexical rules, as defined in XQuery 3.1: an XML Query Language

[SOURCE: XQuery 3.1: an XML Query Language, 2.2]

3.80
XQuery expression
XQuery expression with XML 1.0 lexical rules, or an XQuery expression with XML 1.1 lexical rules

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.81
XQuery expression context
expression context, consisting of a static context and a dynamic context, as defined in XQuery 3.1: an XML Query Language

[SOURCE: XQuery 3.1: an XML Query Language, 2.1]

3.82
XQuery expression with XML 1.0 lexical rules
expression, as defined by rule [31], "Expr", in XQuery 3.1: an XML Query Language, with rules [4], "NCName", and [7], "QName", interpreted to reference Namespaces in XML 1.0

[SOURCE: XQuery 3.1: an XML Query Language, 4.17]

3.83
XQuery expression with XML 1.1 lexical rules
expression, as defined by rule [31], "Expr", in XQuery 3.1: an XML Query Language, with rules [4], "NCName", and [7], "QName", interpreted to reference Namespaces in XML 1.1

[SOURCE: XQuery 3.1: an XML Query Language, 4.17]

3.84
XQuery formal type notation
formal type notation, as defined by rule [24 (Formal)] "Type", in XQuery 1.0 and XPath 2.0 Formal Semantics

3.85
XQuery item
item, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 2.8]

3.86
XQuery namespace node
namespace node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6.4]

3.87
XQuery node
node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6]

3.88
XQuery node identity
node identity, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 2.3]

3.89
XQuery node property
property of a node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6]

3.90
XQuery processing instruction node
processing instruction node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6.5]

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3.91
XQuery sequence
sequence, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 2.5]

3.92
XQuery static context
static context, as defined in XQuery 3.1: an XML Query Language

[SOURCE: XQuery 3.1: an XML Query Language, 2.1.1]

3.93
XQuery text node
text node, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 6.7]

3.94
XQuery tree
tree, as defined in XQuery and XPath Data Model 3.1

[SOURCE: XQuery and XPath Data Model 3.1, 2.1]

3.95
XQuery variable
variable, as defined in XQuery 3.1: an XML Query Language

[SOURCE: XQuery 3.1: an XML Query Language, 4.13]

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

4 Concepts

This Clause modifies Clause 4, "Concepts", in ISO/IEC 9075-2.

4.1 Notations and conventions

This Subclause modifies Subclause 4.1, "Notations and conventions", in ISO/IEC 9075-2.

4.1.1 Notations

This Subclause modifies Subclause 4.1.1, "Notations", in ISO/IEC 9075-2.

The notations used in this document are defined in ISO/IEC 9075-1.

The syntax defined in this document is available from the ISO website as a "digital artifact". See `https://standards.iso.org/iso-iec/9075/-14/ed-6/en/` to download digital artifacts for this document. To download the syntax defined in a plain-text format, select the file named `ISO_IEC_9075-14(E)_XML.bnf.txt`. To download the syntax defined in an XML format, select the file named `ISO_IEC_9075-14(E)_XML.bnf.xml`.

4.1.2 XML-related notations

Many of the standards referenced in Clause 2, "Normative references", are produced and maintained by the World Wide Web Consortium, including XML, XML Path Language (XPath) 3.1, XML Namespaces, XML Schema Part 1: Structures, XML Schema Part 2: Datatypes, XML Information Set, XQuery 3.1: an XML Query Language. W3C calls these international standards "Recommendations", and places them on its web site (`https://www.w3.org`). Whenever this document mentions a value, type, or other object (broadly conceived) that is specified by a W3C Recommendation, then that object is indicated using bold monospace font (for example, **`<xs:element>`**).

Similarly, when a textual variable in a Rule denotes an object that is specified by a W3C Recommendation, then the variable is written in italicized bold monospace font (for example, ***`FACETP`***).

The following list enumerates many, but not necessarily all, of the kinds of objects that are specified by W3C Recommendations.

— XML text;

— XML values;

— XML namespaces and XML namespace prefixes;

— XML schema types;

— XQuery types;

— XQuery nodes;

— XQuery node properties;

— XQuery expressions;

— XQuery functions;

— XQuery variables;

— XQuery formal type notations;

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

— XQuery static and dynamic contexts.

On the other hand, an object that has some feature or aspect that is not specified by W3C is not in bold. For example, W3C objects do not have the ability to be null. Since a value of XML type may be null, it is not represented in bold. As another example, a registered XML schema has an SQL privilege associated with it. In addition, non-bold variables are used for SQL BNF non-terminals, even when they are used to identify W3C objects.

This document specifies an interface between SQL and objects specified by W3C. As such, some values may be both W3C objects and also SQL objects. Therefore no notational convention can be totally consistent. In particular, it is permitted to assign a bold value to a non-bold variable, or to assign a non-bold value to a bold variable (provided the non-bold value has been determined to have no SQL-only feature, such as being a null). It is also permitted to compare a bold value and a non-bold value.

Whenever XML text is presented, an SQL/XML implementation may substitute equivalent XML text, for example, through insertion or deletion of insignificant blanks or new lines.

In this document, <left bracket>/<right bracket> pairs occur in five contexts.

— As part of BNF productions, in which a pair of brackets surrounds one or more non-terminal and/or terminal symbols that are, treated as a single group, optional.

— In code examples, in which the brackets are literal characters that represent themselves as part of the code.

— In ordinary text, in which a pair of brackets enclose a word or phrase that identifies a publication referenced in Clause 2, "Normative references".

— In ordinary text, in which a pair of brackets enclose a number that identifies the number of a BNF production specified in the publication whose reference appears nearby.

— In ordinary text, in which a pair of brackets enclose a word or phrase that identifies the name of a property defined in XML Information Set.

In this document, <left brace>/<right brace> pairs occur in three contexts.

— As part of BNF productions, in which a pair of braces surrounds one or more non-terminal and/or terminal symbols that are to be handled as a single group.

— In code examples, in which the braces are literal characters that represent themselves as part of the code.

— In ordinary text, in which the braces enclose a word or phrase that identifies the name of a property defined in XML Schema Part 1: Structures.

4.2 Data types

This Subclause modifies Subclause 4.2, "Data types", in ISO/IEC 9075-2.

4.2.1 Naming of predefined types

This Subclause modifies Subclause 4.2.2, "Naming of predefined types", in ISO/IEC 9075-2.

Insert after the 1st paragraph: SQL defines a predefined data type named by the following <key word>: XML.

Insert after the 3rd paragraph: The data types XML(DOCUMENT(UNTYPED)), XML(DOCUMENT(ANY)), XML(DOCUMENT(XMLSCHEMA)), XML(CONTENT(UNTYPED)), XML(CONTENT(ANY)), XML(CONTENT(XMLSCHEMA)), and XML(SEQUENCE) are referred to as the *XML types*. Values of XML types are called *XML values*.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

4.2.2 Data type terminology

This Subclause modifies Subclause 4.2.4, "Data type terminology", in ISO/IEC 9075-2.

Insert into the 11th paragraph, after the last list item:

— A type *T* is *XML-ordered* if *T* is *S*-ordered, where *S* is the set consisting of the XML types.

4.3 XML

4.3.1 Introduction

Fuller descriptions of XML concepts can be found in the documents about XML included in Clause 2, "Normative references".

4.3.2 XML types

An XML type is described by an XML type descriptor. An XML type descriptor contains the following items.

— The name of the data type (XML).

— The primary XML type modifier described by the <primary XML type modifier> (DOCUMENT, CONTENT, or SEQUENCE).

— The secondary XML type modifier described by the <secondary XML type modifier> (UNTYPED, ANY, or XMLSCHEMA), if any.

— The registered XML schema descriptor of the indicated registered XML schema, if any.

— The XML namespace URI of the indicated XML namespace, if any.

— The XML NCName of the indicated global element declaration schema component, if any.

NOTE 1 — "Indicated registered XML schema", "indicated XML namespace", and "indicated global element declaration schema component" are defined in Subclause 11.5, "<XML valid according to clause>".

An XML type whose primary XML type modifier is DOCUMENT and whose secondary XML type modifier is UNTYPED is called an XML(DOCUMENT(UNTYPED)) type.

An XML type whose primary XML type modifier is DOCUMENT and whose secondary XML type modifier is ANY is called an XML(DOCUMENT(ANY)) type.

An XML type whose primary XML type modifier is DOCUMENT and whose secondary XML type modifier is XMLSCHEMA is called an XML(DOCUMENT(XMLSCHEMA)) type.

NOTE 2 — The number of XML(DOCUMENT(XMLSCHEMA)) types is determined by the number of registered XML schemas, the number of XML namespaces in those XML schemas, and the number of global element declaration schema components in those XML schemas.

An XML type whose primary XML type modifier is CONTENT and whose secondary XML type modifier is UNTYPED is called an XML(CONTENT(UNTYPED)) type.

An XML type whose primary XML type modifier is CONTENT and whose secondary XML type modifier is ANY is called an XML(CONTENT(ANY)) type.

An XML type whose primary XML type modifier is CONTENT and whose secondary XML type modifier is XMLSCHEMA is called an XML(CONTENT(XMLSCHEMA)) type.

NOTE 3 — The number of XML(CONTENT(XMLSCHEMA)) types is determined by the number of registered XML schemas, the number of XML namespaces in those XML schemas, and the number of global element declaration schema components in those XML schemas.

An XML type whose primary XML type modifier is SEQUENCE is called an XML(SEQUENCE) type.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

When determining the declared type of an <XML value expression>, an SQL-implementation is permitted to use the XML(DOCUMENT(ANY)) type in place of XML(DOCUMENT(UNTYPED)) and XML(DOCUMENT(XMLSCHEMA)) types, the XML(CONTENT(ANY)) type in place of XML(CONTENT(UNTYPED)), XML(CONTENT(XMLSCHEMA)), and XML(DOCUMENT(ANY)) types, and the XML(SEQUENCE) type in place of the XML(CONTENT(ANY)) type.

NOTE 4 — The above substitutions can be applied transitively. For instance, an SQL-implementation is permitted to use the XML(SEQUENCE) type in place of the XML(DOCUMENT(UNTYPED)) type.

4.3.3 Characteristics of XML values

An *XML value* is either the null value or an XQuery sequence.

Every XML value is a value of type XML(SEQUENCE).

Every XML value that is either the null value or an XQuery document node is a value of type XML(CONTENT(ANY)).

Every XML value that is either:

— the null value;

— a non-null value of type XML(CONTENT(ANY)) that is an XQuery document node ***D*** such that all 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`,
- 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)).

Every XML value that is either:

— the null value;

— a non-null value of type XML(CONTENT(ANY)) that is an XQuery document node ***D*** such that every XQuery element node that is contained in the XQuery tree ***T*** rooted in ***D*** is valid according to at least one of the following:

- an XML schema ***S***,
- an XML namespace ***N*** in an XML schema ***S***,
- a global element declaration schema component ***E*** in an XML schema ***S***.

is a value of type XML(CONTENT(XMLSCHEMA)) whose type descriptor includes the registered XML schema descriptor of ***S*** and, if ***N*** is specified, the XML namespace URI of ***N*** or, if ***E*** is specified, the XML namespace URI of ***E*** and the XML NCName of ***E***.

Every XML value that is either:

— the null value;

— a non-null value of type XML(CONTENT(ANY)) that is an XQuery document node whose **children** property has exactly one XQuery element node, zero or more XQuery comment nodes, and zero or more XQuery processing instruction nodes.

is a value of type XML(DOCUMENT(ANY)).

Every XML value that is either:

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

— the null value;

— a non-null value of type XML(CONTENT(UNTYPED)) that is an XQuery document node whose **children** property has exactly one XQuery element node, zero or more XQuery comment nodes, and zero or more XQuery processing instruction nodes.

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

Every XML value that is either:

— the null value;

— a non-null value of type XML(DOCUMENT(ANY)) that is valid according to at least one of the following:

- an XML schema ***S***,
- an XML namespace ***N*** in an XML schema ***S***,
- a global element declaration schema component ***E*** in an XML schema ***S***.

is a value of type XML(DOCUMENT(XMLSCHEMA)) whose type descriptor includes the registered XML schema descriptor of ***S*** and, if ***N*** is specified, the XML namespace URI of ***N***, or, if ***E*** is specified, the XML namespace URI of ***E*** and the XML NCName of ***E***.

Two XML values are regarded as equivalent as specified in Subclause 10.7, "Determination of equivalent XML values".

4.3.4 XML comparison and assignment

A value of an XML type *S* is assignable to a site of an XML type *T* if exactly one of the following is true.

— *T* is either XML(DOCUMENT(UNTYPED)) or XML(CONTENT(UNTYPED)) and *S* is either XML(DOCUMENT(UNTYPED)) or XML(CONTENT(UNTYPED)).

— *T* is either XML(DOCUMENT(XMLSCHEMA)) or XML(CONTENT(XMLSCHEMA)) and *S* is either XML(DOCUMENT(XMLSCHEMA)) or XML(CONTENT(XMLSCHEMA)) such that the registered XML schema descriptors included in the XML type descriptors of both *T* and *S* identify identical XML schemas, the XML namespace URI included in the XML type descriptor of *T*, if any, is identical to the XML namespace URI included in the XML type descriptor of *S*, as defined by XML Namespaces, and the XML NCName of the global element declaration schema component included in the XML type descriptor of *T*, if any, is identical to the XML NCName of the global element declaration schema component included in the XML type descriptor of *S*.

NOTE 5 — The notion of identical XML schemas is defined in Subclause 4.3.6, "Registered XML schemas".

— *T* is either XML(DOCUMENT(ANY)), XML(CONTENT(ANY)), or XML(SEQUENCE) and *S* is an XML type.

Operations that involve assignment of XML values use the keywords BY REF to indicate that the assignment preserves XQuery node identity, and BY VALUE to indicate that the assignment loses XQuery node identity.

XML values are not comparable.

4.3.5 Operations involving XML values

<XML document> is an operator that returns an XML value, given another XML value. The new XML value consists of an XQuery document node that is constructed according to the rules of the computed document constructor in XQuery 3.1: an XML Query Language.

<XML element> is an operator that returns an XML value, given an XML element name, an optional list of XML namespaces, an optional list of XML attributes, and an optional list of values as the content of the

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

new element. The value of <XML element content> can be any value that has a mapping to an XML value. The result, an XQuery element node, may optionally be placed as the sole child of an XQuery document node.

<XML forest> is an operator that returns an XML value, given an optional list of XML namespaces and a list of <forest element>s. An XQuery element node is produced from each <forest element>, using the column name or, if provided, the <forest element name> as the XML element name and the <forest element value> as the element content. The value of <forest element value> can be any value that has a mapping to an XML value. The result, an XQuery sequence, may optionally be placed in an XQuery document node.

<XML concatenation> is an operator that returns an XML value by concatenating a list of XML values, as defined in Subclause 6.13, "<XML concatenation>". The result, an XQuery sequence, may optionally be placed in an XQuery document node.

<XML comment> is an operator that returns an XML value, given a character string *CS*. The XML value consists of an XQuery comment node whose **content** property is *CS*, mapped to Unicode. This XQuery comment node may optionally be placed as the sole child of an XQuery document node.

<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.

<XML text> is an operator that returns an XML value, given a character string *CS*. The XML value consists of an XQuery text node whose **content** property is *CS*, mapped to Unicode. The XQuery text node may optionally be placed as the sole child of an XQuery document node.

<XML query> is an operator that evaluates an XQuery expression, which may be parameterized with any number of input parameters. The result, an XQuery sequence, may optionally be placed in an XQuery document node.

<XML table> is a kind of <derived table>, which may be used to query an XML value as a table. The result is specified by means of <XML table row pattern>, <XML table argument list>, and <XML table column definitions>. The result is computed in the following steps.

— The <XML table row pattern> is an XQuery expression that is evaluated, with <XML table argument list> as the (optional) input arguments. This produces an intermediate result, an XQuery sequence.

— Each XQuery item in the intermediate result is used to create a row of the final result.

— The columns of the result are specified by <XML table column definitions>, which is a collection of <XML table column definition>s modeled on a <table definition>. Thus, each result column has a <data type>, and, optionally, a <default clause>. Each <XML table column definition> also specifies, explicitly or implicitly, an <XML table column pattern> *XTCP*, which is an XQuery expression.

— To generate the value for a particular row and column of the final result, the column pattern (i.e., the XQuery expression *XTCP*) is evaluated using a particular XQuery item of the intermediate result as the input parameter, and then cast to the <data type> of the result column. If *XTCP* evaluates to the empty XQuery sequence, then the column's <default clause>, if any, determines the column's default value.

<XML character string serialization> is an operator that returns a character string, given an XML value, using the algorithm in XSLT and XQuery Serialization 3.1. Optionally, the input XML value may be checked to ensure that it is an XQuery document node whose **children** property has exactly one XQuery element node. Other options may be used to specify the version of XML text that is produced (i.e., either a textual XML 1.0 content or a textual XML 1.1 content); and whether an XML declaration is produced or not.

<XML binary string serialization> is an operator that returns a binary string, given an XML value, using the algorithm in XSLT and XQuery Serialization 3.1. <XML binary string serialization> supports the same

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

options as <XML character string serialization>, plus the ability to specify the encoding (character set) of the result.

<XML parse> is an operator that parses a character string or binary string value, i.e., converts XML text to a set of XML information items, according to the rules of XML Information Set, and then converts that set of XML information items into an XQuery document node, according to the rules of XQuery and XPath Data Model 3.1.

<XML validate> is an operator that validates an XML value according to a registered XML schema. Optionally, the registered XML schema against which to validate may be chosen on the basis of the contents of the data value, or it may be specified directly in the <XML validate>. In the latter case, additional options allow the validation to be confined to a particular namespace of the designated registered XML schema, or to a particular global element declaration schema component. Upon successful validation, <XML validate> returns a copy of the input XML value augmented with default values and type annotations. If validation fails, <XML validate> raises an exception.

<XML content predicate> is a predicate that determines if an XML value is an XQuery document node.

<XML document predicate> is a predicate that determines if an XML value is an XQuery document node whose **children** property contains exactly one XQuery element node, zero or more XQuery comment nodes, and zero or more XQuery processing instruction nodes.

<XML exists predicate> is a predicate that evaluates an XQuery expression and determines if the result is a null value, an empty XQuery sequence, or a non-empty XQuery sequence.

<XML valid predicate> is a predicate that determines if an XML value is valid according to a registered XML schema. Optionally, the registered XML schema to validate against may be chosen on the basis of the contents of the data value, or it may be specified directly in the <XML valid predicate>. In the latter case, additional options allow the validation to be confined to a particular namespace of the designated registered XML schema, or to a particular global element declaration schema component.

4.3.6 Registered XML schemas

A *registered XML schema* is an XML schema that has been made known to the SQL-server. The means by which an XML schema is registered is implementation-defined (IW200).

A global element declaration schema component of a registered XML schema is *non-deterministic* if it contains or references a wildcard schema component whose **{namespace constraint}** property is either **not** together with a namespace name, or **any**, and whose **{process contents}** property is either **strict** or **lax** (as defined in XML Schema Part 1: Structures section 3.10.1, "The Wildcard Schema Component").

NOTE 6 — The elements of an XML schema Document corresponding to such wildcard schema components are elements **`<xs:any>`** or **`<xs:anyAttribute>`** in which the **`namespace`** attribute is either missing or has the value "**`##any`**" or "**`##other`**", and the **`processContents`** attribute is either missing or has either the value "**`strict`**" or the value "**`lax`**".

An XML namespace *NS* contained in a registered XML schema is *non-deterministic* if *NS* contains a global element declaration schema component that is non-deterministic.

A registered XML schema is *non-deterministic* if it contains a non-deterministic XML namespace.

A registered XML schema is described by a registered XML schema descriptor. A registered XML schema descriptor includes the following.

— The target namespace URI of the registered XML schema.

— The schema location URI of the registered XML schema.

— The <registered XML schema name> of the registered XML schema.

— An indication of whether the registered XML schema is permanently registered.

— An indication of whether the registered XML schema is non-deterministic.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

— An unordered collection of the namespaces defined by the registered XML schema (the target namespace is one of these namespaces).

— For each namespace defined by the registered XML schema, an unordered collection of the global element declaration schema components in that namespace, with an indication for each global element declaration schema component whether that global element declaration schema component is non-deterministic.

NOTE 7 — Without Feature X161, "Advanced Information Schema for registered XML schemas", information whether an XML schema is deterministic, information about the collection of namespaces defined in that XML schema, and, for each such namespace information about the global element declaration schema components in that namespace, is not available in the XML_SCHEMAS, XML_SCHEMA_NAMESPACES, and XML_SCHEMA_ELEMENTS views.

A registered XML schema is identified by its <registered XML schema name>.

Two registered XML schemas are considered *identical* if both are identified by the same <registered XML schema name>.

Certain XML schemas, defined by either this document or by some normative reference, are always registered. These XML schemas are enumerated in Table 1, "Permanently registered XML schemas".

Table 1 — Permanently registered XML schemas

Common prefix (non-normative)	Target namespace URI[1]
`xs`	`http://www.w3.org/2001/XMLSchema`
`xsi`	`http://www.w3.org/2001/XMLSchema-instance`
`sqlxml`	`http://standards.iso.org/iso/9075/2003/sqlxml`

[1] The schema location URIs and <registered XML schema name>s of these XML schemas are implementation-defined (IV200).

NOTE 8 — The "common prefix" column in the preceding table indicates the prefix(es) commonly associated with the target namespaces of these XML schemas, but there is no requirement to refer to them by these prefixes.

If a <data type>, an <XML validate>, or an <XML valid predicate> that contains an <XML valid according to what> *XVACC* is contained in a <query expression> of a view, a check constraint, or an assertion, the <triggered action> of a trigger, or in an <SQL-invoked routine>, then the registered XML schema that is referenced by *XVACC* is determined at the time the view is created, the check constraint is defined, the assertion is created, the trigger is created, or the SQL-invoked routine is created. The same registered XML schema is referenced whenever the view is used, or the check constraint or assertion is evaluated, the trigger is executed, or the SQL-invoked routine is invoked.

4.4 Data conversions

This Subclause modifies Subclause 4.13, "Data conversions", in ISO/IEC 9075-2.

Insert before the 3rd paragraph: Data conversions between the predefined data types defined in ISO/IEC 9075-2 and the XML types can be specified by an <XML cast specification>.

A conversion from an XML type to a non-XML type is computed as follows:

1) XQuery document nodes are removed from the source value.

2) The result is converted to an XQuery sequence of XQuery atomic values ***AV*** using the XQuery function `fn:data()`.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

3) An XML schema data type ***XT*** is chosen to correspond to the target <data type>. For example, if the target <data type> is INTEGER then *XT* is **`xs:integer`**.

4) ***AV*** is converted to ***XT*** using the rules of XQuery 3.1: an XML Query Language, producing ***BV***.

5) ***BV*** is a value of an XML schema data type, which is either an XML schema primitive type, or derived from an XML schema primitive type. For example, ***BV*** might be a value of XML schema type **`xs:short`**, which is derived from **`xs:decimal`**. The value spaces of most XML schema primitive types are identified with an SQL value space. For example, the XML value space of **`xs:decimal`** is identified with the SQL value space of exact numeric values. If, say, ***BV*** has the value **`9075.14`** in the XML value space of **`xs:decimal`**, then ***BV*** is also regarded as being the value **`9075.14`** in the SQL value space of exact numeric values. These value space identifications are defined in Subclause 4.11.8, "Mapping XQuery atomic values to SQL values".

6) ***BV***, regarded now as a value of an SQL value space, is converted to the target <data type>, using a <cast specification>.

A conversion from a non-XML type to an XML type is computed as follows:

1) The source value is converted to a <character string literal> whose value is the result of applying the General Rules of Subclause 9.8, "Mapping values of SQL data types to values of XML schema data types".

2) The <character string literal> is parsed using <XML parse>, producing an XQuery document node ***D***.

3) If the target <data type> is XML(CONTENT(UNTYPED)) or XML(CONTENT(ANY)), then ***D*** is the result.

4) If the target <data type> is XML(SEQUENCE), then ***D*** is converted to an XQuery atomic value by evaluating an XQuery cast expression. The XQuery type to convert to is chosen based on the declared type of the source value. For example, if the declared type of the source is exact numeric, then the XQuery type of the result is **`xs:decimal`**.

4.5 Data analysis operations

This Subclause modifies Subclause 4.18, "Data analysis operations", in ISO/IEC 9075-2.

4.5.1 Aggregate functions

This Subclause modifies Subclause 4.18.4, "Aggregate functions", in ISO/IEC 9075-2.

Insert into the 7th paragraph, after the last list item:

— If XMLAGG is specified, then an XML value formed from the <XML value expression> evaluated for each row that qualifies.

4.6 SQL-invoked routines

This Subclause modifies Subclause 4.35, "SQL-invoked routines", in ISO/IEC 9075-2.

4.6.1 Routine descriptors

This Subclause modifies Subclause 4.35.5, "Routine descriptors", in ISO/IEC 9075-2.

Insert into the 1st paragraph, in the 6th list item, after the last list item:

— For every SQL parameter whose declared type is an XML type or a distinct type whose source type is an XML type, an indication of the <XML passing mechanism>.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

Insert into the 1st paragraph, in the 11th list item, after the last list item:

— If the <returns clause> contains an <XML passing mechanism>, then that <XML passing mechanism>.

Insert into the 1st paragraph, in the 15th list item, after the last list item:

— For every SQL parameter that has an associated string type, the character string type descriptor of the associated string type.

— For every SQL parameter that has an associated XML option, an indication of the associated XML option.

4.7 SQL-statements

This Subclause modifies Subclause 4.41, "SQL-statements", in ISO/IEC 9075-2.

4.7.1 SQL-statements classified by function

This Subclause modifies Subclause 4.41.2, "SQL-statements classified by function", in ISO/IEC 9075-2.

4.7.1.1 SQL-session statements

This Subclause modifies Subclause 4.41.2.7, "SQL-session statements", in ISO/IEC 9075-2.

Insert into the 1st paragraph, after the last list item:

— <set XML option statement>

4.8 Basic security model

This Subclause modifies Subclause 4.42, "Basic security model", in ISO/IEC 9075-2.

4.8.1 Privileges

This Subclause modifies Subclause 4.42.2, "Privileges", in ISO/IEC 9075-2.

Insert into the 1st paragraph, after the last list item:

— registered XML schema

Insert into the 11th paragraph, after the last list item:

— registered XML schema

Insert after the 11th paragraph: USAGE privileges on registered XML schemas are granted or revoked by implementation-defined (IW201) means.

4.9 SQL-sessions

This Subclause modifies Subclause 4.45, "SQL-sessions", in ISO/IEC 9075-2.

4.9.1 SQL-session properties

This Subclause modifies Subclause 4.45.3, "SQL-session properties", in ISO/IEC 9075-2.

Insert after the 6th paragraph: An SQL-session has an XML option that is used to identify the <document or content> option needed in the implicit invocation of XMLSERIALIZE and XMLPARSE operators during the execution of <preparable statement>s that are prepared in the current SQL-session by either an <execute immediate statement> or a <prepare statement>. The XML option is initially set to an imple-

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

mentation-defined (ID200) value, but can subsequently be changed by the successful execution of a <set XML option statement>.

Insert into the 13th paragraph, after the last list item:

— The current XML option.

4.10 XML namespaces

This document references certain XML namespaces that are defined by the World-Wide Web Consortium or by this document. Each XML namespace is referenced using an XML namespace prefix. The XML namespace prefixes and their definitions are shown in Table 2, "XML namespace prefixes and their URIs".

Table 2 — XML namespace prefixes and their URIs

XML namespace prefix	XML namespace URI
`xs`	`http://www.w3.org/2001/XMLSchema`
`xsi`	`http://www.w3.org/2001/XMLSchema-instance`
`sqlxml`	`http://standards.iso.org/iso/9075/2003/sqlxml`

A conforming SQL/XML implementation is not required to use the XML namespace prefixes **`xs`**, **`xsi`**, or **`sqlxml`** to reference these XML namespaces, but whatever XML namespace prefix it uses shall be associated with the proper URI.

The XML namespace identified by the XML namespace prefix "**`sqlxml`**" is normatively defined in Clause 23, "SQL/XML XML schema".

A resource containing the XML schema definition of the XML namespace identified by the XML namespace prefix "**`sqlxml`**" (that is, a file containing an XML schema document) has been made available on the World Wide Web. The URI of that resource is:

`http://standards.iso.org/iso/9075/2003/sqlxml.xsd`

It is intended that the contents of that file be identical to the contents of Clause 23, "SQL/XML XML schema". This file has been created for the convenience of the implementors of this document.

4.11 Overview of mappings

4.11.1 Introduction to mappings

This document defines mappings from SQL to XML, and from XML to SQL. The mappings from SQL to XML include the following,

— Mapping SQL character sets to Unicode.

— Mapping SQL <identifier>s to XML names.

— Mapping SQL data types (as used in SQL-schemas to define SQL-schema objects such as columns) to XML schema data types.

— Mapping values of SQL data types to values of XML schema data types.

— Mapping an SQL table to an XML document and an XML schema document.

— Mapping an SQL schema to an XML document and an XML schema document.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

— Mapping an SQL catalog to an XML document and an XML schema document.

The mappings from XML to SQL include the following.

— Mapping Unicode to SQL character sets.

— Mapping XML names to SQL <identifier>s.

4.11.2 Mapping SQL character sets to Unicode

For each character set *SQLCS* in the SQL-environment, there shall be a mapping *CSM* of strings of *SQLCS* to strings of Unicode, as defined in The Unicode® Standard. In this document, "Unicode" refers to the character repertoire named "UCS". The mapping *CSM* is called *homomorphic* if for each non-negative integer *N*, there exists a non-negative integer *M* such that all strings of length *N* in *SQLCS* are mapped to strings of length *M* in Unicode. *CSM* is implementation-defined (IA091). However, if any Unicode code point is mapped to a character that is not a valid XML character, an exception condition is raised.

NOTE 9 — The entity references **`<`**, **`&`**, **`>`**, **`'`**, and **`"`**, as well as character references, as defined by XML are regarded as each representing a single character in XML, and do not pose an obstacle to defining homomorphic mappings.

4.11.3 Mapping Unicode to SQL character sets

For each character set *SQLCS* in the SQL-environment, there shall be an implementation-defined (IA092) mapping *CSM* of strings of Unicode to strings of *SQLCS*.

4.11.4 Mapping SQL <identifier>s to XML

Since not every SQL <identifier> is an acceptable XML name, it is necessary to define a mapping of SQL <identifier>s to XML names. This mapping is defined in Subclause 9.1, "Mapping SQL <identifier>s to XML names". The basic idea of this mapping is that characters that are not valid in XML names are converted to a sequence of hexadecimal digits derived from the Unicode encoding of the character, bracketed by an introductory underscore and lower-case `x` and a trailing underscore.

There are two variants of the mapping, known as *partially escaped* and *fully escaped*. The two differences are in the treatment of non-initial <colon> and the treatment of an <identifier> beginning with the letters `xml` in any combination of upper or lower-case. The fully escaped variant maps a non-initial <colon> to **`_x003A_`**, whereas the partially escaped variant maps non-initial <colon> to **`:`**. Also, the fully escaped variant maps initial `xml` and `XML` to **`_x0078_ml`** and **`_x0058_ML`**, respectively, whereas the partially escaped does not.

NOTE 10 — This document specifies no syntax for invoking the partially escaped mapping specified in Subclause 9.1, "Mapping SQL <identifier>s to XML names". This specification is intended to be used by applications and referenced by other standards.

4.11.5 Mapping XML names to SQL

A single algorithm suffices to reverse both the partially escaped and the fully escaped variants of the mapping of SQL <identifier>s to XML names. This algorithm is found in Subclause 9.3, "Mapping XML names to SQL <identifier>s". The basic idea is to scan the XML name from left to right, looking for escape sequences of the form **`_x`*NNNN*`_`** or **`_x`*NNNNNN*`_`** where ***N*** denotes a hexadecimal digit. Such sequences are converted to the character of SQL_TEXT that corresponds to the Unicode code point U+0000*NNNN* or U+00*NNNNNN*, respectively.

NOTE 11 — This document specifies no syntax for invoking the mapping specified in Subclause 9.3, "Mapping XML names to SQL <identifier>s". This specification is intended to be used by applications and referenced by other standards. It is the responsibility of any such application or other standard to ensure that the correct number of arguments as well as a valid value for each argument are supplied for this mapping.

NOTE 12 — The sequence of mappings from SQL <identifier> to XML name (using either the fully escaped mapping or the partially escaped mapping) to SQL <identifier> restores the original SQL <identifier> (assuming that every character in the source SQL-implementation's SQL <identifier> is a character of SQL_TEXT in the target SQL-implementation). However,

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

the sequence of mappings from XML name to SQL <identifier> to XML Name does not necessarily restore the XML name. Also, more than one XML name can be mapped to the same SQL <identifier>.

4.11.6 Mapping SQL data types to XML

For each SQL type or domain that is not unmappable, there is a corresponding XML schema type. The mapping is fully specified in Subclause 9.5, "Mapping SQL data types to XML schema data types". The following is a conceptual description of this mapping.

In general, each SQL predefined type, distinct type, or domain *SQLT* is mapped to the XML schema type ***XMLT*** that is the closest analog to *SQLT*. Since the value space of ***XMLT*** is frequently richer than the set of values that can be represented by *SQLT*, facets are used to restrict ***XMLT*** in order to capture the restrictions on *SQLT* as much as possible.

In addition, many of the distinctions in the SQL type system (for example, CHARACTER VARYING *versus* CHARACTER LARGE OBJECT) have no corresponding distinction in the XML schema type system. In order to represent these distinctions, XML schema annotations are defined. The content of the annotations is defined by this document; however, whether such annotations are actually generated is implementation-defined (IA100). Elements from the XML namespace identified by the XML namespace prefix "**sqlxml**" are used to populate these annotations.

The SQL character string types are mapped to the XML schema type **xs:string**. For the SQL type CHARACTER, if the mapping of the SQL character set to Unicode is homomorphic, then fixed length strings are mapped to fixed length strings, and the facet **xs:length** is used. Otherwise (i.e., CHARACTER when the mapping is not homomorphic, as well as CHARACTER VARYING and CHARACTER LARGE OBJECT), the facet **xs:maxLength** is used. Annotations optionally indicate the precise SQL type (CHARACTER, CHARACTER VARYING, or CHARACTER LARGE OBJECT), the length or maximum length of the SQL type, the character set, and the default collation.

The SQL binary string types are mapped to either the XML schema type **xs:hexBinary** or the XML schema type **xs:base64Binary**. The **xs:maxLength** facet is set to the maximum length of the binary string in octets. Annotations optionally indicate the SQL type (BINARY, BINARY VARYING, BINARY LARGE OBJECT) and the maximum length in octets. For <XML element> and <XML forest>, the choice of whether to map to **xs:hexBinary** or **xs:base64Binary** is governed by the innermost <XML binary encoding> whose scope includes the <XML element> or <XML forest>; the default is implementation-defined (ID201). When mapping an SQL table, schema or catalog to XML, the choice is governed by a parameter, as specified in Subclause 9.11, "Mapping an SQL table to XML and an XML schema document", Subclause 9.14, "Mapping an SQL schema to an XML document and an XML schema document", and Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML schema document".

The exact numeric SQL types NUMERIC and DECIMAL are mapped to the XML schema type **xs:decimal** using the facets **xs:precision** and **xs:scale**. It is implementation-defined (IA093) whether the SQL types INTEGER, SMALLINT, and BIGINT are mapped to the XML schema type **xs:integer** using the facets **xs:maxInclusive** and **xs:minInclusive** or to the closest XML schema type that is a subtype of **xs:integer**, using the facets **xs:maxInclusive** and **xs:minInclusive** if the range of the SQL type does not exactly match the range of the XML schema type to which it is mapped. Annotations optionally indicate the SQL type (NUMERIC, DECIMAL, INTEGER, SMALLINT, or BIGINT), precision of NUMERIC, user-specified precision of DECIMAL (which may be less than the actual precision), and scale of NUMERIC and DECIMAL.

The approximate numeric SQL types are mapped to either the XML schema type **xs:float**, if the binary precision is less than or equal to 24 binary digits (bits) and the range of the binary exponent lies between -149 and 104, inclusive; otherwise, the XML schema type **xs:double** is used. Annotations optionally indicate the SQL type (REAL, DOUBLE PRECISION, or FLOAT), the binary precision, the minimum and maximum values of the range of binary exponents, and, for FLOAT, the user-specified binary precision (which may be less than the actual precision).

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

The SQL type BOOLEAN is mapped to the XML schema type **`xs:boolean`**. Optionally, an annotation indicates the SQL type (BOOLEAN).

The SQL type DATE is mapped to the XML schema type **`xs:date`**. The **`xs:pattern`** facet is used to exclude the possibility of a time zone displacement. Optionally, an annotation indicates the SQL type, DATE.

The SQL types TIME WITHOUT TIME ZONE and TIME WITH TIME ZONE are mapped to the XML schema type **`xs:time`**. The **`xs:pattern`** facet is used to exclude the possibility of a time zone displacement, in the case of TIME WITHOUT TIME ZONE, or to require a time zone displacement, in the case of TIME WITH TIME ZONE. The **`xs:pattern`** facet also reflects the fractional seconds precision of the SQL type. Annotations optionally indicate the SQL type (TIME or TIME WITH TIME ZONE) and the fractional seconds precision.

The SQL types TIMESTAMP WITHOUT TIME ZONE and TIMESTAMP WITH TIME ZONE are mapped to the XML schema type **`xs:dateTime`**. The **`xs:pattern`** facet is used to exclude the possibility of a time zone displacement, in the case of TIMESTAMP WITHOUT TIME ZONE, or to require a time zone displacement, in the case of TIMESTAMP WITH TIME ZONE. The **`xs:pattern`** facet also reflects the fractional seconds precision of the SQL type. Annotations optionally indicate the SQL type (TIMESTAMP or TIMESTAMP WITH TIME ZONE) and the fractional seconds precision.

The SQL interval types are mapped to the XML schema types **`xs:yearMonthDuration`** and **`xs:dayTimeDuration`**. The **`xs:pattern`** facet is used to require precisely the year, month, day, hour, minute and second fields indicated by the SQL type. The **`xs:pattern`** facet also reflects the leading field precision and the fractional seconds precision (when applicable). Annotations optionally indicate the SQL type, leading field precision and (when applicable) the fractional seconds precision.

An SQL row type is mapped to an XML schema complex type that consists of one element for each field of the SQL row type. For each field *F* of the SQL row type, the name of the corresponding XML element is obtained by mapping the field name of *F* using the fully escaped variant, and the XML schema type of the element is obtained by mapping the field type of *F*.

An SQL domain is mapped to XML by mapping the domain's data type to XML and then optionally applying to the generated XML schema type an annotation that identifies the name of the domain.

An SQL distinct type is mapped to an XML schema simple type by mapping the source type of the distinct type. Optionally, an annotation specifying the name of the distinct type is applied to the generated XML schema type.

An SQL collection type is mapped to an XML schema complex type having a single XML element named **`element`** whose XML schema type is obtained by mapping the element type of the SQL collection type. This XML element is defined using **`minOccurs="0"`**. For an SQL array type, **`maxOccurs`** is the maximum cardinality of the array, whereas for an SQL multiset type, **`maxOccurs="unbounded"`**.

An SQL XML type is mapped to an XML schema complex type that allows mixed content and an unvalidated **`any`** section. Optionally, an annotation indicates the SQL type, XML.

4.11.7 Mapping values of SQL data types to XML

For each SQL type or domain *SQLT*, with the exception of structured types and reference types, there is also a mapping of values of type *SQLT* to the value space of the corresponding XML schema type. The mappings of values are largely determined by the data type mappings. The precise rules for non-null values are found in Subclause 9.8, "Mapping values of SQL data types to values of XML schema data types". The mappings for values of predefined types are designed to exploit <cast specification> as much as possible. As for null values, there is generally a choice of whether to represent nulls using absence or **`xsi:nil="true"`**. However, for elements of a collection type, null values are always represented by **`xsi:nil="true"`**.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

4.11.8 Mapping XQuery atomic values to SQL values

As defined in XQuery and XPath Data Model 3.1, an XQuery atomic type is either an XML schema primitive type, or derived from an XML schema primitive type by restriction (and not by union or list).

Let ***AV*** be an XQuery atomic value. Let ***AT*** be the XQuery atomic type of ***AV***. Let ***PT*** be given by

Case:

— If ***AT*** is an XML schema primitive type, then ***AT***.

— If ***AT*** is **`xs:yearMonthDuration`**, or derived from **`xs:yearMonthDuration`**, then **`xs:yearMonthDuration`**.

— If ***AT*** is **`xs:dayTimeDuration`**, or derived from **`xs:dayTimeDuration`**, then **`xs:dayTimeDuration`**.

— Otherwise, the XML schema primitive type from which ***AT*** is derived.

This document (notably, in Subclause 6.7, "<XML cast specification>") regards ***AV*** as being a value belonging to some category of SQL predefined type, as follows.

Case:

— If ***PT*** is **`xs:string`**, then ***AV*** is regarded as being a character string whose character repertoire is Unicode.

— If ***PT*** is **`xs:hexBinary`** or **`xs:base64Binary`**, then ***AV*** is regarded as being a binary string.

— If ***PT*** is **`xs:decimal`**, then ***AV*** is regarded as being an exact numeric value.

— If ***PT*** is **`xs:float`** or **`xs:double`**, then ***AV*** is regarded as being an approximate numeric value.

— If ***PT*** is **`xs:time`** and the XQuery datetime time zone component of ***AV*** is an empty XQuery sequence, then the XQuery datetime normalized value of ***AV*** is regarded as being a value of type TIME WITHOUT TIME ZONE.

— If ***PT*** is **`xs:time`** and the XQuery datetime time zone component of ***AV*** is not an empty XQuery sequence, then ***AV*** is regarded as being a value of type TIME WITH TIME ZONE, in which the XQuery datetime time zone component of ***AV*** is the time zone component, and the XQuery datetime normalized value is the UTC component.

— If ***PT*** is **`xs:dateTime`**, the XQuery datetime normalized value ***XDNV*** of ***AV*** is positive, and the XQuery datetime time zone component of ***AV*** is an empty XQuery sequence, then ***XDNV*** is regarded as being a value of type TIMESTAMP WITHOUT TIME ZONE. If ***XDNV*** would have a SECOND field greater than 59 in a minute of UTC that has exactly 59 seconds, then it is implementation-defined (IA055) whether an implementation-defined (IV125) value of type TIMESTAMP WITHOUT TIME ZONE is identified with ***XDNV***, or whether an exception condition is raised: *data exception — datetime field overflow (22008)*.

— If ***PT*** is **`xs:dateTime`**, the XQuery datetime normalized value of ***AV*** is positive, and the XQuery datetime time zone component of ***AV*** is not an empty XQuery sequence, then ***AV*** is regarded as being a value of type TIMESTAMP WITH TIME ZONE, in which the XQuery datetime time zone component of ***AV*** is the time zone component, and the XQuery datetime normalized value is the UTC component. If AV denotes a minute of UTC that has exactly 59 seconds and the SECOND field of ***AV*** is greater than 59, then it is implementation-defined (IA055) whether an implementation-defined (IV125) value of type TIMESTAMP WITHOUT TIME ZONE is identified with ***AV***, or whether an exception condition is raised: *data exception — datetime field overflow (22008)*.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

— If ***PT*** is **`xs:date`**, the XQuery datetime normalized value of ***AV*** is positive, and the XQuery datetime time zone component of ***AV*** is an empty XQuery sequence, then the XQuery datetime normalized value of ***AV*** is regarded as being a value of type DATE.

— If ***PT*** is **`xs:yearMonthDuration`**, then ***AV*** is regarded as being a year-month interval.

— If ***PT*** is **`xs:dayTimeDuration`**, then ***AV*** is regarded as being a day-time interval.

— If ***PT*** is **`xs:boolean`**, then ***AV*** is regarded as being a value of type BOOLEAN.

4.11.9 Visibility of columns, tables, and schemas in mappings from SQL to XML

An *XML unmappable data type* is a data type that is one of the following: a JSON type, a structured type, a reference type, XML(SEQUENCE), or a type defined in some part of the ISO/IEC 9075 series other than ISO/IEC 9075-2 and this document. An *XML unmappable column* is a column that has a declared type that is one of the XML unmappable data types or has a declared type that is based on an XML unmappable data type.

A column *C* of table *T* is a *visible column* of *T* for authorization identifier *U* if the applicable privileges for *U* include the SELECT privilege on *C* and, if the declared type of *C* is a distinct type, the applicable privileges for *U* include EXECUTE on the user-defined cast function identified by the Syntax Rules of Subclause 6.6, "<cast specification>". A column *C* of table *T* is an *XML visible column* of *T* for authorization identifier *U* if *C* is a visible column of *T* for authorization identifier *U* and the declared type of *C* is not an XML unmappable data type.

A table *T* of schema *S* is a *visible table* of *S* for authorization identifier *U* if *T* is either a base table or a viewed table that contains a column *C* that is a visible column for *U*. A table *T* of schema *S* is an *XML visible table* of *S* for authorization identifier *U* if *T* is either a base table or a viewed table that contains a column *C* that is an XML visible column for *U*.

A schema *S* of catalog *C* is a *visible schema* of *C* for authorization identifier *U* if *S* contains a table *T* that is a visible table for *U*. A schema *S* of catalog *C* is an *XML visible schema* of *C* for authorization identifier *U* if *S* contains a table *T* that is an XML visible table for *U*.

4.11.10 Mapping an SQL table to XML

Subclause 9.11, "Mapping an SQL table to XML and an XML schema document", defines a mapping of an SQL table to one or both of two documents: an XML schema document that describes the structure of the mapped XML and either an XML document or a sequence of XML elements. Only base tables and viewed tables may be the source of this mapping.

NOTE 13 — This document specifies no syntax for invoking the mapping specified in Subclause 9.11, "Mapping an SQL table to XML and an XML schema document". This specification is intended to be used by applications and referenced by other standards. It is the responsibility of any such application or other standard to ensure that the correct number of arguments as well as a valid value for each argument are supplied for this mapping.

Only the XML visible columns of this table for the user that invokes this mapping will be represented in the generated XML.

This mapping allows the invoker to specify the following.

— Whether to map the table to a sequence of XML elements where the name of each top-level element is derived from the table name and represents a row in the table, or to map the table to an XML document with a single root element whose name is derived from the table name and to map each row to an element named **`<row>`**.

— The XML target namespace URI of the XML schema and data to be mapped (if the XML target namespace URI is specified as a zero-length string, then no XML namespace is added).

— Whether to map null values to absent elements, or whether to map them to elements that are marked with **`xsi:nil="true"`**.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

— Whether to map the table into XML data, to map the table into an XML schema document, or both.

Some of the XML schema type definitions and element declarations may contain annotations to represent SQL metadata that is not directly relevant to XML. It is implementation-defined (IA100) whether these annotations are generated.

The rules of Subclause 9.11, "Mapping an SQL table to XML and an XML schema document", are supported by the rules of Subclause 9.9, "Mapping an SQL table to XML schema data types", and Subclause 9.10, "Mapping an SQL table to an XML element or a sequence of XML elements".

4.11.11 Mapping an SQL schema to XML

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML schema document", defines a mapping between the tables of an SQL-schema and either or both of two documents: an XML document that represents the data in these tables, and an XML schema document that describes the first document.

NOTE 14 — This document specifies no syntax for invoking the mapping specified in Subclause 9.14, "Mapping an SQL schema to an XML document and an XML schema document". This specification is intended to be used by applications and referenced by other standards. It is the responsibility of any such application or other standard to ensure that the correct number of arguments as well as a valid value for each argument are supplied for this mapping.

Only the XML visible tables of the schema for the user that invokes this mapping will be represented in these two XML documents. Only the XML visible columns of these tables for the user that invokes this mapping will be represented in these two XML documents.

This mapping allows the invoker to specify the following.

— Whether to map each table to a sequence of XML elements where the name of each top-level element is derived from the table name and represents a row in the table, or to map each table to a single XML element whose name is derived from the table name and to map each row to an element named **`<row>`**.

— The XML target namespace URI of the XML schema and data to be mapped (if the XML target namespace URI is specified as a zero-length string, then no XML namespace is added).

— Whether to map null values to absent elements, or whether to map them to elements that are marked with **`xsi:nil="true"`**.

— Whether to map the schema into XML data, to map the schema into an XML schema document, or both.

Some of the XML schema type definitions and element declarations may contain annotations to represent SQL metadata that is not directly relevant to XML. It is implementation-defined (IA100) whether these annotations are generated.

The SQL-schema mapping assumes an implementation-dependent (UA031) *repeatable ordering* when iterating over the XML visible tables in the SQL-schema. This allows generating the correct alignment of the table data with the element declarations in situations where the generated XML schema uses a sequence content model instead of an all group content model.

The rules of Subclause 9.14, "Mapping an SQL schema to an XML document and an XML schema document", are supported by the rules of Subclause 9.12, "Mapping an SQL schema to XML schema data types", and Subclause 9.13, "Mapping an SQL schema to an XML element".

4.11.12 Mapping an SQL catalog to XML

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML schema document", defines a mapping between the tables of an SQL catalog and either or both of two documents: an XML document that represents the data in these tables, and an XML schema document that describes the first document.

NOTE 15 — This document specifies no syntax for invoking the mapping specified in Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML schema document". This specification is intended to be used by applications and

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

referenced by other standards. It is the responsibility of any such application or other standard to ensure that the correct number of arguments as well as a valid value for each argument are supplied for this mapping.

Only the XML visible schemas of this catalog for the user that invokes this mapping will be represented in these two XML documents. Only the XML visible tables of these schemas for the user that invokes this mapping will be represented in these two XML documents. Only the XML visible columns of these tables for the user that invokes this mapping will be represented in these two XML documents.

This mapping allows the user that invokes this mapping to specify the following.

— Whether to map each table to a sequence of XML elements where the name of each top-level element is derived from the table name and represents a row in the table, or to map each table to a single XML element whose name is derived from the table name and each row is mapped to an element names **`<row>`**.

— The XML target namespace URI of the XML schema and data to be mapped (if the XML target namespace URI is specified as a zero-length string, then no XML namespace is added).

— Whether to map null values to absent elements, or whether to map them to elements that are marked with **`xsi:nil="true"`**.

— Whether to map the catalog into XML data, to map the catalog into an XML schema document, or both.

Some of the XML schema type definitions and element declarations may contain annotations to represent SQL metadata that is not directly relevant to XML. It is implementation-defined (IA100) whether these annotations are generated.

The rules of Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML schema document", are supported by the rules of Subclause 9.15, "Mapping an SQL catalog to XML schema data types", and Subclause 9.16, "Mapping an SQL catalog to an XML element".

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

5 Lexical elements

This Clause modifies Clause 5, "Lexical elements", in ISO/IEC 9075-2.

5.1 <token> and <separator>

This Subclause modifies Subclause 5.2, "<token> and <separator>", in ISO/IEC 9075-2.

Function

Specify lexical units (tokens and separators) that participate in SQL language.

Format

```
<reserved word> ::=
    !! All alternatives from ISO/IEC 9075-2

      | XML | XMLAGG | XMLATTRIBUTES | XMLBINARY | XMLCAST | XMLCOMMENT | XMLCONCAT
      | XMLDOCUMENT | XMLELEMENT | XMLEXISTS | XMLFOREST | XMLITERATE | XMLNAMESPACES
      | XMLPARSE | XMLPI | XMLQUERY | XMLSERIALIZE | XMLTABLE | XMLTEXT | XMLVALIDATE

<non-reserved word> ::=
    !! All alternatives from ISO/IEC 9075-2

  | ACCORDING

  | BASE64 | BOM

  | COLUMNS | CONTENT

  | DOCUMENT

  | ENCODING

  | HEX

  | ID | INDENT

  | LOCATION

  | NAMESPACE | NIL

  | PATH | PRESERVE

  | SEQUENCE | STRIP

  | UNTYPED | URI

  | VALID | VERSION

  | WHITESPACE

  | XMLDECLARATION | XMLSCHEMA
```


© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

Syntax Rules

No additional Syntax Rules.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

Conformance Rules

No additional Conformance Rules.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

5.2 <literal>

This Subclause modifies Subclause 5.3, "<literal>", in ISO/IEC 9075-2.

Function

Specify a non-null value.

Format

```
<upper-case hexit> ::=
    <digit> | A | B | C | D | E | F
```

Syntax Rules

No additional Syntax Rules.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

Conformance Rules

No additional Conformance Rules.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

5.3 Names and identifiers

This Subclause modifies Subclause 5.4, "Names and identifiers", in ISO/IEC 9075-2.

Function

Specify names.

Format

```
<registered XML schema name> ::=
  <schema qualified name>
```

Syntax Rules

No additional Syntax Rules.

Access Rules

No additional Access Rules.

General Rules

1) Insert after the last GR: A <registered XML schema name> identifies a registered XML schema.

Conformance Rules

No additional Conformance Rules.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

6 Scalar expressions

This Clause modifies Clause 6, "Scalar expressions", in ISO/IEC 9075-2.

6.1 <data type>

This Subclause modifies Subclause 6.1, "<data type>", in ISO/IEC 9075-2.

Function

Specify a data type.

Format

```
<predefined type> ::=
    !! All alternatives from ISO/IEC 9075-2
  | <XML type>

<XML type> ::=
  XML [ <left paren> <XML type modifier> <right paren> ]

<XML type modifier> ::=
  <primary XML type modifier>
      [ <left paren> <secondary XML type modifier> <right paren> ]

<primary XML type modifier> ::=
    DOCUMENT
  | CONTENT
  | SEQUENCE

<secondary XML type modifier> ::=
    ANY
  | UNTYPED
  | XMLSCHEMA <XML valid according to what> [ <XML valid element clause> ]
```

Syntax Rules

1) Insert after the last SR: XML specifies an XML data type.

2) Insert after the last SR: If an <XML type> does not specify <XML type modifier>, then it is implementation-defined (ID202) whether SEQUENCE, CONTENT(ANY), or CONTENT(UNTYPED) is implicit.

3) Insert after the last SR: Case:

   a) If <primary XML type modifier> specifies SEQUENCE, then <secondary XML type modifier> shall not be specified.

   b) Otherwise, if <secondary XML type modifier> is not specified, then it is implementation-defined (ID203) whether UNTYPED or ANY is implicit.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

4) Insert after the last SR: If <secondary XML type modifier> specifies XMLSCHEMA, then let *RXS* be the indicated registered XML schema, let *ENSURI* be the indicated XML namespace, if any, and let ***GEDSC*** be the indicated global element declaration schema component, if any.

NOTE 16 — Indicated registered XML schema, indicated XML namespace, and indicated global element declaration schema component are defined in Subclause 11.5, "<XML valid according to clause>".

Access Rules

No additional Access Rules.

General Rules

1) Insert after GR 24): If <data type> is an <XML type>, then an XML type descriptor is created, including the following:

   a) The name of the data type (XML).

   b) The primary XML type modifier described by the <primary XML type modifier> (DOCUMENT, CONTENT, or SEQUENCE).

   c) The secondary XML type modifier described by the <secondary XML type modifier> (UNTYPED, ANY, or XMLSCHEMA), if any.

   d) The registered XML schema descriptor of the indicated registered XML schema, if any.

   e) The XML namespace URI of the indicated XML namespace, if any.

   f) The XML NCName of the indicated global element declaration schema component, if any.

Conformance Rules

1) Insert after the last CR: Without Feature X010, "XML type", conforming SQL language shall not contain an <XML type>.

2) Insert after the last CR: Without Feature X011, "Arrays of XML type", conforming SQL language shall not contain an <array type> that is based on a <data type> that is either an XML type or a distinct type whose source type is an XML type.

3) Insert after the last CR: Without Feature X012, "Multisets of XML type", conforming SQL language shall not contain a <multiset type> that is based on a <data type> that is either an XML type or a distinct type whose source type is an XML type.

4) Insert after the last CR: Without Feature X181, "XML(DOCUMENT(UNTYPED)) type", conforming SQL language shall not contain an <XML type> whose <primary XML type modifier> is DOCUMENT and <secondary XML type modifier> is UNTYPED.

5) Insert after the last CR: Without Feature X182, "XML(DOCUMENT(ANY)) type", conforming SQL language shall not contain an <XML type> whose <primary XML type modifier> is DOCUMENT and <secondary XML type modifier> is ANY.

6) Insert after the last CR: Without Feature X231, "XML(CONTENT(UNTYPED)) type", conforming SQL language shall not contain an <XML type> whose <primary XML type modifier> is CONTENT and <secondary XML type modifier> is UNTYPED.

7) Insert after the last CR: Without Feature X232, "XML(CONTENT(ANY)) type", conforming SQL language shall not contain an <XML type> whose <primary XML type modifier> is CONTENT and <secondary XML type modifier> is ANY.

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 09075-14-2023

8) Insert after the last CR: Without Feature X191, "XML(DOCUMENT(XMLSCHEMA)) type", conforming SQL language shall not contain an <XML type> whose <primary XML type modifier> is DOCUMENT and <secondary XML type modifier> specifies XMLSCHEMA.

9) Insert after the last CR: Without Feature X192, "XML(CONTENT(XMLSCHEMA)) type", conforming SQL language shall not contain an <XML type> whose <primary XML type modifier> is CONTENT and <secondary XML type modifier> specifies XMLSCHEMA.

10) Insert after the last CR: Without Feature X260, "XML type: ELEMENT clause", conforming SQL language shall not contain an <XML type> that contains <XML valid element clause>.

11) Insert after the last CR: Without Feature X261, "XML type: NAMESPACE without ELEMENT clause", conforming SQL language shall not contain an <XML type> that contains an <XML valid element clause> that does not contain an <XML valid element name specification>.

12) Insert after the last CR: Without Feature X263, "XML type: NO NAMESPACE with ELEMENT clause", conforming SQL language shall not contain an <XML type> that contains an <XML valid element namespace specification> that contains NO NAMESPACE.

13) Insert after the last CR: Without Feature X264, "XML type: schema location", conforming SQL language shall not contain an <XML type> that contains <XML valid schema location>.

14) Insert after the last CR: Without Feature X190, "XML(SEQUENCE) type", conforming SQL language shall not contain an <XML type> whose <XML type modifier> is SEQUENCE.

© ISO/IEC 2023 – All rights reserved
Documents Document Sharing Hub
Full document details: ISO IEC 09075-14-2023