

# [MS-TSQLISO14]: SQL Server Transact-SQL ISO/IEC 9075-14 Standards Support Document

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
06/29/2012	1.0	New	Released new document.
07/16/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/26/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/11/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.

# Contents

<b>1 Introduction</b> .....	<b>8</b>
1.1 Glossary .....	8
1.2 References.....	8
1.2.1 Normative References.....	8
1.2.2 Informative References .....	9
1.3 Microsoft Implementations.....	9
1.4 Standards Support Requirements .....	9
1.5 Notation .....	11
<b>2 Standards Support Statements</b> .....	<b>12</b>
2.1 Normative Variations.....	12
2.1.1 Mandatory Features.....	12
2.1.1.1 [XML10/5] 2.8 Prolog and Document Type Declaration .....	12
2.1.1.2 [XML10/5] 2.10 White Space Handling .....	13
2.1.1.3 [XMLSCHEMA1/2] minOccurs and maxOccurs (throughout).....	14
2.1.1.4 [XMLSCHEMA1/2] 2.2.2.2 Element Substitution Group .....	14
2.1.1.5 [XMLSCHEMA1/2] 3.4.3 Constraints on XML Representations of Complex Type Definitions .....	15
2.1.1.6 [XMLSCHEMA1/2] 3.8.2 XML Representation of Model Group Schema Components.....	15
2.1.1.7 [XMLSCHEMA1/2] 3.10.2 XML Representation of Wildcard Schema Components.....	16
2.1.1.8 [XMLSCHEMA1/2] 3.11 Identity-constraint Definitions .....	16
2.1.1.9 [XMLSCHEMA1/2] 4.2.1 Assembling a schema for a single target namespace from multiple schema definition documents .....	17
2.1.1.10 [XMLSCHEMA2/2] 2.5.1.3 Union datatypes .....	17
2.1.1.11 [XMLSCHEMA2/2] 3.2.1 string .....	18
2.1.1.12 [XMLSCHEMA2/2] 3.2.3 decimal .....	18
2.1.1.13 [XMLSCHEMA2/2] 3.2.4 float.....	19
2.1.1.14 [XMLSCHEMA2/2] 3.2.5 double.....	20
2.1.1.15 [XMLSCHEMA2/2] 3.2.6 duration .....	20
2.1.1.16 [XMLSCHEMA2/2] 3.2.7 dateTime .....	21
2.1.1.17 [XMLSCHEMA2/2] 3.2.8 time.....	22
2.1.1.18 [XMLSCHEMA2/2] 3.2.9 date.....	23
2.1.1.19 [XMLSCHEMA2/2] 3.2.10 gYearMonth.....	24
2.1.1.20 [XMLSCHEMA2/2] 3.2.11 gYear .....	24
2.1.1.21 [XMLSCHEMA2/2] 3.2.12 gMonthDay .....	25
2.1.1.22 [XMLSCHEMA2/2] 3.2.13 gDay .....	25
2.1.1.23 [XMLSCHEMA2/2] 3.2.14 gMonth.....	26
2.1.1.24 [XMLSCHEMA2/2] 3.2.17 anyURI.....	26
2.1.1.25 [XMLSCHEMA2/2] 3.2.18 QName .....	26
2.1.1.26 [XMLSCHEMA2/2] 3.2.19 NOTATION .....	27
2.1.1.27 [XMLSCHEMA2/2] 3.3.6 Name.....	27
2.1.1.28 [XMLSCHEMA2/2] 3.3.8 ID.....	28
2.1.1.29 [XMLSCHEMA2/2] 3.3.9 IDREF .....	29
2.1.1.30 [XMLSCHEMA2/2] 3.3.10 IDREFS.....	29
2.1.1.31 [XMLSCHEMA2/2] 4.1.2 XML Representation of Simple Type Definition Schema Components .....	29
2.1.1.32 [XMLSCHEMA2/2] 4.3.1 length .....	30
2.1.1.33 [XMLSCHEMA2/2] 4.3.2 minLength .....	30

2.1.1.34	[XMLSCHEMA2/2] 4.3.3 maxLength .....	31
2.1.1.35	[XMLSCHEMA2/2] 4.3.4 pattern.....	31
2.1.2	Optional Features.....	32
2.1.2.1	[XML10/5] 2.1 Well-Formed XML Documents .....	32
2.1.2.2	[XMLSCHEMA1/2] 2.6.3 xsi:schemaLocation, xsi:noNamespaceSchemaLocation.....	33
2.1.2.3	[XMLSCHEMA1/2] 3.3.2 XML Representation of Element Declaration Schema Components.....	33
2.1.2.4	[XMLSCHEMA1/2] 4.2.2 Including modified component definitions .....	34
2.1.2.5	[ISO/IEC9075-14:2011] X010, XML type.....	35
2.1.2.6	[ISO/IEC9075-14:2011] X011, Arrays of XML type .....	35
2.1.2.7	[ISO/IEC9075-14:2011] X012, Multisets of XML type.....	35
2.1.2.8	[ISO/IEC9075-14:2011] X013, Distinct types of XML type .....	36
2.1.2.9	[ISO/IEC9075-14:2011] X014, Attributes of XML type .....	36
2.1.2.10	[ISO/IEC9075-14:2011] X015, Fields of XML type .....	36
2.1.2.11	[ISO/IEC9075-14:2011] X020, XMLConcat.....	37
2.1.2.12	[ISO/IEC9075-14:2011] X025, XMLCast .....	37
2.1.2.13	[ISO/IEC9075-14:2011] X026, XMLCast: SQL cast semantics.....	37
2.1.2.14	[ISO/IEC9075-14:2011] X030, XMLDocument.....	38
2.1.2.15	[ISO/IEC9075-14:2011] X031, XMLElement.....	38
2.1.2.16	[ISO/IEC9075-14:2011] X032, XMLForest .....	39
2.1.2.17	[ISO/IEC9075-14:2011] X034, XMLAgg .....	39
2.1.2.18	[ISO/IEC9075-14:2011] X035, XMLAgg: ORDER BY option .....	39
2.1.2.19	[ISO/IEC9075-14:2011] X036, XMLComment.....	40
2.1.2.20	[ISO/IEC9075-14:2011] X037, XMLPI .....	40
2.1.2.21	[ISO/IEC9075-14:2011] X038, XMLText .....	40
2.1.2.22	[ISO/IEC9075-14:2011] X040, Basic table mapping.....	41
2.1.2.23	[ISO/IEC9075-14:2011] X041, Basic table mapping: null absent .....	41
2.1.2.24	[ISO/IEC9075-14:2011] X042, Basic table mapping: null as nil .....	41
2.1.2.25	[ISO/IEC9075-14:2011] X043, Basic table mapping: table as forest .....	42
2.1.2.26	[ISO/IEC9075-14:2011] X044, Basic table mapping: table as element .....	42
2.1.2.27	[ISO/IEC9075-14:2011] X045, Basic table mapping: with target namespace..	42
2.1.2.28	[ISO/IEC9075-14:2011] X046, Basic table mapping: data mapping .....	43
2.1.2.29	[ISO/IEC9075-14:2011] X047, Basic table mapping: metadata mapping .....	43
2.1.2.30	[ISO/IEC9075-14:2011] X048, Basic table mapping: base64 encoding of binary strings.....	44
2.1.2.31	[ISO/IEC9075-14:2011] X049, Basic table mapping: hex encoding of binary strings.....	44
2.1.2.32	[ISO/IEC9075-14:2011] X050, Advanced table mapping .....	45
2.1.2.33	[ISO/IEC9075-14:2011] X051, Advanced table mapping: null absent.....	45
2.1.2.34	[ISO/IEC9075-14:2011] X052, Advanced table mapping: null as nil .....	45
2.1.2.35	[ISO/IEC9075-14:2011] X053, Advanced table mapping: table as forest .....	46
2.1.2.36	[ISO/IEC9075-14:2011] X054, Advanced table mapping: table as element ....	46
2.1.2.37	[ISO/IEC9075-14:2011] X055, Advanced table mapping: with target namespace .....	47
2.1.2.38	[ISO/IEC9075-14:2011] X056, Advanced table mapping: data mapping .....	47
2.1.2.39	[ISO/IEC9075-14:2011] X057, Advanced table mapping: metadata mapping .....	48
2.1.2.40	[ISO/IEC9075-14:2011] X058, Advanced table mapping: base64 encoding of binary strings .....	49
2.1.2.41	[ISO/IEC9075-14:2011] X059, Advanced table mapping: hex encoding of binary strings.....	49

2.1.2.42	[ISO/IEC9075-14:2011] X060, XMLParse: Character string input and CONTENT option.....	50
2.1.2.43	[ISO/IEC9075-14:2011] X061, XMLParse: Character string input and DOCUMENT option.....	50
2.1.2.44	[ISO/IEC9075-14:2011] X065, XMLParse: BLOB input and CONTENT option ..	50
2.1.2.45	[ISO/IEC9075-14:2011] X066, XMLParse: BLOB input and DOCUMENT option .....	51
2.1.2.46	[ISO/IEC9075-14:2011] X068, XMLSerialize: BOM.....	51
2.1.2.47	[ISO/IEC9075-14:2011] X069, XMLSerialize: INDENT .....	52
2.1.2.48	[ISO/IEC9075-14:2011] X070, XMLSerialize: Character string serialization and CONTENT option.....	52
2.1.2.49	[ISO/IEC9075-14:2011] X071, XMLSerialize: Character string serialization and DOCUMENT option .....	52
2.1.2.50	[ISO/IEC9075-14:2011] X072, XMLSerialize: Character string serialization....	53
2.1.2.51	[ISO/IEC9075-14:2011] X073, XMLSerialize: BLOB serialization and CONTENT option.....	53
2.1.2.52	[ISO/IEC9075-14:2011] X074, XMLSerialize: BLOB serialization and DOCUMENT option .....	53
2.1.2.53	[ISO/IEC9075-14:2011] X075, XMLSerialize: BLOB serialization .....	54
2.1.2.54	[ISO/IEC9075-14:2011] X076, XMLSerialize: VERSION .....	54
2.1.2.55	[ISO/IEC9075-14:2011] X077, XMLSerialize: explicit ENCODING option .....	55
2.1.2.56	[ISO/IEC9075-14:2011] X078, XMLSerialize: explicit XML declaration .....	55
2.1.2.57	[ISO/IEC9075-14:2011] X084, XML namespace declarations in compound statements .....	55
2.1.2.58	[ISO/IEC9075-14:2011] X085, Predefined namespace prefixes .....	56
2.1.2.59	[ISO/IEC9075-14:2011] X086, XML namespace declarations in XMLTable.....	56
2.1.2.60	[ISO/IEC9075-14:2011] X090, XML document predicate .....	57
2.1.2.61	[ISO/IEC9075-14:2011] X091, XML content predicate.....	57
2.1.2.62	[ISO/IEC9075-14:2011] X096, XMLExists .....	57
2.1.2.63	[ISO/IEC9075-14:2011] X100, Host language support for XML: CONTENT option .....	58
2.1.2.64	[ISO/IEC9075-14:2011] X101, Host language support for XML: DOCUMENT option .....	58
2.1.2.65	[ISO/IEC9075-14:2011] X110, Host language support for XML: VARCHAR mapping.....	58
2.1.2.66	[ISO/IEC9075-14:2011] X111, Host language support for XML: CLOB mapping.....	59
2.1.2.67	[ISO/IEC9075-14:2011] X112, Host language support for XML: BLOB mapping.....	59
2.1.2.68	[ISO/IEC9075-14:2011] X113, Host language support for XML: STRIP WHITESPACE option.....	60
2.1.2.69	[ISO/IEC9075-14:2011] X114, Host language support for XML: PRESERVE WHITESPACE option.....	60
2.1.2.70	[ISO/IEC9075-14:2011] X120, XML parameters in SQL routines.....	61
2.1.2.71	[ISO/IEC9075-14:2011] X121, XML parameters in external routines .....	61
2.1.2.72	[ISO/IEC9075-14:2011] X131, Query-level XMLBINARY clause.....	61
2.1.2.73	[ISO/IEC9075-14:2011] X132, XMLBINARY clause in DML .....	62
2.1.2.74	[ISO/IEC9075-14:2011] X133, XMLBINARY clause in DDL .....	62
2.1.2.75	[ISO/IEC9075-14:2011] X134, XMLBINARY clause in compound statements ..	63
2.1.2.76	[ISO/IEC9075-14:2011] X135, XMLBINARY clause in subqueries .....	63
2.1.2.77	[ISO/IEC9075-14:2011] X141, IS VALID predicate: data-driven case.....	64
2.1.2.78	[ISO/IEC9075-14:2011] X142, IS VALID predicate: ACCORDING TO clause ...	64
2.1.2.79	[ISO/IEC9075-14:2011] X143, IS VALID predicate: ELEMENT clause .....	64

2.1.2.80	[ISO/IEC9075-14:2011] X144, IS VALID predicate: schema location.....	65
2.1.2.81	[ISO/IEC9075-14:2011] X145, IS VALID predicate outside check constraints.....	65
2.1.2.82	[ISO/IEC9075-14:2011] X151, IS VALID predicate with DOCUMENT option....	65
2.1.2.83	[ISO/IEC9075-14:2011] X152, IS VALID predicate with CONTENT option .....	66
2.1.2.84	[ISO/IEC9075-14:2011] X153, IS VALID predicate with SEQUENCE option ....	66
2.1.2.85	[ISO/IEC9075-14:2011] X155, IS VALID predicate: NAMESPACE without ELEMENT clause .....	67
2.1.2.86	[ISO/IEC9075-14:2011] X157, IS VALID predicate: NO NAMESPACE with ELEMENT clause .....	67
2.1.2.87	[ISO/IEC9075-14:2011] X160, Basic Information Schema for registered XML Schemas.....	67
2.1.2.88	[ISO/IEC9075-14:2011] X161, Advanced Information Schema for registered XML Schemas.....	69
2.1.2.89	[ISO/IEC9075-14:2011] X170, XML null handling options .....	71
2.1.2.90	[ISO/IEC9075-14:2011] X171, NIL ON NO CONTENT option.....	71
2.1.2.91	[ISO/IEC9075-14:2011] X181, XML(DOCUMENT(UNTYPED)) type.....	71
2.1.2.92	[ISO/IEC9075-14:2011] X182, XML(DOCUMENT(ANY)) type .....	72
2.1.2.93	[ISO/IEC9075-14:2011] X190, XML(SEQUENCE) type .....	72
2.1.2.94	[ISO/IEC9075-14:2011] X191, XML(DOCUMENT(XMLSCHEMA)) type .....	72
2.1.2.95	[ISO/IEC9075-14:2011] X192, XML(CONTENT(XMLSCHEMA)) type.....	73
2.1.2.96	[ISO/IEC9075-14:2011] X200, XMLQuery .....	73
2.1.2.97	[ISO/IEC9075-14:2011] X201, XMLQuery: RETURNING CONTENT .....	73
2.1.2.98	[ISO/IEC9075-14:2011] X202, XMLQuery: RETURNING SEQUENCE .....	74
2.1.2.99	[ISO/IEC9075-14:2011] X203, XMLQuery: passing a context item.....	74
2.1.2.100	[ISO/IEC9075-14:2011] X204, XMLQuery: initializing an XQuery variable....	74
2.1.2.101	[ISO/IEC9075-14:2011] X205, XMLQuery: EMPTY ON EMPTY option .....	75
2.1.2.102	[ISO/IEC9075-14:2011] X206, XMLQuery: NULL ON EMPTY option .....	75
2.1.2.103	[ISO/IEC9075-14:2011] X211, XML 1.1 support .....	75
2.1.2.104	[ISO/IEC9075-14:2011] X221, XML passing mechanism BY VALUE .....	76
2.1.2.105	[ISO/IEC9075-14:2011] X222, XML passing mechanism BY REF .....	77
2.1.2.106	[ISO/IEC9075-14:2011] X231, XML(CONTENT(UNTYPED)) type.....	77
2.1.2.107	[ISO/IEC9075-14:2011] X232, XML(CONTENT(ANY)) type .....	77
2.1.2.108	[ISO/IEC9075-14:2011] X241, RETURNING CONTENT in XML publishing .....	78
2.1.2.109	[ISO/IEC9075-14:2011] X242, RETURNING SEQUENCE in XML publishing ...	79
2.1.2.110	[ISO/IEC9075-14:2011] X251, Persistent XML values of XML(DOCUMENT(UNTYPED)) type.....	80
2.1.2.111	[ISO/IEC9075-14:2011] X252, Persistent XML values of XML(DOCUMENT(ANY)) type .....	80
2.1.2.112	[ISO/IEC9075-14:2011] X253, Persistent XML values of XML(CONTENT(UNTYPED)) type .....	81
2.1.2.113	[ISO/IEC9075-14:2011] X254, Persistent XML values of XML(CONTENT(ANY)) type.....	81
2.1.2.114	[ISO/IEC9075-14:2011] X255, Persistent XML values of XML(SEQUENCE) type.....	82
2.1.2.115	[ISO/IEC9075-14:2011] X256, Persistent XML values of XML(DOCUMENT(XMLSCHEMA)) type .....	82
2.1.2.116	[ISO/IEC9075-14:2011] X257, Persistent XML values of XML(CONTENT(XMLSCHEMA)) type.....	83
2.1.2.117	[ISO/IEC9075-14:2011] X260, XML type: ELEMENT clause .....	83
2.1.2.118	[ISO/IEC9075-14:2011] X261, XML type: NAMESPACE without ELEMENT clause .....	83

2.1.2.119	[ISO/IEC9075-14:2011] X263, XML type: NO NAMESPACE with ELEMENT clause .....	84
2.1.2.120	[ISO/IEC9075-14:2011] X264, XML type: schema location.....	84
2.1.2.121	[ISO/IEC9075-14:2011] X271, XMLValidate: data-driven case.....	84
2.1.2.122	[ISO/IEC9075-14:2011] X272, XMLValidate: ACCORDING TO clause .....	85
2.1.2.123	[ISO/IEC9075-14:2011] X273, XMLValidate: ELEMENT clause .....	85
2.1.2.124	[ISO/IEC9075-14:2011] X274, XMLValidate: schema location .....	85
2.1.2.125	[ISO/IEC9075-14:2011] X281, XMLValidate: with DOCUMENT option .....	86
2.1.2.126	[ISO/IEC9075-14:2011] X282, XMLValidate with CONTENT option .....	86
2.1.2.127	[ISO/IEC9075-14:2011] X283, XMLValidate with SEQUENCE option .....	86
2.1.2.128	[ISO/IEC9075-14:2011] X284, XMLValidate NAMESPACE without ELEMENT clause .....	87
2.1.2.129	[ISO/IEC9075-14:2011] X286, XMLValidate: NO NAMESPACE with ELEMENT clause .....	87
2.1.2.130	[ISO/IEC9075-14:2011] X300, XMLTable.....	88
2.1.2.131	[ISO/IEC9075-14:2011] X301, XMLTable: derived column list option .....	88
2.1.2.132	[ISO/IEC9075-14:2011] X302, XMLTable: ordinality column option.....	88
2.1.2.133	[ISO/IEC9075-14:2011] X303, XMLTable: column default option .....	89
2.1.2.134	[ISO/IEC9075-14:2011] X304, XMLTable: passing a context item .....	89
2.1.2.135	[ISO/IEC9075-14:2011] X305, XMLTable: initializing an XQuery variable.....	89
2.1.2.136	[ISO/IEC9075-14:2011] X400, Name and identifier mapping.....	90
2.2	Clarifications .....	90
2.3	Error Handling .....	90
2.4	Security.....	90
<b>3</b>	<b>Change Tracking.....</b>	<b>91</b>
<b>4</b>	<b>Index .....</b>	<b>92</b>

# 1 Introduction

The SQL Server Transact-SQL ISO/IEC 9075-14 Standards Support Document provides a statement of standards support. It is intended for use in conjunction with the Microsoft technical specifications, publicly available standards specifications, network programming art, and Microsoft distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A Standards Support document does not require the use of Microsoft programming tools or programming environments in order to implement the standard. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

The **Transact-SQL** language is a procedural extension of the SQL database programming language as implemented by Microsoft [[MSDN-Transact-SQLRef](#)]. Transact-SQL supports and extends ANSI SQL. The Transact-SQL dialect is based on the SQL language specification (International Standard ISO/IEC 9075).

The SQL Server Transact-SQL ISO/IEC 9075-14 Standards Support Document describes the level of support that is provided by Transact-SQL in both SQL Server 2008 R2 and SQL Server 2012 for Part 14: XML-Related Specifications (SQL/XML) of both the [[ISO/IEC9075-14:2008](#)] and [[ISO/IEC9075-14:2011](#)] specifications and for the XML standards on which the international standards are based [[XML10/5](#)] [[XMLNS3](#)] [[XMLSCHEMA1/2](#)] [[XMLSCHEMA2/2](#)]. Unless otherwise stated, the specification excerpts are quoted from [[ISO/IEC9075-14:2011](#)]. Differences between the [[ISO/IEC9075-14:2008](#)] and [[ISO/IEC9075-14:2011](#)] excerpts are called out where they occur, unless the difference is minor, such as in subclause renumbering.

## 1.1 Glossary

The following terms are specific to this document:

**Transact-SQL:** The Microsoft proprietary version of SQL, the structured query language.

## 1.2 References

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[ISO/IEC9075-14:2008] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 14: XML-Related Specifications (SQL/XML)", INCITS/ISO/IEC 9075-14:2008, <http://webstore.ansi.org/RecordDetail.aspx?sku=INCITS%2fISO%2fIEC+9075-14-2008>

**Note** There is a charge to download the specification.

[ISO/IEC9075-14:2011] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 14: XML-Related Specifications (SQL/XML)", ISO/IEC 9075-14:2011, December 2011, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=53686](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53686)

**Note** There is a charge to download the specification.



[XML10/5] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>

[XMLNS3] Bray, T., Hollander, D., and Layman, A., Eds., "Namespaces in XML", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1/2] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, <http://www.w3.org/TR/xmlschema-1/>

[XMLSCHEMA2/2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, <http://www.w3.org/TR/xmlschema-2>

### 1.2.2 Informative References

[ISO/IEC9075-1:2011] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 1: Framework (SQL/Framework)", ISO/IEC 9075-1:2011, December 2011, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=53681](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53681)

**Note** There is a charge to download the specification.

[MSDN-SSDBEng] Microsoft Corporation, "SQL Server Database Engine", <http://msdn.microsoft.com/en-us/library/ms187875.aspx>

[MSDN-Transact-SQLRef] Microsoft Corporation, "Transact-SQL Reference (Database Engine)", <http://msdn.microsoft.com/en-us/library/bb510741.aspx>

### 1.3 Microsoft Implementations

Microsoft SQL Server 2008 R2

Microsoft SQL Server 2012

Transact-SQL

### 1.4 Standards Support Requirements

An SQL implementation that is fully compliant with the SQL standard and the related XML standards enumerated in the preceding Normative References section implements all mandatory features of the standards and optionally implements any optional features. For Part 14 of the standard (International Standard ISO/IEC 9075-14), the normative variations from mandatory features are listed in [Mandatory Features \(section 2.1.1\)](#) and the normative variations from optional features are listed in [Optional Features \(section 2.1.2\)](#).

This document covers Transact-SQL alignment with normative statements in the ISO/IEC standard. This document does not include:

- Clarifications of ambiguity in the target specification.
- Intended points of variability in the target specification, such as the use of MAY, SHOULD, or RECOMMENDED.
- The use of extensibility points (such as optional implementation-specific data).

The following table lists the sections of [\[XML10/5\]](#) that are considered normative and that are considered informative.

Section	Normative/Informative
1	Informative
2 - 5	Normative
6	Informative
Appendices A - J	Informative

The following table lists the sections of [\[XMLNS3\]](#) that are considered normative and that are considered informative.

Section	Normative/Informative
1	Informative
2 - 8	Normative
Appendices A - F	Informative

The following table lists the sections of [\[XMLSCHEMA1/2\]](#) that are considered normative and that are considered informative.

Section	Normative/Informative
1	Informative
2 - 5	Normative
Appendices A - D	Normative
Appendices E - J	Informative

The following table lists the sections of [\[XMLSCHEMA2/2\]](#) that are considered normative and that are considered informative.

Section	Normative/Informative
1	Informative
2 - 5	Normative
Appendix A	Normative
Appendix B	Informative
Appendices C - F	Normative
Appendices G - I	Informative

The following table lists the sections of [\[ISO/IEC9075-14:2008\]](#) that are considered normative and that are considered informative.

Section	Normative/Informative
---------	-----------------------

Section	Normative/Informative
1 - 2	Informative
3 - 24	Normative
Appendices A - G	Informative

The following table lists the sections of [\[ISO/IEC9075-14:2011\]](#) that are considered normative and that are considered informative.

Section	Normative/Informative
1 - 2	Informative
3 - 24	Normative
Appendices A - G	Informative

## 1.5 Notation

The following notations are used to identify clarifications in Standards Support Statements (section [2](#)).

Notation	Explanation
C####	This notation identifies a clarification of ambiguity in the target specification. This includes imprecise statements, omitted information, discrepancies, and errata. This does not include data formatting clarifications.
V####	This notation identifies an intended point of variability in the target specification, such as the use of MAY, SHOULD, or RECOMMENDED. This does not include extensibility points.
E####	Because the use of extensibility points (such as optional implementation-specific data) could impair interoperability, this notation identifies such points in the target specification.

## 2 Standards Support Statements

### 2.1 Normative Variations

The following subsections detail the normative variations in Transact-SQL from [\[XML10/5\]](#), [\[XMLNS3\]](#), [\[XMLSCHEMA1/2\]](#), [\[XMLSCHEMA2/2\]](#), [\[ISO/IEC9075-14:2008\]](#), and [\[ISO/IEC9075-14:2011\]](#), as applicable.

#### 2.1.1 Mandatory Features

##### 2.1.1.1 [XML10/5] 2.8 Prolog and Document Type Declaration

V0001:

The [\[XML10/5\]](#) (section 2.8) specification states the following:

The document type declaration MUST appear before the first element in the document.

The [\[XML10/5\]](#) (section 5.1) specification states the following:

Non-validating processors are REQUIRED to check only the document entity, including the entire internal DTD subset, for well-formedness. [Definition: While they are not required to check the document for validity, they are REQUIRED to process all the declarations they read in the internal DTD subset and in any parameter entity that they read, up to the first reference to a parameter entity that they do not read; that is to say, they MUST use the information in those declarations to normalize attribute values, include the replacement text of internal entities, and supply default attribute values.]

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this declaration. By default, the parser does not allow for an internal DTD subset. However, setting the *style* parameter of the Transact-SQL CONVERT function to 2 or 3 enables limited internal DTD subset processing.

If limited internal DTD subset processing is enabled, the server can use the following information that is provided in an internal DTD subset to perform nonvalidating parse operations:

- Defaults for attributes are applied.
- Internal entity references are resolved and expanded.
- The DTD content model is checked for syntactical correctness.

The parser ignores external DTD subsets. It also does not evaluate the XML declaration to check whether the **standalone** attribute is set to yes or no, but instead parses the XML instance as if it is a stand-alone document, as indicated by the [\[XML10/5\]](#) (section 2.9) specification:

Markup declarations can affect the content of the document, as passed from an XML processor to an application; examples are attribute defaults and entity declarations. The standalone document declaration, which may appear as a component of the XML declaration, signals whether

or not there are such declarations which appear external to the document entity or in parameter entities. [Definition: An external markup declaration is defined as a markup declaration occurring in the external subset or in a parameter entity (external or internal, the latter being included because non-validating processors are not required to read them).]

This variation pertains to XML parser functionality in SQL Server.

### 2.1.1.2 [XML10/5] 2.10 White Space Handling

V0002:

The [\[XML10/5\]](#) (section 2.10) specification states the following:

An XML processor MUST always pass all characters in a document that are not markup through to the application. A validating XML processor MUST also inform the application which of these characters constitute white space appearing in element content.

The [\[XML10/5\]](#) (section 2.4) specification states the following:

[Definition: Markup takes the form of start-tags, end-tags, empty-element tags, entity references, character references, comments, CDATA section delimiters, document type declarations, processing instructions, XML declarations, text declarations, and any white space that is at the top level of the document entity (that is, outside the document element and not inside any other markup).]

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this attribute. By default, the parser discards insignificant white space. (White space is defined by the specification as spaces, tabs, and blank lines.)

White space that is inside element content is considered insignificant if it occurs inside a sequence of white-space-only character data that is delimited by markup, such as begin or end tags, and is not entitized. CDATA sections are ignored.

Setting the *style* parameter of the Transact-SQL CONVERT function to 1 or 3 preserves insignificant white space. This style setting sets the default **xml:space** handling to behave as if **xml:space="preserve"** is specified instead. The [\[XML10/5\]](#) (section 2.10) specification defines the behavior of the **xml:space** attribute as follows:

A special attribute named xml:space may be attached to an element to signal an intention that in that element, white space should be preserved by applications. In valid documents, this attribute, like any other, MUST be declared if it is used. When declared, it MUST be given as an enumerated type whose values are one or both of "default" and "preserve"....

The value "default" signals that applications' default white-space processing modes are acceptable for this element; the value "preserve" indicates the intent that applications preserve all the white space. This declared intent is considered to apply to all elements within the content of the element where it is specified, unless overridden with another instance of the xml:space attribute.

This variation pertains to XML parser functionality in SQL Server.

### 2.1.1.3 [XMLSCHEMA1/2] minOccurs and maxOccurs (throughout)

V0003:

The [\[XMLSCHEMA1/2\]](#) specification states the following:

```
maxOccurs = (nonNegativeInteger | unbounded) : 1
minOccurs = nonNegativeInteger : 1
```

The [\[XMLSCHEMA2/2\]](#) (section 3.3.20) specification states the following:

3.3.20 nonNegativeInteger

[Definition:] nonNegativeInteger is derived from integer by setting the value of minInclusive to be 0. This results in the standard mathematical concept of the non-negative integers. The value space of nonNegativeInteger is the infinite set {0,1,2,...}. The base type of nonNegativeInteger is integer.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this attribute. The values for **minOccurs** and **maxOccurs** attributes must fit into 4-byte integers. Schemas that do not conform are rejected by the server.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.4 [XMLSCHEMA1/2] 2.2.2.2 Element Substitution Group

V0004:

The specification states the following:

[Definition:] Through the new mechanism of element substitution groups, XML Schemas provides a more powerful model supporting substitution of one named element for another. Any top-level element declaration can serve as the defining member, or head, for an element substitution group. Other top-level element declarations, regardless of target namespace, can be designated as members of the substitution group headed by this element. In a suitably enabled content model, a reference to the head validates not just the head itself, but elements corresponding to any other member of the substitution group as well.

All such members must have type definitions which are either the same as the head's type definition or restrictions or extensions of it. Therefore, although the names of elements can vary widely as new namespaces and members of the substitution group are defined, the content of member elements is strictly limited according to the type definition of the substitution group head.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this mechanism. Members may not be added to an existing substitution group in an XML schema collection. A substitution group in a Transact-SQL XML schema is restricted, in that the head element and all its member elements must be defined in the same CREATE XML SCHEMA COLLECTION or ALTER XML SCHEMA COLLECTION statement.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.5 [XMLSCHEMA1/2] 3.4.3 Constraints on XML Representations of Complex Type Definitions

V0005:

The specification states the following:

```
2 If the <simpleContent> alternative is chosen, all of the following must be true:
2.1 The type definition ·resolved· to by the ·actual value· of the base [attribute] must be
one of the following:
2.1.1 a complex type definition whose {content type} is a simple type definition;
2.1.2 only if the <restriction> alternative is also chosen, a complex type definition whose
{content type} is mixed and a particle which is ·emptiable·, as defined in Particle Emptiable
($3.9.6);
2.1.3 only if the <extension> alternative is also chosen, a simple type definition.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this representation. Transact-SQL does not support restricting a mixed type to a simple content data type. For more information, see [\[MSDN-Transact-SQLRef\]](#).

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.6 [XMLSCHEMA1/2] 3.8.2 XML Representation of Model Group Schema Components

V0006:

The specification states the following:

```
The XML representation for a model group schema component is either an <all>, a <choice> or a
<sequence> element information item. The correspondences between the properties of those
information items and properties of the component they correspond to are as follows:...
```

```
<choice
  id = ID
  maxOccurs = (nonNegativeInteger | unbounded) : 1
  minOccurs = nonNegativeInteger : 1
  {any attributes with non-schema namespace . . .}>
  Content: (annotation?, (element | group | choice | sequence | any)*)
</choice>
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this representation. Transact-SQL rejects schemas that contain an **xsd:choice** particle without children, unless the particle is defined by using a **minOccurs** attribute value of zero.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.7 [XMLSCHEMA1/2] 3.10.2 XML Representation of Wildcard Schema Components

V0007:

The specification states the following:

```
The XML representation for a wildcard schema component is an <any> or <anyAttribute> element
information item...
{namespace constraint} Dependent on the 'actual value' of the namespace [attribute]: if
absent, then any, otherwise as follows:
##any
    any
##other
    a pair of not and the 'actual value' of the targetNamespace [attribute] of the <schema>
ancestor element information item if present, otherwise 'absent'.
otherwise
    a set whose members are namespace names corresponding to the space-delimited substrings
of the string, except
    1 if one such substring is ##targetNamespace, the corresponding member is the 'actual
value' of the targetNamespace [attribute] of the <schema> ancestor element information item
if present, otherwise 'absent'.
    2 if one such substring is ##local, the corresponding member is 'absent'.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this representation. In Transact-SQL, the local namespace must be explicitly specified for the **xsd:any** element. Schemas must not contain an empty string ("") as a value for the namespace attribute. Instead, to indicate an unqualified element or attribute as the instance of the wildcard character, the implementer must use "##local" explicitly.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.8 [XMLSCHEMA1/2] 3.11 Identity-constraint Definitions

V0008:

The [\[XMLSCHEMA1/2\]](#) (section 3.11) specification states the following:

```
Identity-constraint definition components provide for uniqueness and reference constraints
with respect to the contents of multiple elements and attributes.
```

The [\[XMLSCHEMA1/2\]](#) (section 3.11.2) specification states the following:

```
The XML representation for an identity-constraint definition schema component is either a
<key>, a <keyref> or a <unique> element information item.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:



Transact-SQL does not support these definitions. Transact-SQL does not support the **xsd:key**, **xsd:keyref**, and **xsd:unique** XSD-based constraints for enforcing uniqueness or establishing keys and key references. XML schemas that contain these elements cannot be registered.

See [\[ISO/IEC9075-1:2011\]](#) for the definition of "constraint".

This variation pertains to XML Schemas [XMLSCHEMA1/2] [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.9 [XMLSCHEMA1/2] 4.2.1 Assembling a schema for a single target namespace from multiple schema definition documents

V0009:

The specification states the following:

Schema components for a single target namespace can be assembled from several `.schema` documents, that is several `<schema>` element information items....

A `<schema>` information item may contain any number of `<include>` elements. Their `schemaLocation` attributes, consisting of a URI reference, identify other `.schema` documents, that is `<schema>` information items.

The `.XML Schema` corresponding to `<schema>` contains not only the components corresponding to its definition and declaration [children], but also all the components of all the `.XML Schemas` corresponding to any `<include>`d schema documents. Such included schema documents must either (a) have the same `targetNamespace` as the `<include>`ing schema document, or (b) no `targetNamespace` at all, in which case the `<include>`d schema document is converted to the `<include>`ing schema document's `targetNamespace`.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this element. XML schemas that include the **xsd:include** element are rejected by the server.

As a solution, XML schemas that include the **xsd:include** directive can be preprocessed to copy and merge the contents of any included schemas into a single schema for uploading to the server. For more information, see [\[MSDN-Transact-SQLRef\]](#).

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.10 [XMLSCHEMA2/2] 2.5.1.3 Union datatypes

V0010:

The [\[XMLSCHEMA2/2\]](#) (section 2.5.1.3) specification states the following:

The `.value space` and `.lexical space` of a `.union` datatype are the union of the `.value space`s and `.lexical space`s of its `.memberTypes`. `.union` datatypes are always `.derived`. Currently, there are no `.built-in` `.union` datatypes.

The [\[XMLSCHEMA2/2\]](#) (section 2.5.2.1) specification states the following:

[Definition:] A datatype is said to be *derived* by restriction from another datatype when values for zero or more *constraining facets* are specified that serve to constrain its *value space* and/or its *lexical space* to a subset of those of its *base type*.

[Definition:] Every datatype that is *derived* by restriction is defined in terms of an existing datatype, referred to as its *base type*. *base types* can be either *primitive* or *derived*.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support these restrictions. SQL Server does not support restrictions from union types.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.11 [\[XMLSCHEMA2/2\]](#) 3.2.1 string

V0011:

The specification states the following:

[Definition:] The string datatype represents character strings in XML. The *value space* of string is the set of finite-length sequences of characters (as defined in [\[XML 1.0 \(Second Edition\)\]](#)) that *match* the Char production from [\[XML 1.0 \(Second Edition\)\]](#). A character is an atomic unit of communication; it is not further specified except to note that every character has a corresponding Universal Character Set code point, which is an integer.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Values of this type must comply with the format of the SQL **nvarchar(max)** type.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.12 [\[XMLSCHEMA2/2\]](#) 3.2.3 decimal

V0012:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.3) specification states the following:

[Definition:] decimal represents a subset of the real numbers, which can be represented by decimal numerals. The *value space* of decimal is the set of numbers that can be obtained by multiplying an integer by a non-positive power of ten, i.e., expressible as  $i \times 10^{-n}$  where  $i$  and  $n$  are integers and  $n \geq 0$ . Precision is not reflected in this value space; the number 2.0 is not distinct from the number 2.00. The *order-relation* on decimal is the order relation on real numbers, restricted to this subset.

Note: All *minimally conforming* processors *must* support decimal numbers with a minimum of 18 decimal digits (i.e., with a *totalDigits* of 18). However, *minimally conforming*

processors may set an application-defined limit on the maximum number of decimal digits they are prepared to support, in which case that application-defined maximum number must be clearly documented....

The [XMLSCHEMA2/2] (section 3.2.3.1) specification states the following:

decimal has a lexical representation consisting of a finite-length sequence of decimal digits (#x30-#x39) separated by a period as a decimal indicator. An optional leading sign is allowed. If the sign is omitted, "+" is assumed. Leading and trailing zeroes are optional. If the fractional part is zero, the period and following zero(es) can be omitted. For example: -1.23, 12678967.543233, +100000.00, 210.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The **xs:decimal** type represents arbitrary precision decimal numbers. Transact-SQL does not support variable precision decimals. Minimally conforming XML processors must support decimal numbers with a minimum of totalDigits=18. Transact-SQL supports totalDigits=38, but limits the fractional digits to 10. All **xs:decimal**-instanced values are represented internally on the server by the SQL type numeric (38, 10).

Values of this type must comply with the format of the SQL **numeric** type. This type internally represents the support of numbers up to a total of 38 digits, with 10 of those digit positions reserved for fractional precision.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [XMLSCHEMA2/2] functionality in SQL Server.

### 2.1.1.13 [XMLSCHEMA2/2] 3.2.4 float

V0013:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.4.1) specification states the following:

Float values have a lexical representation consisting of a mantissa followed, optionally, by the character "E" or "e", followed by an exponent. The exponent must be an integer. The mantissa must be a decimal number. The representations for exponent and mantissa must follow the lexical rules for integer and decimal. If the "E" or "e" and the following exponent are omitted, an exponent value of 0 is assumed.

The special values positive and negative infinity and not-a-number have lexical representations INF, -INF and NaN, respectively. Lexical representations for zero may take a positive or negative sign.

For example, -1E4, 1267.43233E12, 12.78e-2, 12 , -0, 0 and INF are all legal literals for float.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Values of this type must comply with the format of the SQL **real** type.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### 2.1.1.14 [\[XMLSCHEMA2/2\]](#) 3.2.5 double

V0014:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.5.1) specification states the following:

double values have a lexical representation consisting of a mantissa followed, optionally, by the character "E" or "e", followed by an exponent. The exponent must be an integer. The mantissa must be a decimal number. The representations for exponent and mantissa must follow the lexical rules for integer and decimal. If the "E" or "e" and the following exponent are omitted, an exponent value of 0 is assumed.

The special values positive and negative infinity and not-a-number have lexical representations INF, -INF and NaN, respectively. Lexical representations for zero may take a positive or negative sign.

For example, -1E4, 1267.43233E12, 12.78e-2, 12 , -0, 0 and INF are all legal literals for double.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Values of this type must comply with the format of the SQL **float** type.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### 2.1.1.15 [\[XMLSCHEMA2/2\]](#) 3.2.6 duration

V0015:

The specification states the following:

[Definition:] duration represents a duration of time. The value space of duration is a six-dimensional space where the coordinates designate the Gregorian year, month, day, hour, minute, and second components defined in § 5.5.3.2 of [ISO 8601], respectively. These components are ordered in their significance by their order of appearance i.e. as year, month, day, hour, minute, and second.

Note:

All minimally conforming processors must support year values with a minimum of 4 digits (i.e., YYYY) and a minimum fractional second precision of milliseconds or three decimal digits (i.e. s.sss). However, minimally conforming processors may set an application-defined limit on the maximum number of digits they are prepared to support in these two cases, in which case that application-defined maximum number must be clearly documented.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The year part has to be within the range of  $-2^{31}$  to  $2^{31}-1$ . The month, day, hour, minute, and second parts must all be within the range of 0 to 9999. The seconds part has an additional three digits of precision to the right of the decimal point.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.16 [XMLSCHEMA2/2] 3.2.7 dateTime

V0016:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.7) specification states the following:

```
[Definition:] dateTime values may be viewed as objects with integer-valued year, month, day, hour and minute properties, a decimal-valued second property, and a boolean timezoned property....
```

```
All timezoned times are Coordinated Universal Time (UTC, sometimes called "Greenwich Mean Time"). Other timezones indicated in lexical representations are converted to UTC during conversion of literals to values. "Local" or untimezoned times are presumed to be the time in the timezone of some unspecified locality as prescribed by the appropriate legal authority; currently there are no legally prescribed timezones which are durations whose magnitude is greater than 14 hours....
```

The [\[XMLSCHEMA2/2\]](#) (section 3.2.7.1) specification states the following:

```
The 'lexical space' of dateTime consists of finite-length sequences of characters of the form: '-'? yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzz)?, where
```

```
'-'? yyyy is a four-or-more digit optionally negative-signed numeral that represents the year; if more than four digits, leading zeros are prohibited, and '0000' is prohibited (see the Note above (§3.2.7); also note that a plus sign is not permitted); the remaining '-'s are separators between parts of the date portion;
the first mm is a two-digit numeral that represents the month;
dd is a two-digit numeral that represents the day;
'T' is a separator indicating that time-of-day follows;
hh is a two-digit numeral that represents the hour; '24' is permitted if the minutes and seconds represented are zero, and the dateTime value so represented is the first instant of the following day (the hour property of a dateTime object in the 'value space' cannot have a value greater than 23);
':' is a separator between parts of the time-of-day portion;
the second mm is a two-digit numeral that represents the minute;
ss is a two-integer-digit numeral that represents the whole seconds;
'.' s+ (if present) represents the fractional seconds;
zzzzz (if present) represents the timezone (as described below).
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Time zone information is fully supported for **xs:dateTime** values for XML Schema validation. In the Microsoft SQL Server 2005 backwards-compatibility mode, time zone information is always normalized to Coordinated Universal Time (UTC).

For elements of **dateTime** type, the server converts the time provided to UTC by using the offset value (for example, "-05:00") and returning the corresponding UTC time.

The hour part in the **time zone** subfield must be within the accepted range of -14 to +14. The year part must be within the range of 1 to 9999. The month part must be within the range of 1 to 12. The day part must be within the range of 1 to 31 and must be a valid calendar date. For example, SQL Server detects and returns an error for an invalid date, such as 1974-02-31, because February does not have 31 days.

The second component supports 100-nanosecond precision. The time zone indication is optional.

SQL Server 2005 supports years in the range -9999 to 9999. SQL Server now supports a more restricted range of years. For more information, see "Typed XML Compared to Untyped XML" in [\[MSDN-SSDBEng\]](#).

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.17 [\[XMLSCHEMA2/2\]](#) 3.2.8 time

V0017:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.8) specification states the following:

[Definition:] time represents an instant of time that recurs every day. The value space of time is the space of time of day values as defined in § 5.3 of [ISO 8601]. Specifically, it is a set of zero-duration daily time instances.

Since the lexical representation allows an optional time zone indicator, time values are partially ordered because it may not be able to determine the order of two values one of which has a time zone and the other does not. The order relation on time values is the Order relation on dateTime (§3.2.7.4) using an arbitrary date.... Pairs of time values with or without time zone indicators are totally ordered.

The [\[XMLSCHEMA2/2\]](#) (section 3.2.8.1) specification states the following:

The lexical representation for time is the left truncated lexical representation for dateTime: hh:mm:ss.sss with optional following time zone indicator. For example, to indicate 1:20 pm for Eastern Standard Time which is 5 hours behind Coordinated Universal Time (UTC), one would write: 13:20:00-05:00....

The [\[XMLSCHEMA2/2\]](#) (section 3.2.8.2) specification states the following:

The canonical representation for time is defined by prohibiting certain options from the Lexical representation (§3.2.8.1). Specifically, either the time zone must be omitted or, if present, the time zone must be Coordinated Universal Time (UTC) indicated by a "Z". Additionally, the canonical representation for midnight is 00:00:00.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Time zone information is fully supported for **xs:time** values for XML Schema validation. In SQL Server 2005 backwards-compatibility mode, time zone information is always normalized to Coordinated Universal Time (UTC).

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.18 [\[XMLSCHEMA2/2\]](#) 3.2.9 date

V0018:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.9) specification states the following:

[Definition:] The `·value space·` of `date` consists of top-open intervals of exactly one day in length on the timelines of `dateTime`, beginning on the beginning moment of each day (in each `timezone`), i.e. `'00:00:00'`, up to but not including `'24:00:00'` (which is identical with `'00:00:00'` of the next day). For `nontimezoned` values, the top-open intervals disjointly cover the `nontimezoned` timeline, one per day. For `timezoned` values, the intervals begin at every minute and therefore overlap.

A "date object" is an object with `year`, `month`, and `day` properties just like those of `dateTime` objects, plus an optional `timezone`-valued `timezone` property. (As with values of `dateTime` `timezones` are a special case of durations.) Just as a `dateTime` object corresponds to a point on one of the timelines, a date object corresponds to an interval on one of the two timelines as just described.

`Timezoned` date values track the starting moment of their day, as determined by their `timezone`; said `timezone` is generally recoverable for canonical representations. [Definition:] The recoverable `timezone` is that duration which is the result of subtracting the first moment (or any moment) of the `timezoned` date from the first moment (or the corresponding moment) UTC on the same date. `·recoverable timezone·s` are always durations between `'+12:00'` and `'-11:59'....`

The [\[XMLSCHEMA2/2\]](#) (section 3.2.9.1) specification states the following:

For the following discussion, let the "date portion" of a `dateTime` or date object be an object similar to a `dateTime` or date object, with similar `year`, `month`, and `day` properties, but no others, having the same value for these properties as the original `dateTime` or date object.

The `·lexical space·` of `date` consists of finite-length sequences of characters of the form: `'? yyyy '-' mm '-' dd zzzzzz?` where the date and optional `timezone` are represented exactly the same way as they are for `dateTime`. The first moment of the interval is that represented by: `'-' yyyy '-' mm '-' dd 'T00:00:00' zzzzzz?` and the least upper bound of the interval is the timeline point represented (noncanonically) by: `'-' yyyy '-' mm '-' dd 'T24:00:00' zzzzzz?`.

Note: The `·recoverable timezone·` of a date will always be a duration between `'+12:00'` and `'-11:59'`. `Timezone` lexical representations, as explained for `dateTime`, can range from `'+14:00'` to `'-14:00'`. The result is that literals of dates with very large or very negative `timezones` will map to a "normalized" date value with a `·recoverable timezone·` different from that represented in the original representation, and a matching difference of `+/- 1` day in the date itself.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Time zone information is fully supported for **xs:date** values for XML Schema validation. In Microsoft SQL Server 2005 backwards-compatibility mode, time zone information is always normalized to Coordinated Universal Time (UTC).

The year part must be within the range of 1 to 9999. The month part must be within the range of 1 to 12. The day part must be within the range of 1 to 31 and must be a valid calendar date. For example, SQL Server detects and returns an error for an invalid date, such as 1974-02-31, because February does not have 31 days.

SQL Server 2005 supports years in the range -9999 to 9999. SQL Server now supports a more restricted range of years. For more information, see "Typed XML Compared to Untyped XML" in [\[MSDN-SSDBEng\]](#).

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [XMLSCHEMA2/2] functionality in SQL Server.

### **2.1.1.19 [XMLSCHEMA2/2] 3.2.10 gYearMonth**

V0019:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.10.1) specification states the following:

The lexical representation for gYearMonth is the reduced (right truncated) lexical representation for dateTime: CCYY-MM. No left truncation is allowed. An optional following time zone qualifier is allowed. To accommodate year values outside the range from 0001 to 9999, additional digits can be added to the left of this representation and a preceding "-" sign is allowed.

For example, to indicate the month of May 1999, one would write: 1999-05. See also ISO 8601 Date and Time Formats (§D).

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The year part must be within the range of -9999 to 9999.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [XMLSCHEMA2/2] functionality in SQL Server.

### **2.1.1.20 [XMLSCHEMA2/2] 3.2.11 gYear**

V0020:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.11.1) specification states the following:

The lexical representation for gYear is the reduced (right truncated) lexical representation for dateTime: CCYY. No left truncation is allowed. An optional following time zone qualifier is allowed as for dateTime. To accommodate year values outside the range from 0001 to 9999, additional digits can be added to the left of this representation and a preceding "-" sign is allowed.



For example, to indicate 1999, one would write: 1999. See also ISO 8601 Date and Time Formats (\$D).

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The year part must be within the range of -9999 to 9999.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### **2.1.1.21 [XMLSCHEMA2/2] 3.2.12 gMonthDay**

V0021:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.12.1) specification states the following:

The lexical representation for gMonthDay is the left truncated lexical representation for date: --MM-DD. An optional following time zone qualifier is allowed as for date. No preceding sign is allowed. No other formats are allowed.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The month part must be within the range of 1 to 12. The day part must be within the range of 1 to 31.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### **2.1.1.22 [XMLSCHEMA2/2] 3.2.13 gDay**

V0022:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.13.1) specification states the following:

The lexical representation for gDay is the left truncated lexical representation for date: --DD. An optional following time zone qualifier is allowed as for date. No preceding sign is allowed. No other formats are allowed.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The day part must be within the range of 1 to 31.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.23 [XMLSCHEMA2/2] 3.2.14 gMonth

V0023:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.14.1) specification states the following:

The lexical representation for gMonth is the left and right truncated lexical representation for date: --MM. An optional following time zone qualifier is allowed as for date. No preceding sign is allowed. No other formats are allowed.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The month part must be within the range of 1 to 12.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [XMLSCHEMA2/2] functionality in SQL Server.

### 2.1.1.24 [XMLSCHEMA2/2] 3.2.17 anyURI

V0024:

The [\[XMLSCHEMA2/2\]](#) (section 3.2.17.1) specification states the following:

The lexical space of anyURI is finite-length character sequences which, when the algorithm defined in Section 5.4 of [XML Linking Language] is applied to them, result in strings which are legal URIs according to [RFC 2396], as amended by [RFC 2732].

Note: Spaces are, in principle, allowed in the lexical space of anyURI, however, their use is highly discouraged (unless they are encoded by %20).

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Values of this type can be no more than 4000 Unicode characters in length.

For more information, see section [2.1.1.32](#) of this document.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [XMLSCHEMA2/2] functionality in SQL Server.

### 2.1.1.25 [XMLSCHEMA2/2] 3.2.18 QName

V0025:

The specification states the following:

[Definition:] QName represents XML qualified names. The value space of QName is the set of tuples {namespace name, local part}, where namespace name is an anyURI and local part is an NCName. The lexical space of QName is the set of strings that match the QName production of [Namespaces in XML].`

Note: The mapping between literals in the `lexical space` and values in the `value space` of QName requires a namespace declaration to be in scope for the context in which QName is used.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Transact-SQL does not support data types that are derived from **xs:QName** and that use an XML Schema restriction element. Additionally, Transact-SQL does not support union types that contain **xs:QName** as a member element.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.26 [\[XMLSCHEMA2/2\]](#) 3.2.19 NOTATION

V0026:

The specification states the following:

```
[Definition:] NOTATION represents the NOTATION attribute type .... The value space of NOTATION is the set of QNames of notations declared in the current schema. The lexical space of NOTATION is the set of all names of notations declared in the current schema (in the form of QNames)....
```

```
It is an error for NOTATION to be used directly in a schema. Only datatypes that are derived from NOTATION by specifying a value for enumeration can be used in a schema.
```

```
For compatibility ... NOTATION should be used only on attributes and should only be used in schemas with no target namespace.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this data type. The NOTATION attribute type is defined in [\[XML10/5\]](#).

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.27 [\[XMLSCHEMA2/2\]](#) 3.3.6 Name

V0027:

The [\[XMLSCHEMA2/2\]](#) (section 3.3.6) specification states the following:

```
[Definition:] Name represents XML Names. The value space of Name is the set of all strings which match the Name production of . The lexical space of Name is the set of all strings which match the Name production of [XML 1.0 (Second Edition)]. The base type of Name is token.
```

```
[XML 1.0 (Second Edition)] 2.3 Common Syntactic Constructs
```

```
Names and Tokens
```

```
[4] NameChar ::= Letter | Digit | '.' | '-' | '_' | ':' | CombiningChar | Extender
```

```
[5] Name ::= (Letter | '_' | ':') ( NameChar)*
```

```
[6] Names ::= Name (#x20 Name)* /* [E62] */
[7] Nmtoken ::= (NameChar)+
[8] Nmtokens ::= Nmtoken (#x20 Nmtoken)* /* [E62] */ (XML 1.0 (Second Edition))
```

Microsoft SQL Server 2008 R2 varies as follows:

Transact-SQL partially supports this data type. Transact-SQL limits identifiers of schema components to a maximum length of 1000 Unicode characters.

Surrogate character pairs within identifiers are not supported. The data types **nchar** and **nvarchar** store each character as a 16-bit value in an encoding called UCS-2. This encoding, which is defined by versions of Unicode prior to 1996, supports characters in the range U+0000 to U+FFFF. Newer versions of Unicode have defined additional characters in the range U+10000 to U+10FFFF that are called "supplementary characters". These characters are stored as pairs of 16-bit values, called "surrogate pairs", in an encoding that is called "UTF-16". All new `_100` level collations support linguistic sorting with supplementary characters. For more information, see [\[MSDN-Transact-SQLRef\]](#).

Microsoft SQL Server 2012 varies as follows:

UTF-16 and the correct handling of surrogate pairs are supported. For more information, see [\[MSDN-Transact-SQLRef\]](#).

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.28 [XMLSCHEMA2/2] 3.3.8 ID

V0028:

The specification states the following:

```
[Definition:] ID represents the ID attribute type from [XML 1.0 (Second Edition)]. The ·value space· of ID is the set of all strings that ·match· the NCName production in [Namespaces in XML]. The ·lexical space· of ID is the set of all strings that ·match· the NCName production in [Namespaces in XML]. The ·base type· of ID is NCName.
```

```
For compatibility ... ID should be used only on attributes.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Every XML schema component may have an **ID** attribute. Transact-SQL enforces uniqueness for **xsd:attribute** declarations of **ID** type, but does not store these values. The scope for enforcement of uniqueness is the CREATE XML SCHEMA COLLECTION and ALTER XML SCHEMA COLLECTION statements.

Transact-SQL does not support elements of type **xs:ID**. A schema must not declare elements of this type or elements derived by restriction of or extension from this type.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.29 [XMLSCHEMA2/2] 3.3.9 IDREF

V0029:

The specification states the following:

[Definition:] IDREF represents the IDREF attribute type from [XML 1.0 (Second Edition)]. The `·value space·` of IDREF is the set of all strings that `·match·` the NCName production in [Namespaces in XML]. The `·lexical space·` of IDREF is the set of strings that `·match·` the NCName production in [Namespaces in XML]. The `·base type·` of IDREF is NCName.

For compatibility ... this datatype should be used only on attributes.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this data type. A schema must not declare elements of type **xs:IDREF** or elements derived by restriction of or extension from this type.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.30 [XMLSCHEMA2/2] 3.3.10 IDREFS

V0030:

The specification states the following:

[Definition:] IDREFS represents the IDREFS attribute type from [XML 1.0 (Second Edition)]. The `·value space·` of IDREFS is the set of finite, non-zero-length sequences of IDREFs. The `·lexical space·` of IDREFS is the set of space-separated lists of tokens, of which each token is in the `·lexical space·` of IDREF. The `·itemType·` of IDREFS is IDREF.

For compatibility ... IDREFS should be used only on attributes.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this data type. A schema must not declare elements of type **xs:IDREFS** or elements derived by restriction of or extension from this type.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.31 [XMLSCHEMA2/2] 4.1.2 XML Representation of Simple Type Definition Schema Components

V0031:

The specification states the following:

The XML representation for a Simple Type Definition schema component is a `<simpleType>` element information item. The correspondences between the properties of the information item and properties of the component are as follows:

XML Representation Summary: simpleType Element Information Item

```
<simpleType
  final = (#all | List of (list | union | restriction))
  id = ID
  name = NCName
  {any attributes with non-schema namespace . . .}>
  Content: (annotation?, (restriction | list | union))
</simpleType>
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. Transact-SQL supports only millisecond precision for simple types that have second components other than **xs:time** and **xs:dateTime**, and only 100-nanosecond precision for **xs:time** and **xs:dateTime**. SQL Server puts limitations on all recognized XSD simple type enumerations.

SQL Server does not support using the "NaN" value in **xsd:simpleType** declarations.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.32 [XMLSCHEMA2/2] 4.3.1 length

V0032:

The specification states the following:

[Definition:] length is the number of units of length, where units of length varies depending on the type that is being derived from. The value of length must be a nonNegativeInteger.

See [\[XMLSCHEMA2/2\]](#) (section 3.3.20) for the definition of **nonNegativeInteger**.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The **length** facet is stored as a 32-bit **long** type. Therefore, the range of acceptable values for this value is 231.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

### 2.1.1.33 [XMLSCHEMA2/2] 4.3.2 minLength

V0033:

The specification states the following:

[Definition:] minLength is the minimum number of units of length, where units of length varies depending on the type that is being derived from. The value of minLength must be a nonNegativeInteger.

See [\[XMLSCHEMA2/2\]](#) (section 3.3.20) for the definition of **nonNegativeInteger**.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The **minLength** facet is stored as a 32-bit **long** type. Therefore, the range of acceptable values for this value is 231.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### 2.1.1.34 [\[XMLSCHEMA2/2\]](#) 4.3.3 **maxLength**

V0034:

The specification states the following:

```
[Definition:] maxLength is the maximum number of units of length, where units of length varies depending on the type that is being derived from. The value of maxLength must be a nonNegativeInteger.
```

See [\[XMLSCHEMA2/2\]](#) (section 3.3.20) for the definition of **nonNegativeInteger**.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The **maxLength** facet is stored as a 32-bit **long** type. Therefore, the range of acceptable values for this value is 231.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### 2.1.1.35 [\[XMLSCHEMA2/2\]](#) 4.3.4 **pattern**

V0035:

The specification states the following:

```
[Definition:] pattern is a constraint on the value space of a datatype which is achieved by constraining the lexical space to literals which match a specific pattern. The value of pattern must be a regular expression.
```

```
pattern provides for:
```

```
Constraining a value space to values that are denoted by literals which match a specific regular expression.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this data type. The canonical representation of a value must not violate the pattern restriction for its type.

The XSD pattern facet allows for the restriction of the lexical space of simple types. When a pattern restriction is put on a type for which there is more than one possible lexical representation, some values could cause unexpected behavior upon validation.

This behavior occurs because lexical representations of these values are not stored in the database. Therefore, the values are converted to their canonical representations when serialized as output. If a

document contains a value whose canonical form does not comply with the pattern restriction for its type, the document is rejected if a user tries to reinsert it.

To prevent this, Transact-SQL rejects any XML document that contains values that cannot be reinserted, because of the violation of pattern restrictions by their canonical forms. For example, the value "33.000" does not validate against a type derived from **xs:decimal** with a pattern restriction of "33\.\d+". Although "33.000" complies with this pattern, the canonical form, "33", does not.

Therefore, care should be taken when pattern facets are applied to types derived from the following primitive types: **boolean**, **decimal**, **float**, **double**, **dateTime**, **time**, **date**, **hexBinary**, and **base64Binary**. Transact-SQL issues a warning when any such components are added to a schema collection.

Imprecise serialization of floating-point values has a similar problem. Because of the floating-point serialization algorithm that is used by Transact-SQL, it is possible for similar but non-identical values to share the same canonical form. When a floating-point value is serialized and then reinserted, its value may change slightly. In rare cases, this may result in a value that violates any of the following facets for its type on reinsertion: **enumeration**, **minInclusive**, **minExclusive**, **maxInclusive**, or **maxExclusive**. To prevent this, Transact-SQL rejects any values of types that are derived from **xs:float** or **xs:double** and that cannot be serialized and reinserted.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in Transact-SQL.

## 2.1.2 Optional Features

### 2.1.2.1 [XML10/5] 2.1 Well-Formed XML Documents

V0036:

The specification states the following:

[Definition: A textual object is a well-formed XML document if:]

1. Taken as a whole, it matches the production labeled document.
2. It meets all the well-formedness constraints given in this specification.
3. Each of the parsed entities which is referenced directly or indirectly within the document is well-formed.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this definition. Transact-SQL enables limited use of XML fragments, as specified by [\[ISO/IEC9075-14:2008\]](#) [\[ISO/IEC9075-14:2011\]](#), with respect to the XML(CONTENT) data type.

The XML(DOCUMENT) data type may also be used if the user specifies an associated XML Schema collection. The user can create one with the CREATE XML SCHEMA COLLECTION statement.

This variation pertains to XML parser functionality in SQL Server.



## 2.1.2.2 [XMLSCHEMA1/2] 2.6.3 xsi:schemaLocation, xsi:noNamespaceSchemaLocation

V0037:

The [\[XMLSCHEMA1/2\]](#) (section 2.6.3) specification states the following:

```
The xsi:schemaLocation and xsi:noNamespaceSchemaLocation attributes can be used in a document to provide hints as to the physical location of schema documents which may be used for assessment.....
```

The [\[XMLSCHEMA1/2\]](#) (section 4.3.2.) specification states the following:

```
Unless directed otherwise, for example by the invoking application or by command line option, processors should attempt to dereference each schema document location URI in the actual value of such xsi:schemaLocation and xsi:noNamespaceSchemaLocation [attributes]....
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports these attributes. SQL Server ignores the **xsi:schemaLocation** and **xsi:noNamespaceSchemaLocation** attributes if they are present in the XML instance data that is inserted into a column or variable of **xml** data type [\[MSDN-Transact-SQLRef\]](#).

See [\[ISO/IEC9075-1:2011\]](#) for the definition of "column".

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

## 2.1.2.3 [XMLSCHEMA1/2] 3.3.2 XML Representation of Element Declaration Schema Components

V0038:

The specification states the following:

```
The XML representation for an element declaration schema component is an <element> element information item. It specifies a type definition for an element either by reference or explicitly, and may provide occurrence and default information. The correspondences between the properties of the information item and properties of the component(s) it corresponds to are as follows:
```

XML Representation Summary: ... Element Information Item

```
<element
  abstract = boolean : false
  block = (#all | List of (extension | restriction | substitution))
  default = string
  final = (#all | List of (extension | restriction))
  fixed = string
  form = (qualified | unqualified)
  id = ID
  maxOccurs = (nonNegativeInteger | unbounded) : 1
  minOccurs = nonNegativeInteger : 1
```

```

name = NCName
nillable = boolean : false
ref = QName
substitutionGroup = QName
type = QName
{any attributes with non-schema namespace . . .}>
Content: (annotation?, ((simpleType | complexType)?, (unique | key | keyref)*))
</element>

```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this representation. Transact-SQL rejects schemas in which the **block** or **final** attribute has repeated values, such as "restriction restriction" and "extension extension".

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### 2.1.2.4 [XMLSCHEMA1/2] 4.2.2 Including modified component definitions

V0039:

The specification states the following:

In order to provide some support for evolution and versioning, it is possible to incorporate components corresponding to a schema document with modifications. The modifications have a pervasive impact, that is, only the redefined components are used, even when referenced from other incorporated components, whether redefined themselves or not....

A <schema> information item may contain any number of <redefine> elements. Their schemaLocation attributes, consisting of a URI reference, identify other schema documents, that is <schema> information items.

The XML Schema corresponding to <schema> contains not only the components corresponding to its definition and declaration [children], but also all the components of all the XML Schemas corresponding to any <redefine>d schema documents....

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this element. XML schemas that include the **xsd:redefine** element are rejected by the server.

To update a schema or its components, the user can take the following steps instead:

Create a new XML Schema collection with the modified schema components.

Retype all **xml** data types (**XML DT**) that use the XML Schema collection and that are to be redefined to use the new XML Schema collection. To do this, use the ALTER COLUMN option of the ALTER TABLE command for retyping columns, or change the XML Schema collection constraints on variables or parameters.

See [\[ISO/IEC9075-1:2011\]](#) for the definitions of "column" and "constraint".

Drop the old version of the XML Schema collection.

This variation pertains to XML Schemas [\[XMLSCHEMA1/2\]](#) [\[XMLSCHEMA2/2\]](#) functionality in SQL Server.

#### **2.1.2.5 [ISO/IEC9075-14:2011] X010, XML type**

V0040:

The specification states the following:

Subclause 6.1, "<data type>":

Without Feature X010, "XML type", conforming SQL language shall not contain an <XML type>.

Subclause 6.9, "<XML value expression>":

Without Feature X010, "XML type", conforming SQL language shall not contain an <XML value expression>.

Subclause 6.10, "<XML value function>":

Without Feature X010, "XML type", conforming SQL language shall not contain an <XML value function>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. Transact-SQL supports an **xml** data type. The data type partially supports type modifiers, XML value expressions, and XML value functions.

This variation pertains to **xml** data type functionality in SQL Server.

#### **2.1.2.6 [ISO/IEC9075-14:2011] X011, Arrays of XML type**

V0041:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

#### **2.1.2.7 [ISO/IEC9075-14:2011] X012, Multisets of XML type**

V0042:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.8 [ISO/IEC9075-14:2011] X013, Distinct types of XML type**

V0043:

The specification states the following:

Subclause 12.6, "<user-defined type definition>":

Without Feature X013, "Distinct types of XML type", conforming SQL language shall not contain a <representation> that is a <predefined type> that is an XML type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. However, Transact-SQL supports user-defined alias types that have some equivalent functionality. For more information, see feature [S011](#) in [\[MS-TSQLISO02\]](#).

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.9 [ISO/IEC9075-14:2011] X014, Attributes of XML type**

V0044:

The specification states the following:

Subclause 12.7, "<attribute definition>":

Without Feature X014, "Attributes of XML type", conforming SQL language shall not contain an <attribute definition> that contains a <data type> that is based on either an XML type or a distinct type whose source type is an XML type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.10 [ISO/IEC9075-14:2011] X015, Fields of XML type**

V0045:

The specification states the following:

Subclause 6.2, "<field definition>":

Without Feature X015, "Fields of XML type", conforming SQL language shall not contain a <field definition> that contains a <data type> that is based on either an XML type or a distinct type whose source type is an XML type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.11 [ISO/IEC9075-14:2011] X020, XMLConcat**

V0046:

The specification states the following:

Subclause 6.12, "<XML concatenation>":

Without Feature X020, "XMLConcat", conforming SQL language shall not contain an <XML concatenation>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.12 [ISO/IEC9075-14:2011] X025, XMLCast**

V0047:

The specification states the following:

Subclause 6.6, "<XML cast specification>":

Without Feature X025, "XMLCast", conforming SQL language shall not contain an <XML cast specification>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions for equivalent functionality [\[MSDN-Transact-SQLRef\]](#).

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.13 [ISO/IEC9075-14:2011] X026, XMLCast: SQL cast semantics**

V0048:

The [\[ISO/IEC9075-14:2008\]](#) specification states the following:

Subclause 6.6, "<XML cast specification>":

Without Feature X026, "XMLCast: SQL cast semantics", XT is unconditionally the XML Schema type obtained by applying the General Rules of Subclause 9.5, "Mapping SQL data types to XML Schema data types", with SQLT as SQLTYPE, ENC as ENCODING, and "absent" as NULLS.

Microsoft SQL Server 2008 R2 varies as follows:

Transact-SQL does not support this feature. See the `.value()` method of the `xml` data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

Microsoft SQL Server 2012 varies as follows:

This feature is absent in the international standard [\[ISO/IEC9075-14:2011\]](#).

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.14 [ISO/IEC9075-14:2011] X030, XMLDocument**

V0049:

The specification states the following:

Subclause 6.13, "<XML document>":

Without Feature X030, "XMLDocument", conforming SQL language shall not contain an <XML document>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.15 [ISO/IEC9075-14:2011] X031, XMLElement**

V0050:

The specification states the following:

Subclause 6.14, "<XML element>":

Without Feature X031, "XMLElement", conforming SQL language shall not contain an <XML element>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the PATH mode of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.16 [ISO/IEC9075-14:2011] X032, XMLForest**

V0051:

The specification states the following:

```
Subclause 6.15, "<XML forest>":
```

```
Without Feature X032, "XMLForest", conforming SQL language shall not contain an <XML forest>.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the PATH mode of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.17 [ISO/IEC9075-14:2011] X034, XMLAgg**

V0052:

The specification states the following:

```
Subclause 11.2, "<aggregate function>":
```

```
Without Feature X034, "XMLAgg", conforming SQL language shall not contain an <XML aggregate>.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the PATH mode of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.18 [ISO/IEC9075-14:2011] X035, XMLAgg: ORDER BY option**

V0053:

The specification states the following:

```
Subclause 11.2, "<aggregate function>":
```

```
Without Feature X035, "XMLAgg: ORDER BY option", conforming SQL language shall not contain an <XML aggregate> that contains a <sort specification list>.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the PATH mode of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.19 [ISO/IEC9075-14:2011] X036, XMLComment**

V0054:

The specification states the following:

Subclause 6.11, "<XML comment>":

Without Feature X036, "XMLComment", conforming SQL language shall not contain an <XML comment>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the PATH mode of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.20 [ISO/IEC9075-14:2011] X037, XMLPI**

V0055:

The specification states the following:

Subclause 6.17, "<XML PI>":

Without Feature X037, "XMLPI", conforming SQL language shall not contain an <XML PI>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the PATH mode of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.21 [ISO/IEC9075-14:2011] X038, XMLText**

V0056:

The specification states the following:

Subclause 6.19, "<XML text>":

Without Feature X038, "XMLText", conforming SQL language shall not contain an <XML text>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the PATH mode of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.



This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.22 [ISO/IEC9075-14:2011] X040, Basic table mapping**

V0057:

The specification states the following:

Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":

Without Feature X040, "Basic table mapping", a conforming application shall not invoke this Subclause of this part of this International Standard.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.23 [ISO/IEC9075-14:2011] X041, Basic table mapping: null absent**

V0058:

The specification states the following:

Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":

Without Feature X041, "Basic table mapping: nulls absent", a conforming application shall not invoke this Subclause of this part of this International Standard with NULLS set to indicate that nulls are mapped to elements that are marked to absent elements.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the ELEMENTS ABSENT option of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.24 [ISO/IEC9075-14:2011] X042, Basic table mapping: null as nil**

V0059:

The specification states the following:

Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":

Without Feature X042, "Basic table mapping: null as nil", a conforming application shall not invoke this Subclause of this part of this International Standard with NULLS set to indicate that nulls are mapped to elements that are marked with xsi:nil="true".

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the ELEMENTS XSINIL option of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.25 [ISO/IEC9075-14:2011] X043, Basic table mapping: table as forest**

V0060:

The specification states the following:

```
Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":
```

```
Without Feature X043, "Basic table mapping: table as forest", a conforming application shall not invoke this Subclause of this part of this International Standard with TABLEFOREST set to True.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.26 [ISO/IEC9075-14:2011] X044, Basic table mapping: table as element**

V0061:

The specification states the following:

```
Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":
```

```
Without Feature X044, "Basic table mapping: table as element", a conforming application shall not invoke this Subclause of this part of this International Standard with TABLEFOREST set to False.
```

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.27 [ISO/IEC9075-14:2011] X045, Basic table mapping: with target namespace**

V0062:

The specification states the following:

```
Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":
```

Without Feature X045, "Basic table mapping: with target namespace", a conforming application shall not invoke this Subclause of this part of this International Standard with TARGETNS that is not a zero-length string.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the XMLSCHEMA directive of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.28 [ISO/IEC9075-14:2011] X046, Basic table mapping: data mapping**

V0063:

The specification states the following:

Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":

Without Feature X046, "Basic table mapping: data mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with RETURN not set to "metadata only".

The [\[ISO/IEC9075-14:2008\]](#) specification differs as follows:

Without Feature X046, "Basic table mapping: data mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with DATA set to True.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.29 [ISO/IEC9075-14:2011] X047, Basic table mapping: metadata mapping**

V0064:

The specification states the following:

Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":

Without Feature X047, "Basic table mapping: metadata mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with RETURN not set to "data only".

The [\[ISO/IEC9075-14:2008\]](#) specification differs as follows:

Without Feature X047, "Basic table mapping: metadata mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with METADATA set to True.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the XMLSCHEMA argument of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.30 [ISO/IEC9075-14:2011] X048, Basic table mapping: base64 encoding of binary strings**

V0065:

The specification states the following:

Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":

Without Feature X048, "Basic table mapping: base64 encoding of binary strings", a conforming application shall not invoke this Subclause of this part of this International Standard with ENCODING set to indicate that binary strings are to be encoded using base64.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. See the BINARY BASE64 option of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.31 [ISO/IEC9075-14:2011] X049, Basic table mapping: hex encoding of binary strings**

V0066:

The specification states the following:

Subclause 9.11, "Mapping an SQL table to XML and an XML Schema document":

Without Feature X049, "Basic table mapping: hex encoding of binary strings", a conforming application shall not invoke this Subclause of this part of this International Standard with ENCODING set to indicate that binary strings are to be encoded using hex.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.32 [ISO/IEC9075-14:2011] X050, Advanced table mapping

V0067:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X050, "Advanced table mapping", a conforming application shall not invoke this Subclause of this part of this International Standard.

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X050, "Advanced table mapping", a conforming application shall not invoke this Subclause of this part of this International Standard.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.33 [ISO/IEC9075-14:2011] X051, Advanced table mapping: null absent

V0068:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X051, "Advanced table mapping: null absent", a conforming application shall not invoke this Subclause of this part of this International Standard with NULLS set to indicate that nulls are mapped to elements that are marked to absent elements.

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X051, "Advanced table mapping: null absent", a conforming application shall not invoke this Subclause of this part of this International Standard with NULLS set to indicate that nulls are mapped to elements that are marked to absent elements.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the ELEMENTS ABSENT option of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.34 [ISO/IEC9075-14:2011] X052, Advanced table mapping: null as nil

V0069:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X052, "Advanced table mapping: null as nil", a conforming application shall not invoke this Subclause of this part of this International Standard with NULLS set to indicate that nulls are mapped to elements that are marked with xsi:nil="true".

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X052, "Advanced table mapping: null as nil", a conforming application shall not invoke this Subclause of this part of this International Standard with NULLS set to indicate that nulls are mapped to elements that are marked with xsi:nil="true".

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the ELEMENTS XSINIL option of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.35 [ISO/IEC9075-14:2011] X053, Advanced table mapping: table as forest**

V0070:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X053, "Advanced table mapping: table as forest", a conforming application shall not invoke this Subclause of this part of this International Standard with TABLEFOREST set to True.

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X053, "Advanced table mapping: table as forest", a conforming application shall not invoke this Subclause of this part of this International Standard with TABLEFOREST set to True.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.36 [ISO/IEC9075-14:2011] X054, Advanced table mapping: table as element**

V0071:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X054, "Advanced table mapping: table as element", a conforming application shall not invoke this Subclause of this part of this International Standard with TABLEFOREST set to False.

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X054, "Advanced table mapping: table as element", a conforming application shall not invoke this Subclause of this part of this International Standard with TABLEFOREST set to False.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.37 [ISO/IEC9075-14:2011] X055, Advanced table mapping: with target namespace**

V0072:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X055, "Advanced table mapping: with target namespace", a conforming application shall not invoke this Subclause of this part of this International Standard with TARGETNS that is not a zero-length string.

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X055, "Advanced table mapping: with target namespace", a conforming application shall not invoke this Subclause of this part of this International Standard with TARGETNS that is not a zero-length string.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the XMLSCHEMA directive of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.38 [ISO/IEC9075-14:2011] X056, Advanced table mapping: data mapping**

V0073:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X056, "Advanced table mapping: data mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with RETURN not set to "metadata only".

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X056, "Advanced table mapping: data mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with RETURN not set to "metadata only".

The [\[ISO/IEC9075-14:2008\]](#) specification differs as follows:

Without Feature X056, "Advanced table mapping: data mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with DATA set to True.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.39 [ISO/IEC9075-14:2011] X057, Advanced table mapping: metadata mapping**

V0074:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X057, "Advanced table mapping: metadata mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with RETURN not set to "metadata only".

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X057, "Advanced table mapping: metadata mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with RETURN not set to "metadata only".

The [\[ISO/IEC9075-14:2008\]](#) specification differs as follows:

Without Feature X057, "Advanced table mapping: metadata mapping", a conforming application shall not invoke this Subclause of this part of this International Standard with METADATA set to True.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the XMLSCHEMA argument of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.



## 2.1.2.40 [ISO/IEC9075-14:2011] X058, Advanced table mapping: base64 encoding of binary strings

V0075:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X058, "Advanced table mapping: base64 encoding of binary strings", a conforming application shall not invoke this Subclause of this part of this International Standard with ENCODING set to indicate that binary strings are to be encoded using base64.

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X058, "Advanced table mapping: base64 encoding of binary strings", a conforming application shall not invoke this Subclause of this part of this International Standard with ENCODING set to indicate that binary strings are to be encoded using base64.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the BINARY BASE64 option of the FOR XML clause [\[MSDN-SSDBEng\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

## 2.1.2.41 [ISO/IEC9075-14:2011] X059, Advanced table mapping: hex encoding of binary strings

V0076:

The specification states the following:

Subclause 9.14, "Mapping an SQL schema to an XML document and an XML Schema document":

Without Feature X059, "Advanced table mapping: hex encoding of binary strings", a conforming application shall not invoke this Subclause of this part of this International Standard with ENCODING set to indicate that binary strings are to be encoded using hex.

Subclause 9.17, "Mapping an SQL catalog to an XML document and an XML Schema document":

Without Feature X059, "Advanced table mapping: hex encoding of binary strings", a conforming application shall not invoke this Subclause of this part of this International Standard with ENCODING set to indicate that binary strings are to be encoded using hex.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.42 [ISO/IEC9075-14:2011] X060, XMLParse: Character string input and CONTENT option**

V0077:

The specification states the following:

Subclause 6.16, "<XML parse>":

Without Feature X060, "XMLParse: Character string input and CONTENT option", in conforming SQL language, the declared type of the <string value expression> immediately contained in <XML parse> shall not be a character string type and <XML parse> shall not immediately contain a <document or content> that is CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. These functions support parsing from string (NVARCHAR) to XML(CONTENT).

This variation pertains to XML parser functionality in SQL Server.

#### **2.1.2.43 [ISO/IEC9075-14:2011] X061, XMLParse: Character string input and DOCUMENT option**

V0078:

The specification states the following:

Subclause 6.16, "<XML parse>":

Without Feature X061, "XMLParse: Character string input and DOCUMENT option", in conforming SQL language, the declared type of the <string value expression> immediately contained in <XML parse> shall not be a character string type and <XML parse> shall not immediately contain a <document or content> that is DOCUMENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. These functions support parsing from string (NVARCHAR) to XML(DOCUMENT).

This variation pertains to XML parser functionality in SQL Server.

#### **2.1.2.44 [ISO/IEC9075-14:2011] X065, XMLParse: BLOB input and CONTENT option**

V0079:

The specification states the following:

Subclause 6.16, "<XML parse>":

Without Feature X065, "XMLParse: BLOB input and CONTENT option", in conforming SQL language, the declared type of the <string value expression> immediately contained in <XML parse> shall not be a binary string type and <XML parse> shall not immediately contain a <document or content> that is CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. These functions support parsing from BLOB (VARBINARY) to XML(CONTENT).

This variation pertains to XML parser functionality in SQL Server.

### **2.1.2.45 [ISO/IEC9075-14:2011] X066, XMLParse: BLOB input and DOCUMENT option**

V0080:

The specification states the following:

Subclause 6.16, "<XML parse>":

Without Feature X066, "XMLParse: BLOB input and DOCUMENT option", in conforming SQL language, the declared type of the <string value expression> immediately contained in <XML parse> shall not be a binary string type and <XML parse> shall not immediately contain a <document or content> that is DOCUMENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. The CAST and CONVERT functions support parsing from BLOB (VARBINARY) to XML(DOCUMENT).

This variation pertains to XML parser functionality in SQL Server.

### **2.1.2.46 [ISO/IEC9075-14:2011] X068, XMLSerialize: BOM**

V0081:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X068, "XMLSerialize: BOM", in conforming SQL language, <XML serialize bom> shall not be specified.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. However, Transact-SQL adds a byte-order mark implicitly when casting or converting to a BLOB (VARBINARY).

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.47 [ISO/IEC9075-14:2011] X069, XMLSerialize: INDENT

V0082:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X069, "XMLSerialize: INDENT", in conforming SQL language, <XML serialize indent> shall not be specified.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.48 [ISO/IEC9075-14:2011] X070, XMLSerialize: Character string serialization and CONTENT option

V0083:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X070, "XMLSerialize: character string serialization and CONTENT option", conforming SQL language shall not contain an <XML character string serialization> that immediately contains a <document or content> that is CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. The CAST and CONVERT functions support serializing from XML(CONTENT) to string (either as NVARCHAR or VARCHAR).

This variation pertains to XML parser functionality in SQL Server.

### 2.1.2.49 [ISO/IEC9075-14:2011] X071, XMLSerialize: Character string serialization and DOCUMENT option

V0084:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X071, "XMLSerialize: character string serialization and DOCUMENT option", conforming SQL language shall not contain an <XML character string serialization> that immediately contains a <document or content> that is DOCUMENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. The CAST and CONVERT functions support serializing from XML(DOCUMENT) to string (either as NVARCHAR or VARCHAR).

This variation pertains to XML parser functionality in SQL Server.

#### **2.1.2.50 [ISO/IEC9075-14:2011] X072, XMLSerialize: Character string serialization**

V0085:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X072, "XMLSerialize: character string serialization", conforming SQL language shall not contain an <XML character string serialization>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. The CAST and CONVERT functions support serializing both from XML(CONTENT) and from XML(DOCUMENT) to string (either as NVARCHAR or VARCHAR).

This variation pertains to XML parser functionality in SQL Server.

#### **2.1.2.51 [ISO/IEC9075-14:2011] X073, XMLSerialize: BLOB serialization and CONTENT option**

V0086:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X073, "XMLSerialize: BLOB serialization and CONTENT option", conforming SQL language shall not contain an <XML binary string serialization> that immediately contains a <document or content> that is CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. The CAST and CONVERT functions support serializing from XML(CONTENT) to BLOB (VARBINARY). A byte order mark will be added.

This variation pertains to XML parser functionality in SQL Server.

#### **2.1.2.52 [ISO/IEC9075-14:2011] X074, XMLSerialize: BLOB serialization and DOCUMENT option**

V0087:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X074, "XMLSerialize: BLOB serialization and DOCUMENT option", conforming SQL language shall not contain an <XML binary string serialization> that immediately contains a <document or content> that is DOCUMENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. The CAST and CONVERT functions support serializing from XML(DOCUMENT) to BLOB (VARBINARY). A byte-order mark will be added.

This variation pertains to XML parser functionality in SQL Server.

### **2.1.2.53 [ISO/IEC9075-14:2011] X075, XMLSerialize: BLOB serialization**

V0088:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X075, "XMLSerialize: BLOB serialization", conforming SQL language shall not contain an <XML binary string serialization>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CAST and CONVERT functions [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality. The CAST and CONVERT functions support serializing both from XML(CONTENT) and from XML(DOCUMENT) to string (either as NVARCHAR or VARCHAR). A byte order mark will be added.

This variation pertains to XML parser functionality in SQL Server.

### **2.1.2.54 [ISO/IEC9075-14:2011] X076, XMLSerialize: VERSION**

V0089:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X076, "XMLSerialize: VERSION", in conforming SQL language, <XML character string serialization> shall not contain VERSION.

Without Feature X076, "XMLSerialize: VERSION", in conforming SQL language, <XML binary string serialization> shall not contain VERSION.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.55 [ISO/IEC9075-14:2011] X077, XMLSerialize: explicit ENCODING option**

V0090:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X077, "XMLSerialize: explicit ENCODING option", conforming SQL language shall not contain an <XML binary string serialization> that contains ENCODING.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.56 [ISO/IEC9075-14:2011] X078, XMLSerialize: explicit XML declaration**

V0091:

The specification states the following:

Subclause 6.8, "<string value function>":

Without Feature X078, "XMLSerialize: explicit XML declaration", in conforming SQL language, <XML character string serialization> shall not contain XMLDECLARATION.

Without Feature X078, "XMLSerialize: explicit XML declaration", in conforming SQL language, <XML binary string serialization> shall not contain XMLDECLARATION.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.57 [ISO/IEC9075-14:2011] X084, XML namespace declarations in compound statements**

V0092:

The specification states the following:

Subclause 15.1, "<compound statement>":

Without Feature X084, "XML namespace declarations in compound statements", in conforming SQL language, a <compound statement> shall not immediately contain an <XML lexically scoped options> that contains an <XML namespace declaration>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML Namespaces [\[XMLNS3\]](#) functionality in SQL Server.

### **2.1.2.58 [ISO/IEC9075-14:2011] X085, Predefined namespace prefixes**

V0093:

The specification states the following:

Subclause 6.14, "<XML element>":

Without Feature X085, "Predefined namespace prefixes", conforming SQL language shall not contain an <XML element name> E that has an XML QName prefix that is not equivalent to an <XML namespace prefix> contained in one or more <XML namespace declaration>s that are the scope of the <XML element> that contains E.

Without Feature X085, "Predefined namespace prefixes", conforming SQL language shall not contain an explicit or implicit <XML attribute name> A that has an XML QName prefix other than 'xml' that is not equivalent to an <XML namespace prefix> contained in one or more <XML namespace declaration>s that are the scope of the <XML element> that contains A.

Subclause 6.15, "<XML forest>":

Without Feature X085, "Predefined namespace prefixes", conforming SQL language shall not contain an explicit or implicit <forest element name> F that has an XML QName prefix that is not equivalent to an <XML namespace prefix> contained in one or more <XML namespace declaration>s that are the scope of the <XML forest> that contains F.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML Namespaces [\[XMLNS3\]](#) functionality in SQL Server.

### **2.1.2.59 [ISO/IEC9075-14:2011] X086, XML namespace declarations in XMLTable**

V0094:

The specification states the following:

Subclause 7.1, "<table reference>":

Without Feature X086, "XML namespace declarations in XMLTable", in conforming SQL language, an <XML table> shall not contain an <XML namespace declaration>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML Namespaces [\[XMLNS3\]](#) functionality in SQL Server.



### 2.1.2.60 [ISO/IEC9075-14:2011] X090, XML document predicate

V0095:

The specification states the following:

Subclause 8.3, "<XML document predicate>":

Without Feature X090, "XML document predicate", conforming SQL language shall not contain <XML document predicate>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.61 [ISO/IEC9075-14:2011] X091, XML content predicate

V0096:

The specification states the following:

Subclause 8.2, "<XML content predicate>":

Without Feature X091, "XML content predicate", conforming SQL language shall not contain <XML content predicate>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.62 [ISO/IEC9075-14:2011] X096, XMLExists

V0097:

The specification states the following:

Subclause 8.4, "<XML exists predicate>":

Without Feature X096, "XMLExists", conforming SQL language shall not contain <XML exists predicate>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the **.exist()** method of the **xml** data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.63 [ISO/IEC9075-14:2011] X100, Host language support for XML: CONTENT option

V0098:

The specification states the following:

Subclause 16.1, "<set XML option statement>":

Without Feature X100, "Host language support for XML: CONTENT option", conforming SQL language shall not contain a <set XML option statement> that contains CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the CONTENT argument of the **xml** data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.64 [ISO/IEC9075-14:2011] X101, Host language support for XML: DOCUMENT option

V0099:

The specification states the following:

Subclause 12.8, "<SQL-invoked routine>":

Without Feature X101, "Host language support for XML: DOCUMENT option", conforming SQL language shall not contain a <document or content> that is DOCUMENT.

Subclause 16.1, "<set XML option statement>":

Without Feature X101, "Host language support for XML: DOCUMENT option", conforming SQL language shall not contain a <set XML option statement> that contains DOCUMENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the DOCUMENT argument of the **xml** data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.65 [ISO/IEC9075-14:2011] X110, Host language support for XML: VARCHAR mapping

V0100:

The specification states the following:

Subclause 12.8, "<SQL-invoked routine>":

Without Feature X110, "Host language support for XML: VARCHAR mapping", conforming SQL language shall not contain a <string type option> that contains CHARACTER VARYING, CHAR VARYING, or VARCHAR.

Subclause 18.3, "<embedded SQL C program>":

Without Feature X110, "Host language support for XML: VARCHAR mapping", conforming SQL language shall not contain an <C XML VARCHAR variable>.

Subclause 18.7, "<embedded SQL PL/I program>":

Without Feature X110, "Host language support for XML: VARCHAR mapping", conforming SQL language shall not contain an <PL/I XML VARCHAR variable>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. PL/I and C language support is not available.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.66 [ISO/IEC9075-14:2011] X111, Host language support for XML: CLOB mapping**

V0101:

The specification states the following:

Subclause 12.8, "<SQL-invoked routine>":

Without Feature X111, "Host language support for XML: CLOB mapping", conforming SQL language shall not contain a <string type option> that contains CHARACTER LARGE OBJECT, CHAR LARGE OBJECT, or CLOB.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.67 [ISO/IEC9075-14:2011] X112, Host language support for XML: BLOB mapping**

V0102:

The specification states the following:

Subclause 18.3, "<embedded SQL C program>":

Without Feature X112, "Host language support for XML: BLOB mapping", conforming SQL language shall not contain a <C XML BLOB variable>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.68 [ISO/IEC9075-14:2011] X113, Host language support for XML: STRIP WHITESPACE option**

V0103:

The specification states the following:

Subclause 18.3, "<embedded SQL C program>":

Without Feature X113, "Host language support for XML: STRIP WHITESPACE option", in conforming SQL language, <C XML CLOB variable> shall not immediately contain an <XML whitespace option> that is STRIP WHITESPACE.

Without Feature X113, "Host language support for XML: STRIP WHITESPACE option", in conforming SQL language, <C XML BLOB variable> shall not immediately contain an <XML whitespace option> that is STRIP WHITESPACE.

Without Feature X113, "Host language support for XML: STRIP WHITESPACE option", in conforming SQL language, <C XML VARCHAR variable> shall not immediately contain an <XML whitespace option> that is STRIP WHITESPACE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.69 [ISO/IEC9075-14:2011] X114, Host language support for XML: PRESERVE WHITESPACE option**

V0104:

The specification states the following:

Subclause 18.3, "<embedded SQL C program>":

Without Feature X114, "Host language support for XML: PRESERVE WHITESPACE option", in conforming SQL language, <C XML CLOB variable> shall not immediately contain an <XML whitespace option> that is PRESERVE WHITESPACE.

Without Feature X114, "Host language support for XML: PRESERVE WHITESPACE option", in conforming SQL language, <C XML BLOB variable> shall not immediately contain an <XML whitespace option> that is PRESERVE WHITESPACE.

Without Feature X114, "Host language support for XML: PRESERVE WHITESPACE option", in conforming SQL language, <C XML VARCHAR variable> shall not immediately contain an <XML whitespace option> that is PRESERVE WHITESPACE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.70 [ISO/IEC9075-14:2011] X120, XML parameters in SQL routines**

V0105:

The specification states the following:

Subclause 12.8, "<SQL-invoked routine>":

Without Feature X120, "XML parameters in SQL routines", conforming SQL language shall not contain an <SQL-invoked routine> that simply contains a <language clause> that contains SQL and that simply contains a <parameter type> or a <returns data type> that contains a <data type> that is based on either an XML type or a distinct type whose source type is an XML type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. The **xml** data type cannot be used as a parameter to any scalar, built-in functions other than ISNULL, COALESCE, and DATALENGTH.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.71 [ISO/IEC9075-14:2011] X121, XML parameters in external routines**

V0106:

The specification states the following:

Subclause 12.8, "<SQL-invoked routine>":

Without Feature X121, "XML parameters in external routines", conforming SQL language shall not contain an <SQL-invoked routine> that simply contains a <language clause> that contains ADA, C, COBOL, FORTRAN, MUMPS, PASCAL, or PLI and that simply contains a <parameter type> or a <returns data type> that contains a <data type> that is based on either an XML type or a distinct type whose source type is an XML type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.72 [ISO/IEC9075-14:2011] X131, Query-level XMLBINARY clause**

V0107:

The specification states the following:

Subclause 7.2, "<query expression>":

Without Feature X131, "Query-level XMLBINARY clause", in conforming SQL language, a <with clause> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.73 [ISO/IEC9075-14:2011] X132, XMLBINARY clause in DML**

V0108:

The specification states the following:

Subclause 14.3, "<delete statement: searched>":

Without Feature X132, "XMLBINARY clause in DML", in conforming SQL language, a <delete statement: searched> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Subclause 14.4, "<insert statement>":

Without Feature X132, "XMLBINARY clause in DML", in conforming SQL language, an <insert statement> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Subclause 14.5, "<merge statement>":

Without Feature X132, "XMLBINARY clause in DML", in conforming SQL language, a <merge statement> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Subclause 14.6, "<update statement: positioned>":

Without Feature X132, "XMLBINARY clause in DML", in conforming SQL language, an <update statement: positioned> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Subclause 14.7, "<update statement: searched>":

Without Feature X132, "XMLBINARY clause in DML", in conforming SQL language, an <update statement: searched> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.74 [ISO/IEC9075-14:2011] X133, XMLBINARY clause in DDL**

V0109:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X133, "XMLBINARY clause in DDL", in conforming SQL language, a <generation expression> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Subclause 12.2, "<check constraint definition>":

Without Feature X133, "XMLBINARY clause in DDL", in conforming SQL language, a <check constraint definition> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Subclause 12.5, "<assertion definition>":

Without Feature X133, "XMLBINARY clause in DDL", in conforming SQL language, an <assertion definition> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.75 [ISO/IEC9075-14:2011] X134, XMLBINARY clause in compound statements**

V0110:

The specification states the following:

Subclause 15.1, "<compound statement>":

Without Feature X134, "XMLBINARY clause in compound statements", in conforming SQL language, a <compound statement> shall not immediately contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.76 [ISO/IEC9075-14:2011] X135, XMLBINARY clause in subqueries**

V0111:

The specification states the following:

Subclause 7.2, "<query expression>":

Without Feature X135, "XMLBINARY clause in subqueries", in conforming SQL language, a <query expression> that is contained in another <query expression> shall not contain an <XML lexically scoped options> that contains an <XML binary encoding>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.77 [ISO/IEC9075-14:2011] X141, IS VALID predicate: data-driven case**

V0112:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X141, "IS VALID predicate: data-driven case", conforming SQL language shall not contain an <XML valid predicate> that does not contain <XML valid according to clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.78 [ISO/IEC9075-14:2011] X142, IS VALID predicate: ACCORDING TO clause**

V0113:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X142, "IS VALID predicate: ACCORDING TO clause", conforming SQL language shall not contain an <XML valid predicate> that contains <XML valid according to clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.79 [ISO/IEC9075-14:2011] X143, IS VALID predicate: ELEMENT clause**

V0114:

The specification states the following:



Subclause 8.5, "<XML valid predicate>":

Without Feature X143, "IS VALID predicate: ELEMENT clause", conforming SQL language shall not contain an <XML valid predicate> that contains <XML valid element clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.80 [ISO/IEC9075-14:2011] X144, IS VALID predicate: schema location**

V0115:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X144, "IS VALID predicate: schema location", conforming SQL language shall not contain an <XML valid predicate> that contains <XML valid schema location>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.81 [ISO/IEC9075-14:2011] X145, IS VALID predicate outside check constraints**

V0116:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X145, "IS VALID predicate outside check constraints", conforming SQL language shall not contain an <XML valid predicate> that is not directly contained in the <search condition> of a <check constraint definition>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.82 [ISO/IEC9075-14:2011] X151, IS VALID predicate with DOCUMENT option**

V0117:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X151, "IS VALID predicate: with DOCUMENT option", conforming SQL language shall not contain an <XML valid predicate> that immediately contains a <document or content or sequence> that is DOCUMENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.83 [ISO/IEC9075-14:2011] X152, IS VALID predicate with CONTENT option**

V0118:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X152, "IS VALID predicate: with CONTENT option", conforming SQL language shall not contain an <XML valid predicate> that immediately contains a <document or content or sequence> that is CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.84 [ISO/IEC9075-14:2011] X153, IS VALID predicate with SEQUENCE option**

V0119:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X153, "IS VALID predicate: with SEQUENCE option", conforming SQL language shall not contain an <XML valid predicate> that immediately contains a <document or content or sequence> that is SEQUENCE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.85 [ISO/IEC9075-14:2011] X155, IS VALID predicate: NAMESPACE without ELEMENT clause**

V0120:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X155, "IS VALID predicate: NAMESPACE without ELEMENT clause", conforming SQL language shall not contain an <XML valid predicate> that contains an <XML valid element clause> that does not contain an <XML valid element name specification>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML Namespaces [\[XMLNS3\]](#) functionality in SQL Server.

### **2.1.2.86 [ISO/IEC9075-14:2011] X157, IS VALID predicate: NO NAMESPACE with ELEMENT clause**

V0121:

The specification states the following:

Subclause 8.5, "<XML valid predicate>":

Without Feature X157, "IS VALID predicate: NO NAMESPACE with ELEMENT clause", conforming SQL language shall not contain an <XML valid predicate> that contains an <XML valid element namespace specification> that contains NO NAMESPACE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML Namespaces [\[XMLNS3\]](#) functionality in SQL Server.

### **2.1.2.87 [ISO/IEC9075-14:2011] X160, Basic Information Schema for registered XML Schemas**

V0122:

The specification states the following:

Subclause 20.3, "ATTRIBUTES view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.ATTRIBUTES.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.ATTRIBUTES.XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.ATTRIBUTES.XML\_SCHEMA\_NAME.

Subclause 20.4, "COLUMNS view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.COLUMNS.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.COLUMNS.XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.COLUMNS.XML\_SCHEMA\_NAME.

Subclause 20.5, "DOMAINS view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.DOMAINS.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.DOMAINS.XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.DOMAINS.XML\_SCHEMA\_NAME.

Subclause 20.6, "ELEMENT\_TYPES view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.ELEMENT\_TYPES.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.ELEMENT\_TYPES.XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.ELEMENT\_TYPES.XML\_SCHEMA\_NAME.

Subclause 20.7, "FIELDS view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.FIELDS.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.FIELDS.XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.FIELDS.XML\_SCHEMA\_NAME.

Subclause 20.8, "METHOD\_SPECIFICATION\_PARAMETERS view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.XML\_SCHEMA\_NAME.

Subclause 20.9, "METHOD\_SPECIFICATIONS view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.METHOD\_SPECIFICATIONS.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.METHOD\_SPECIFICATIONS.XML\_SCHEMA\_SCHEMA, INFORMATION\_SCHEMA.METHOD\_SPECIFICATIONS.XML\_SCHEMA\_NAME, INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.RESULT\_CAST\_XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.RESULT\_CAST\_XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.RESULT\_CAST\_XML\_SCHEMA\_NAME.

Subclause 20.10, "PARAMETERS view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference reference INFORMATION\_SCHEMA.PARAMETERS.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.PARAMETERS.XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.PARAMETERS.XML\_SCHEMA\_NAME.

Subclause 20.11, "ROUTINES view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.ROUTINES.XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.ROUTINES.XML\_SCHEMA\_SCHEMA, INFORMATION\_SCHEMA.ROUTINES.XML\_SCHEMA\_NAME, INFORMATION\_SCHEMA.ROUTINES.RESULT\_CAST\_XML\_SCHEMA\_CATALOG, INFORMATION\_SCHEMA.ROUTINES.RESULT\_CAST\_XML\_SCHEMA\_SCHEMA, or INFORMATION\_SCHEMA.ROUTINES.RESULT\_CAST\_XML\_SCHEMA\_NAME.

Subclause 20.14, "XML\_SCHEMAS view":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.XML\_SCHEMAS.

Subclause 20.15, "Short name views":

Without Feature X160, "Basic Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.XML\_SCHEMAS\_S. 92)

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. However, it does support equivalent catalog views. See "XML Schemas (XML Type System) Catalog Views (Transact-SQL)" in [\[MSDN-Transact-SQLRef\]](#) for more information.

See [\[ISO/IEC9075-1:2011\]](#) for the definition of "view".

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.88 [ISO/IEC9075-14:2011] X161, Advanced Information Schema for registered XML Schemas**

V0123:

The specification states the following:

Subclause 20.3, "ATTRIBUTES view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.ATTRIBUTES.XML\_SCHEMA\_NAMESPACE or INFORMATION\_SCHEMA.ATTRIBUTES.XML\_SCHEMA\_ELEMENT.

Subclause 20.4, "COLUMNS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.COLUMNS.XML\_SCHEMA\_NAMESPACE or INFORMATION\_SCHEMA.COLUMNS.XML\_SCHEMA\_ELEMENT.

Subclause 20.5, "DOMAINS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.DOMAINS.XML\_SCHEMA\_NAMESPACE or INFORMATION\_SCHEMA.DOMAINS.XML\_SCHEMA\_ELEMENT.

Subclause 20.6, "ELEMENT\_TYPES view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.ELEMENT\_TYPES.XML\_SCHEMA\_NAMESPACE or INFORMATION\_SCHEMA.ELEMENT\_TYPES.XML\_SCHEMA\_ELEMENT.

Subclause 20.7, "FIELDS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference INFORMATION\_SCHEMA.FIELDS.XML\_SCHEMA\_NAMESPACE or INFORMATION\_SCHEMA.FIELDS.XML\_SCHEMA\_ELEMENT.

Subclause 20.8, "METHOD\_SPECIFICATION\_PARAMETERS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference

INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.XML\_SCHEMA\_NAMESPACE or  
INFORMATION\_SCHEMA.METHOD\_SPECIFICATION\_PARAMETERS.XML\_SCHEMA\_ELEMENT.

Subclause 20.9, "METHOD\_SPECIFICATIONS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.METHOD\_SPECIFICATIONS.XML\_SCHEMA\_NAMESPACE,  
INFORMATION\_SCHEMA.METHOD\_SPECIFICATIONS.XML\_SCHEMA\_ELEMENT,  
INFORMATION\_SCHEMA.METHOD\_SPECIFICATIONS.RESULT\_CAST\_XML\_SCHEMA\_NAMESPACE, or  
INFORMATION\_SCHEMA.METHOD\_SPECIFICATIONS.RESULT\_CAST\_XML\_SCHEMA\_ELEMENT.

Subclause 20.10, "PARAMETERS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.PARAMETERS.XML\_SCHEMA\_NAMESPACE or  
INFORMATION\_SCHEMA.PARAMETERS.XML\_SCHEMA\_ELEMENT.

Subclause 20.11, "ROUTINES view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.ROUTINES.XML\_SCHEMA\_NAMESPACE,  
INFORMATION\_SCHEMA.ROUTINES.XML\_SCHEMA\_ELEMENT,  
INFORMATION\_SCHEMA.ROUTINES.RESULT\_CAST\_XML\_SCHEMA\_NAMESPACE, or  
INFORMATION\_SCHEMA.ROUTINES.RESULT\_CAST\_XML\_SCHEMA\_ELEMENT.

Subclause 20.12, "XML\_SCHEMA\_ELEMENTS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.XML\_SCHEMA\_ELEMENTS.

Subclause 20.13, "XML\_SCHEMA\_NAMESPACES view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.XML\_SCHEMA\_NAMESPACES.

Subclause 20.14, "XML\_SCHEMAS view":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.XML\_SCHEMAS.SCHEMA\_IS\_DETERMINISTIC.

Subclause 20.15, "Short name views":

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.XML\_SCH\_ELEMENTS\_S.

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.XML\_SCH\_NAMESPACES\_S.

Without Feature X161, "Advanced Information Schema for registered XML Schemas", conforming SQL language shall not reference  
INFORMATION\_SCHEMA.XML\_SCHEMAS\_S.XML\_SCHEMA\_IS\_DET.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.89 [ISO/IEC9075-14:2011] X170, XML null handling options

V0124:

The specification states the following:

Subclause 6.14, "<XML element>":

Without Feature X170, "XML null handling options", conforming SQL language shall not specify <XML content option>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.90 [ISO/IEC9075-14:2011] X171, NIL ON NO CONTENT option

V0125:

The specification states the following:

Subclause 6.14, "<XML element>":

Without Feature X171, "NIL ON NO CONTENT option", conforming SQL language shall not specify NIL ON NO CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.91 [ISO/IEC9075-14:2011] X181, XML(DOCUMENT(UNTYPED)) type

V0126:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL partially supports this feature. Transact-SQL supports the primary **xml** type modifier of DOCUMENT, but does not support the secondary **xml** type modifier of UNTYPED.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.92 [ISO/IEC9075-14:2011] X182, XML(DOCUMENT(ANY)) type

V0127:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of DOCUMENT, but does not support the secondary **xml** type modifier of ANY.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.93 [ISO/IEC9075-14:2011] X190, XML(SEQUENCE) type

V0128:

The specification states the following:

Subclause 6.1, "<data type>":

Without Feature X190, "XML(SEQUENCE) type", conforming SQL language shall not contain an <XML type> whose <XML type modifier> is SEQUENCE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifiers of CONTENT and DOCUMENT, but does not support the primary **xml** type modifier of SEQUENCE.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.94 [ISO/IEC9075-14:2011] X191, XML(DOCUMENT(XMLSCHEMA)) type

V0129:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:



Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of DOCUMENT, but does not support the secondary **xml** type modifier of XMLSCHEMA. Transact-SQL requires a schema collection for this data type.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.95 [ISO/IEC9075-14:2011] X192, XML(CONTENT(XMLSCHEMA)) type**

V0130:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of CONTENT, but does not support the secondary **xml** type modifier of XMLSCHEMA. Transact-SQL requires a schema collection for this data type.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.96 [ISO/IEC9075-14:2011] X200, XMLQuery**

V0131:

The specification states the following:

Subclause 6.18, "<XML query>":

Without Feature X200, "XMLQuery", conforming SQL language shall not contain an <XML query>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the **.query()** method of the **xml** data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.97 [ISO/IEC9075-14:2011] X201, XMLQuery: RETURNING CONTENT**

V0132:

The specification states the following:

Subclause 6.18, "<XML query>":

Without Feature X201, "XMLQuery: RETURNING CONTENT", conforming SQL language shall not contain an <XML query> that contains RETURNING CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.98 [ISO/IEC9075-14:2011] X202, XMLQuery: RETURNING SEQUENCE**

V0133:

The specification states the following:

Subclause 6.18, "<XML query>":

Without Feature X202, "XMLQuery: RETURNING SEQUENCE", conforming SQL language shall not contain an <XML query> that contains RETURNING SEQUENCE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.99 [ISO/IEC9075-14:2011] X203, XMLQuery: passing a context item**

V0134:

The specification states the following:

Subclause 6.18, "<XML query>":

Without Feature X203, "XMLQuery: passing a context item", in conforming SQL language, an <XML query> shall not contain an <XML query context item>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the **xml** data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.100 [ISO/IEC9075-14:2011] X204, XMLQuery: initializing an XQuery variable**

V0135:

The specification states the following:

Subclause 6.18, "<XML query>":

Without Feature X204, "XMLQuery: initializing an XQuery variable", in conforming SQL language, an <XML query> shall not contain an <XML query variable>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the **xml** data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.101 [ISO/IEC9075-14:2011] X205, XMLQuery: EMPTY ON EMPTY option**

V0136:

The specification states the following:

Subclause 6.18, "<XML query>":

Without Feature X205, "XMLQuery: EMPTY ON EMPTY option", conforming SQL language shall not contain an <XML query> that contains an <XML query empty handling option> that specifies EMPTY ON EMPTY.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.102 [ISO/IEC9075-14:2011] X206, XMLQuery: NULL ON EMPTY option**

V0137:

The specification states the following:

Subclause 6.18, "<XML query>":

Without Feature X206, "XMLQuery: NULL ON EMPTY option", conforming SQL language shall not contain an <XML query> that contains an <XML query empty handling option> that specifies NULL ON EMPTY.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.103 [ISO/IEC9075-14:2011] X211, XML 1.1 support**

V0138:

The specification states the following:

Subclause 6.14, "<XML element>":

Without Feature X211, "XML 1.1 support", in conforming SQL language, an <XML element name> shall be an XML 1.0 QName.

Without Feature X211, "XML 1.1 support", in conforming SQL language, an <XML attribute name> shall be an XML 1.0 QName.

Subclause 6.15, "<XML forest>":

Without Feature X211, "XML 1.1 support", in conforming SQL language, a <forest element name> shall be an XML 1.0 QName.

Subclause 6.17, "<XML PI>":

Without Feature X211, "XML 1.1 support", in conforming SQL language, an <identifier> contained in an <XML PI target>, when mapped to Unicode, shall be an XML 1.0 NCName.

Note -- The set of XML 1.0 NCNames is a proper subset of the set of XML 1.1 NCNames. That is, all XML 1.0 NCNames are also XML 1.1 NCNames, but not all XML 1.1 NCNames are also XML 1.0 NCNames.

Subclause 6.18, "<XML query>":

Without Feature X211, "XML 1.1 support", in conforming SQL language, the value of the <XQuery expression> shall be an XQuery expression with XML 1.0 lexical rules.

Without Feature X211, "XML 1.1 support", in conforming SQL language, the <identifier> contained in an <XML query variable> shall be an XML 1.0 NCName.

Subclause 11.3, "<XML lexically scoped options>":

Without Feature X211, "XML 1.1 support", in conforming SQL language, each <XML namespace prefix> shall be an XML 1.0 NCName.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.104 [ISO/IEC9075-14:2011] X221, XML passing mechanism BY VALUE**

V0139:

The specification states the following:

Subclause 11.5, "<XML passing mechanism>":

Without Feature X221, "XML passing mechanism BY VALUE", conforming SQL language shall not contain an <XML passing mechanism> that is BY VALUE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. The **xml** data type cannot be used as a parameter to any scalar, built-in functions other than ISNULL, COALESCE, and DATALENGTH.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.105 [ISO/IEC9075-14:2011] X222, XML passing mechanism BY REF**

V0140:

The specification states the following:

Subclause 11.5, "<XML passing mechanism>":

Without Feature X222, "XML passing mechanism BY REF", conforming SQL language shall not contain an <XML passing mechanism> that is BY REF.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. The **xml** data type cannot be used as a parameter to any scalar, built-in functions other than ISNULL, COALESCE, and DATALENGTH.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.106 [ISO/IEC9075-14:2011] X231, XML(CONTENT(UNTYPED)) type**

V0141:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of CONTENT, but does not support the secondary **xml** type modifier of UNTYPED, which is implied by Transact-SQL.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.107 [ISO/IEC9075-14:2011] X232, XML(CONTENT(ANY)) type**

V0142:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of CONTENT, but does not support the secondary **xml** type modifier of ANY.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.108 [ISO/IEC9075-14:2011] X241, RETURNING CONTENT in XML publishing**

V0143:

The specification states the following:

Subclause 6.11, "<XML comment>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML comment> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Subclause 6.12, "<XML concatenation>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML concatenation> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Subclause 6.13, "<XML document>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML document> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Subclause 6.14, "<XML element>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML element> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Subclause 6.15, "<XML forest>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML forest> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Subclause 6.17, "<XML PI>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML PI> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Subclause 6.19, "<XML text>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML text> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Subclause 11.2, "<aggregate function>":

Without Feature X241, "RETURNING CONTENT in XML publishing", in conforming SQL language, an <XML aggregate> shall not specify an <XML returning clause> that is RETURNING CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.109 [ISO/IEC9075-14:2011] X242, RETURNING SEQUENCE in XML publishing**

V0144:

The specification states the following:

Subclause 6.11, "<XML comment>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML comment> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Subclause 6.12, "<XML concatenation>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML concatenation> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Subclause 6.13, "<XML document>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML document> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Subclause 6.14, "<XML element>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML element> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Subclause 6.15, "<XML forest>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML forest> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Subclause 6.17, "<XML PI>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML PI> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Subclause 6.19, "<XML text>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML text> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Subclause 11.2, "<aggregate function>":

Without Feature X242, "RETURNING SEQUENCE in XML publishing", in conforming SQL language, an <XML aggregate> shall not specify an <XML returning clause> that is RETURNING SEQUENCE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.110 [ISO/IEC9075-14:2011] X251, Persistent XML values of XML(DOCUMENT(UNTYPED)) type

V0145:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X251, "Persistent XML values of XML(DOCUMENT(UNTYPED)) type", conforming SQL language shall not contain a <column definition> whose declared type is based on either the XML(DOCUMENT(UNTYPED)) type or a distinct type whose source type is the XML(DOCUMENT(UNTYPED)) type.

Subclause 12.4, "<view definition>":

Without Feature X251, "Persistent XML values of XML(DOCUMENT(UNTYPED)) type", conforming SQL language shall not contain a <view definition> that defines a column whose declared type is based on either the XML(DOCUMENT(UNTYPED)) type or a distinct type whose source type is the XML(DOCUMENT(UNTYPED)) type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of DOCUMENT, but does not support the secondary **xml** type modifier of UNTYPED.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.111 [ISO/IEC9075-14:2011] X252, Persistent XML values of XML(DOCUMENT(ANY)) type

V0146:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X252, "Persistent XML values of XML(DOCUMENT(ANY)) type", conforming SQL language shall not contain a <column definition> whose declared type is based on either the XML(DOCUMENT(ANY)) type or a distinct type whose source type is the XML(DOCUMENT(ANY)) type.

Subclause 12.4, "<view definition>":

Without Feature X252, "Persistent XML values of XML(DOCUMENT(ANY)) type", conforming SQL language shall not contain a <view definition> that defines a column whose declared type is based on either the XML(DOCUMENT(ANY)) type or a distinct type whose source type is the XML(DOCUMENT(ANY)) type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of DOCUMENT, but does not support the secondary **xml** type modifier of ANY.

This variation pertains to **xml** data type functionality in SQL Server.



### 2.1.2.112 [ISO/IEC9075-14:2011] X253, Persistent XML values of XML(CONTENT(UNTYPED)) type

V0147:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X253, "Persistent XML values of XML(CONTENT(UNTYPED)) type", conforming SQL language shall not contain a <column definition> whose declared type is based on either the XML(CONTENT(UNTYPED)) type or a distinct type whose source type is the XML(CONTENT(UNTYPED)) type.

Subclause 12.4, "<view definition>":

Without Feature X253, "Persistent XML values of XML(CONTENT(UNTYPED)) type", conforming SQL language shall not contain a <view definition> that defines a column whose declared type is based on either the XML(CONTENT(UNTYPED)) type or a distinct type whose source type is the XML(CONTENT(UNTYPED)) type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of CONTENT, but does not support the secondary **xml** type modifier of UNTYPED.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.113 [ISO/IEC9075-14:2011] X254, Persistent XML values of XML(CONTENT(ANY)) type

V0148:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X254, "Persistent XML values of XML(CONTENT(ANY)) type", conforming SQL language shall not contain a <column definition> whose declared type is based on either the XML(CONTENT(ANY)) type or a distinct type whose source type is the XML(CONTENT(ANY)) type.

Subclause 12.4, "<view definition>":

Without Feature X254, "Persistent XML values of XML(CONTENT(ANY)) type", conforming SQL language shall not contain a <view definition> that defines a column whose declared type is based on either the XML(CONTENT(ANY)) type or a distinct type whose source type is the XML(CONTENT(ANY)) type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of CONTENT, but does not support the secondary **xml** type modifier of ANY.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.114 [ISO/IEC9075-14:2011] X255, Persistent XML values of XML(SEQUENCE) type

V0149:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X255, "Persistent XML values of XML(SEQUENCE) type", conforming SQL language shall not contain a <column definition> whose declared type is based on either the XML(SEQUENCE) type or a distinct type whose source type is the XML(SEQUENCE) type.

Subclause 12.4, "<view definition>":

Without Feature X255, "Persistent XML values of XML(SEQUENCE) type", conforming SQL language shall not contain a <view definition> that defines a column whose declared type is based on either the XML(SEQUENCE) type or a distinct type whose source type is the XML(SEQUENCE) type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifiers of CONTENT and DOCUMENT, but does not support the primary **xml** type modifier of SEQUENCE.

This variation pertains to **xml** data type functionality in SQL Server.

### 2.1.2.115 [ISO/IEC9075-14:2011] X256, Persistent XML values of XML(DOCUMENT(XMLSCHEMA)) type

V0150:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X256, "Persistent XML values of XML(DOCUMENT(XMLSCHEMA)) type", conforming SQL language shall not contain a <column definition> whose declared type is based on either the XML(DOCUMENT(XMLSCHEMA)) type or a distinct type whose source type is the XML(DOCUMENT(XMLSCHEMA)) type.

Subclause 12.4, "<view definition>":

Without Feature X256, "Persistent XML values of XML(DOCUMENT(XMLSCHEMA)) type", conforming SQL language shall not contain a <view definition> that defines a column whose declared type is based on either the XML(DOCUMENT(XMLSCHEMA)) type or a distinct type whose source type is the XML(DOCUMENT(XMLSCHEMA)) type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of DOCUMENT, but does not support the secondary **xml** type modifier of XMLSCHEMA. Instead, XML Schema collection names are referred to directly.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.116 [ISO/IEC9075-14:2011] X257, Persistent XML values of XML(CONTENT(XMLSCHEMA)) type**

V0151:

The specification states the following:

Subclause 12.1, "<column definition>":

Without Feature X257, "Persistent XML values of XML(CONTENT(XMLSCHEMA)) type", conforming SQL language shall not contain a <column definition> whose declared type is based on either the XML(CONTENT(XMLSCHEMA)) type or a distinct type whose source type is the XML(CONTENT(XMLSCHEMA)) type.

Subclause 12.4, "<view definition>":

Without Feature X257, "Persistent XML values of XML(CONTENT(XMLSCHEMA)) type", conforming SQL language shall not contain a <view definition> that defines a column whose declared type is based on either the XML(CONTENT(XMLSCHEMA)) type or a distinct type whose source type is the XML(CONTENT(XMLSCHEMA)) type.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. Transact-SQL supports the primary **xml** type modifier of CONTENT, but does not support the secondary **xml** type modifier of XMLSCHEMA. Instead, XML Schema collection names are referred to directly.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.117 [ISO/IEC9075-14:2011] X260, XML type: ELEMENT clause**

V0152:

The specification states the following:

Subclause 6.1, "<data type>":

Without Feature X260, "XML type: ELEMENT clause", conforming SQL language shall not contain an <XML type> that contains <XML valid element clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.118 [ISO/IEC9075-14:2011] X261, XML type: NAMESPACE without ELEMENT clause**

V0153:

The specification states the following:

Subclause 6.1, "<data type>":

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>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type and XML Namespaces [\[XMLNS3\]](#) functionalities in SQL Server.

### **2.1.2.119 [ISO/IEC9075-14:2011] X263, XML type: NO NAMESPACE with ELEMENT clause**

V0154:

The specification states the following:

Subclause 6.1, "<data type>":

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.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type and XML Namespaces [\[XMLNS3\]](#) functionalities in SQL Server.

### **2.1.2.120 [ISO/IEC9075-14:2011] X264, XML type: schema location**

V0155:

The specification states the following:

Subclause 6.1, "<data type>":

Without Feature X264, "XML type: schema location", conforming SQL language shall not contain an <XML type> that contains <XML valid schema location>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to **xml** data type functionality in SQL Server.

### **2.1.2.121 [ISO/IEC9075-14:2011] X271, XMLValidate: data-driven case**

V0156:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X271, "XMLValidate: data-driven case", conforming SQL language shall not contain an <XML validate> that does not contain <XML valid according to clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.122 [ISO/IEC9075-14:2011] X272, XMLValidate: ACCORDING TO clause**

V0157:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X272, "XMLValidate: ACCORDING TO clause", conforming SQL language shall not contain an <XML validate> that contains <XML valid according to clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.123 [ISO/IEC9075-14:2011] X273, XMLValidate: ELEMENT clause**

V0158:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X273, "XMLValidate: ELEMENT clause", conforming SQL language shall not contain an <XML validate> that contains <XML valid element clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.124 [ISO/IEC9075-14:2011] X274, XMLValidate: schema location**

V0159:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X274, "XMLValidate: schema location", conforming SQL language shall not contain an <XML validate> that contains <XML valid schema location>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.125 [ISO/IEC9075-14:2011] X281, XMLValidate: with DOCUMENT option**

V0160:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X281, "XMLValidate with DOCUMENT option", conforming SQL language shall not contain an <XML validate> that immediately contains a <document or content or sequence> that is DOCUMENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.126 [ISO/IEC9075-14:2011] X282, XMLValidate with CONTENT option**

V0161:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X282, "XMLValidate with CONTENT option", conforming SQL language shall not contain an <XML validate> that immediately contains a <document or content or sequence> that is CONTENT.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

#### **2.1.2.127 [ISO/IEC9075-14:2011] X283, XMLValidate with SEQUENCE option**

V0162:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X283, "XMLValidate with SEQUENCE option", conforming SQL language shall not contain an <XML validate> that immediately contains a <document or content or sequence> that is SEQUENCE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### **2.1.2.128 [ISO/IEC9075-14:2011] X284, XMLValidate NAMESPACE without ELEMENT clause**

V0163:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X284, "XMLValidate: NAMESPACE without ELEMENT clause", conforming SQL language shall not contain an <XML validate> that contains an <XML valid element clause> that does not contain an <XML valid element name specification>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML Namespaces [\[XMLNS3\]](#) functionality in SQL Server.

### **2.1.2.129 [ISO/IEC9075-14:2011] X286, XMLValidate: NO NAMESPACE with ELEMENT clause**

V0164:

The specification states the following:

Subclause 6.20, "<XML validate>":

Without Feature X286, "XMLValidate: NO NAMESPACE with ELEMENT clause", conforming SQL language shall not contain an <XML validate> that contains an <XML valid element namespace specification> that contains NO NAMESPACE.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML Namespaces [\[XMLNS3\]](#) functionality in SQL Server.

### 2.1.2.130 [ISO/IEC9075-14:2011] X300, XMLTable

V0165:

The specification states the following:

Subclause 7.1, "<table reference>":

Without Feature X300, "XMLTable", conforming SQL language shall not contain <XML table>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature. See the **.nodes()** method of the **xml** data type [\[MSDN-Transact-SQLRef\]](#) for equivalent functionality.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.131 [ISO/IEC9075-14:2011] X301, XMLTable: derived column list option

V0166:

The specification states the following:

Subclause 7.1, "<table reference>":

Without Feature X301, "XMLTable: derived column list option", in conforming SQL language, a <table primary> that is an <XML table> shall not contain a <derived column list>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.132 [ISO/IEC9075-14:2011] X302, XMLTable: ordinality column option

V0167:

The specification states the following:

Subclause 7.1, "<table reference>":

Without Feature X302, "XMLTable: ordinality column option", in conforming SQL language, an <XML table> shall not contain an <XML table ordinality column definition>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.



### 2.1.2.133 [ISO/IEC9075-14:2011] X303, XMLTable: column default option

V0168:

The specification states the following:

Subclause 7.1, "<table reference>":

Without Feature X303, "XMLTable: column default option", in conforming SQL language, an <XML table regular column definition> shall not contain a <default clause>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.134 [ISO/IEC9075-14:2011] X304, XMLTable: passing a context item

V0169:

The specification states the following:

Subclause 7.1, "<table reference>":

Without Feature X304, "XMLTable: passing a context item", in conforming SQL language, an <XML table argument list> shall not contain an <XML query context item>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

### 2.1.2.135 [ISO/IEC9075-14:2011] X305, XMLTable: initializing an XQuery variable

V0170:

The specification states the following:

Subclause 7.1, "<table reference>":

Without Feature X305, "XMLTable: initializing an XQuery variable", in conforming SQL language, an <XML table argument list> shall not contain an <XML query variable>.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

## 2.1.2.136 [ISO/IEC9075-14:2011] X400, Name and identifier mapping

V0171:

The specification states the following:

Subclause 9.3, "Mapping XML Names to SQL <identifier>s":

Without Feature X400, "Name and identifier mapping", a conforming application shall not invoke this Subclause of this part of this International Standard.

Microsoft SQL Server 2008 R2 and Microsoft SQL Server 2012 vary as follows:

Transact-SQL does not support this feature.

This variation pertains to XML 1.0 [\[XML10/5\]](#) functionality in SQL Server.

## 2.2 Clarifications

Unless otherwise stated, the specified products conform to all SHOULD and RECOMMENDED behavior in [\[XML10/5\]](#), [\[XMLNS3\]](#), [\[XMLSCHEMA1/2\]](#), [\[XMLSCHEMA2/2\]](#), [\[ISO/IEC9075-14:2008\]](#), and [\[ISO/IEC9075-14:2011\]](#).

## 2.3 Error Handling

None.

## 2.4 Security

None.

### 3 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

## 4 Index

### C

change tracking ([section 3](#) 91, [section 3](#) 91)

### G

[Glossary](#) 8

### N

[notation](#) 11

### T

[Tracking changes](#) 91