[MS-TSQLISO02]:
SQL Server Transact-SQL ISO/IEC 9075-2 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.
- **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.
- **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.

Contents

1 Introduction

The SQL Server Transact-SQL ISO/IEC 9075-2 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 upon the SQL language specification (International Standard ISO/IEC 9075).

The SQL Server Transact-SQL ISO/IEC 9075-2 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 2: Foundation (SQL/Foundation) of both the [ISO/IEC9075-2:2008] and [ISO/IEC9075-2:2011] specifications. Unless otherwise stated, the specification excerpts are quoted from [ISO/IEC9075-2:2011]. Differences between the [ISO/IEC9075-2:2008] and [ISO/IEC9075-2:2011] 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. 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/IEC9075-2:2008] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 2: Foundation (SQL/Foundation)", INCITS/ISO/IEC 9075-2:2008, January 2009, http://webstore.ansi.org/RecordDetail.aspx?sku=INCITS%2fISO%2fIEC+9075-2-2008

Note There is a charge to download the specification.

[ISO/IEC9075-2:2011] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 2: Foundation (SQL/Foundation)", ISO/IEC 9075-2:2011, December 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53682

Note There is a charge to download the specification.

1.2.2 Informative References

[FIPS127] Federal Information Processing Standards Publication, "Database Language SQL", FIPS PUB 127, June 1993, http://www.itl.nist.gov/fipspubs/fip127-2.htm

[ISO/IEC9075-1:2008] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 1: Framework (SQL/Framework)", INCITS/ISO/IEC 9075-1:2008, January 2009, http://webstore.ansi.org/RecordDetail.aspx?sku=INCITS%2fISO%2fIEC+9075-1-2008

Note There is a charge to download the specification.

[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

Note There is a charge to download the specification.

[ISO/IEC9075-11:2008] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 11: Information and Definition Schemas (SQL/Schemata)", INCITS/ISO/IEC 9075-11:2008, January 2009, http://webstore.ansi.org/RecordDetail.aspx?sku=INCITS%2fISO%2fIEC+9075-11-2008

Note There is a charge to download the specification.

[ISO/IEC9075-11:2011] International Organization for Standardization, "Information technology -- Database languages -- SQL -- Part 11: Information and Definition Schemas (SQL/Schemata)", ISO/IEC 9075-11:2008, December 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53685

Note There is a charge to download the specification.

[SQL92] Digital Equipment Corporation, "Database Language SQL", July 1992, http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt

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

[MSDN-Add] Microsoft Corporation," + (Add) (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms178565.aspx

[MSDN-ALTERTRIGGER] Microsoft Corporation, "ALTER TRIGGER (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms176072.aspx

[MSDN-APPLY] Microsoft Corporation, "Using APPLY", http://msdn.microsoft.com/en-us/library/ms175156.aspx

[MSDN-BEGIN] Microsoft Corporation, "BEGIN TRANSACTION (Transact SQL)", http://msdn.microsoft.com/en-us/library/ms188929.aspx

[MSDN-binaryvarbinary] Microsoft Corporation, "binary and varbinary (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188362.aspx

[MSDN-CEILING] Microsoft Corporation, "CEILING (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms189818.aspx

[MSDN-CHARINDEX] Microsoft Corporation, "CHARINDEX (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms186323.aspx

[MSDN-charvarchar] Microsoft Corporation, "char and varchar (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms176089.aspx

[MSDN-CLRStorProc] Microsoft Corporation, "CLR Stored Procedures", http://msdn.microsoft.com/en-us/library/ms131094.aspx

[MSDN-CREATETRIGGER] Microsoft Corporation, "CREATE TRIGGER (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms189799.aspx

[MSDN-CURRENTTIMESTAMP] Microsoft Corporation, "CURRENT_TIMESTAMP (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188751.aspx

[MSDN-DATALENGTH] Microsoft Corporation, "DATALENGTH (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms173486.aspx

[MSDN-DATEPART] Microsoft Corporation, "DATEPART (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms174420.aspx

[MSDN-datetime2] Microsoft Corporation, "datetime2 (Transact-SQL)", http://msdn.microsoft.com/en-us/library/bb677335.aspx

[MSDN-datetimeoffset] Microsoft Corporation, "datetimeoffset (Transact-SQL)", http://msdn.microsoft.com/en-us/library/bb630289.aspx

[MSDN-EXCEPTINTERSECT] Microsoft Corporation, "EXCEPT and INTERSECT (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188055.aspx

[MSDN-EXECUTE] Microsoft Corporation, "EXECUTE (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188332.aspx

[MSDN-ExtdStorProc] Microsoft Corporation, "Extended Stored Procedures", http://msdn.microsoft.com/en-us/library/ms175200.aspx

[MSDN-GETDATE] Microsoft Corporation, "GETDATE (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188383.aspx

[MSDN-IDENTITY] Microsoft Corporation, "IDENTITY (Property) (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms186775.aspx

[MSDN-LEN] Microsoft Corporation, "LEN (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms190329.aspx

[MSDN-LOG] Microsoft Corporation, "LOG (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms190319.aspx

[MSDN-LTRIM] Microsoft Corporation, "LTRIM (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms177827.aspx

[MSDN-Modulo] Microsoft Corporation, "% (Modulo) (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms190279.aspx

[MSDN-MSODBC] Microsoft Corporation, "Microsoft Open Database Connectivity (ODBC)", http://msdn.microsoft.com/en-us/library/ms710252(VS.85).aspx

[MSDN-NCHAR] Microsoft Corporation, "NCHAR (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms182673.aspx

[MSDN-ntexttextimage] Microsoft Corporation, "ntext, text, and image (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms187993.aspx

[MSDN-RTRIM] Microsoft Corporation, "RTRIM (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms178660.aspx

[MSDN-SAVETRAN] Microsoft Corporation, "SAVE TRANSACTION (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188378.aspx

[MSDN-spaddrolemember] Microsoft Corporation, "sp_addrolemember (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms187750.aspx

[MSDN-spexecutesql] Microsoft Corporation, "sp_executesql (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188001.aspx

[MSDN-STDEV] Microsoft Corporation, "STDEV (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms190474.aspx

[MSDN-STDEVP] Microsoft Corporation, "STDEVP (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms176080.aspx

[MSDN-SUBSTRING] Microsoft Corporation, "SUBSTRING (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms187748.aspx

[MSDN-SYSDATETIME] Microsoft Corporation, "SYSDATETIME (Transact-SQL)", http://msdn.microsoft.com/en-us/library/bb630353

[MSDN-UNION] Microsoft Corporation, "UNION (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms180026.aspx

[MSDN-USE] Microsoft Corporation, "USE (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188366.aspx

[MSDN-VAR] Microsoft Corporation, "VAR (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms186290.aspx

[MSDN-VARP] Microsoft Corporation, "VARP (Transact-SQL)", http://msdn.microsoft.com/en-us/library/ms188735.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 standards implements all mandatory features and optionally implements any optional features. For Part 2 of the standards (International Standard ISO/IEC 9075), 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 [ISO/IEC9075-2:2011] that are considered normative and that are considered informative.

Section	Normative/Informative
1 - 2	Informative
3 - 25	Normative
Annexes A - G	Informative

1.5 Notation

The following notations are used to identify clarifications in the 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 from [ISO/IEC9075-2:2008] and [ISO/IEC9075-2:2011], as applicable.

2.1.1 Mandatory Features

2.1.1.1 E021, Character string types

V0001:

The specification states the following:

```
Subclause 6.1,"<data type>"

<character string type> ::=
CHARACTER [ <left paren> <character length> <right paren> ]
| CHAR [ <left paren> <character length> <right paren> ]
| CHARACTER VARYING <left paren> <character length> <right paren>
| CHAR VARYING <left paren> <character length> <right paren>
| VARCHAR <left paren> <character length> <right paren>
| <character large object type>
```

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

Transact-SQL partially supports this feature. However, Transact-SQL generally specifies string lengths, such as those in function parameters and results, as bytes rather than characters. In addition, although the specification allows trailing blank characters to pad fixed-length strings, some Transact-SQL functions, such as **LEN** [MSDN-LEN], exclude trailing blanks.

2.1.1.2 E021-01, CHARACTER data type (including all its spellings)

V0002:

The specification states the following:

```
Subclause 6.1, "<data type>":

<character string type> ::=
CHARACTER [ <left paren> <character length> <right paren> ]
| CHAR [ <left paren> <character length> <right paren> ]
| CHARACTER VARYING <left paren> <character length> <right paren>
| CHAR VARYING <left paren> <character length> <right paren>
| VARCHAR <left paren> <character length> <right paren>
| <character large object type>
```

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

Transact-SQL partially supports this feature. The length of the string is specified in bytes, not characters.

2.1.1.3 E021-02, CHARACTER VARYING data type (including all its spellings)

V0003:

The specification states the following:

```
Subclause 6.1, "<data type>":

<character string type> ::=
CHARACTER [ <left paren> <character length> <right paren> ]
| CHAR [ <left paren> <character length> <right paren> ]
| CHARACTER VARYING <left paren> <character length> <right paren>
| CHAR VARYING <left paren> <character length> <right paren>
| VARCHAR <left paren> <character length> <right paren>
| <character large object type>
```

Microsoft SQL Server 2008 R2 varies as follows:

Transact-SQL partially supports this feature. The length of the string is specified in bytes, not characters.

2.1.1.4 E021-04, CHARACTER_LENGTH function

V0004:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<char length expression> ::=
{ CHAR_LENGTH | CHARACTER_LENGTH } <left paren> <character value expression>
[ USING <char length units> ] <right paren>
```

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

Transact-SQL does not support this feature. See the **LEN** function [MSDN-LEN] for equivalent functionality. **LEN** accepts both the ASCII and Unicode character sets, but, unlike **CHARACTER_LENGTH**, excludes trailing blank characters.

2.1.1.5 E021-05, OCTET_LENGTH function

V0005:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<octet length expression> ::=
OCTET_LENGTH <left paren> <string value expression> <right paren>
```

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

Transact-SQL does not support this feature. See the **DATALENGTH** function [MSDN-DATALENGTH] for equivalent functionality. Like **OCTET_LENGTH**, **DATALENGTH** counts trailing blank characters.

2.1.1.6 E021-06, SUBSTRING function

V0006:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<character substring function> ::=
SUBSTRING <left paren> <character value expression> FROM <start position>
[ FOR <string length> ] [ USING <char length units> ] <right paren>
```

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

Transact-SQL does not support this feature. See the Transact-SQL **SUBSTRING** function [MSDN-SUBSTRING], which differs significantly in syntax and features, for equivalent functionality.

2.1.1.7 E021-07, Character concatenation

V0007:

The specification states the following:

```
Subclause 5.2, "<token> and <separator>":

<concatenation operator> ::=
||

Subclause 6.29, "<string value expression>":

<concatenation> ::=
<character value expression> <concatenation operator> <character factor>
```

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

Transact-SQL does not support this feature. See the **+ (Add)** operator [MSDN-Add] for equivalent functionality to the concatenation operator (**||**) operator in [ISO/IEC9075-2:2011].

2.1.1.8 E021-09, TRIM function

V0008:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<trim function> ::=
TRIM <left paren> <trim operands> <right paren>
```

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

Transact-SQL does not support this feature. See the **LTRIM** [MSDN-LTRIM] and **RTRIM** [MSDN-RTRIM] functions for equivalent functionality.

2.1.1.9 E021-11, POSITION function

V0009:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<position expression> ::=
<character position expression>
| <binary position expression>
```

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

Transact-SQL does not support this feature. See the **CHARINDEX** function [MSDN-CHARINDEX] for equivalent functionality.

2.1.1.10 E051, Basic query specification

V0010:

The specification states the following:

```
Subclause 7.12, "<query specification>":

<query specification> ::=
SELECT [ <set quantifier> ] <select list> <table expression>

<select list> ::=
<asterisk>
| <select sublist> [ { <comma> <select sublist> }... ]

<select sublist> ::=
<derived column>
| <qualified asterisk>

<qualified asterisk> ::=
<asterisked identifier chain> <period> <asterisk>
| <all fields reference>

<asterisked identifier chain> ::=
<asterisked identifier> [ { <period> <asterisked identifier> }... ]

<asterisked identifier> ::=
<identifier>

<derived column> ::=
<value expression> [ <as clause> ]

<as clause> ::=
[ AS ] <column name>

<all fields reference> ::=
<value expression primary> <period> <asterisk>
```

```
[ AS <left paren> <all fields column name list> <right paren> ]

<all fields column name list> ::=
<column name list>
```

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

Transact-SQL partially supports this feature. Transact-SQL supports all of feature E051's subfeatures except E051-09 (section 2.1.1.11).

2.1.1.11 E051-09, Rename columns in the FROM clause

V0011:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<table primary> ::=
<table or query name> [ [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ] ]
| <derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <lateral derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <collection derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <table function derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <only spec> [ [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ] ]
| <parenthesized joined table>
```

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

Transact-SQL partially supports this feature. Transact-SQL supports this feature only for derived tables.

See [ISO/IEC9075-1:2011] for the definition of "table".

2.1.1.12 E081, Basic Privileges

V0012:

The specification states the following:

```
Subclause 12.3, "<privileges>":

<privileges> ::=
<object privileges> ON <object name>

<object name> ::=
[ TABLE ] <table name>
| DOMAIN <domain name>
| COLLATION <collation name>
```

```
| CHARACTER SET <character set name>
| TRANSLATION <transliteration name>
| TYPE <schema-resolved user-defined type name>
| SEQUENCE <sequence generator name>
| <specific routine designator>

<object privileges> ::=
ALL PRIVILEGES
| <action> [ { <comma> <action> }... ]

<action> ::=
SELECT
| SELECT <left paren> <privilege column list> <right paren>
| SELECT <left paren> <privilege method list> <right paren>
| DELETE
| INSERT [ <left paren> <privilege column list> <right paren> ]
| UPDATE [ <left paren> <privilege column list> <right paren> ]
| REFERENCES [ <left paren> <privilege column list> <right paren> ]
| USAGE
| TRIGGER
| UNDER
| EXECUTE
```

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

Transact-SQL partially supports this feature. Transact-SQL supports all of feature E081's subfeatures except E081-09 (section 2.1.1.13).

2.1.1.13 E081-09, USAGE privilege

V0013:

The specification states the following:

```
Subclause 12.3, "<privileges>":

<privileges> ::=
<object privileges> ON <object name>

<object name> ::=
[ TABLE ] <table name>
DOMAIN <domain name>
COLLATION <collation name>
CHARACTER SET <character set name>
TRANSLATION <transliteration name>
TYPE <schema-resolved user-defined type name>
SEQUENCE <sequence generator name>
<specific routine designator>

<object privileges> ::=
ALL PRIVILEGES
| <action> [ { <comma> <action> }... ]

<action> ::=
SELECT
| SELECT <left paren> <privilege column list> <right paren>
| SELECT <left paren> <privilege method list> <right paren>
```

```
| DELETE
| INSERT [ <left paren> <privilege column list> <right paren> ]
| UPDATE [ <left paren> <privilege column list> <right paren> ]
| REFERENCES [ <left paren> <privilege column list> <right paren> ]
| USAGE
| TRIGGER
| UNDER
| EXECUTE
```

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

Transact-SQL does not support this feature.

2.1.1.14 E121, Basic cursor support

V0014:

The specification states the following:

```
Clause 14, "Data manipulation":

<declare cursor> ::=
DECLARE <cursor name> <cursor properties>
FOR <cursor specification>

<open statement> ::=
OPEN <cursor name>

<fetch statement> ::=
FETCH [ [ <fetch orientation> ] FROM ] <cursor name> INTO <fetch target list>

<close statement> ::=
CLOSE <cursor name>

<delete statement: positioned> ::=
DELETE FROM <target table> [ [ AS ] <correlation name> ]
WHERE CURRENT OF <cursor name>

<update statement: positioned> ::=
UPDATE <target table> [ [ AS ] <correlation name> ]
SET <set clause list>
WHERE CURRENT OF <cursor name>
```

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

Transact-SQL partially supports this feature. Transact-SQL partially supports subfeature E121-17 (section 2.1.1.15).

2.1.1.15 E121-17, WITH HOLD cursors

V0015:

The specification states the following:

```
Subclause 14.1, "<declare cursor>": Where the <value expression> in the <sort key> need not
be a <column reference> and need not be in the <select list>, and <cursor holdability> may be
specified

<declare cursor> ::=
DECLARE <cursor name> <cursor properties>
FOR <cursor specification>
```

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

Transact-SQL partially supports this feature.

2.1.1.16 E141, Basic integrity constraints

V0016:

The specification states the following:

```
Subclause 11.6, "<table constraint definition>":

<table constraint definition> ::=
[ <constraint name definition> ] <table constraint>
[ <constraint characteristics> ]

<table constraint> ::=
<unique constraint definition>
| <referential constraint definition>
| <check constraint definition>
```

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

Transact-SQL partially supports this feature.

2.1.1.17 E141-07, Column defaults

V0017:

The specification states the following:

```
Subclause 11.4, "<column definition>": With <default clause>

<column definition> ::=
<column name> [ <data type or domain name> ]
[ <default clause> | <identity column specification> | <generation clause> ]
[ <column constraint definition>... ]
[ <collate clause> ]

Subclause 11.5, "<default clause>":

<default clause> ::=
DEFAULT <default option>

<default option> ::=
<literal>
| <datetime value function>
```

```
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
| CURRENT_PATH
| <implicitly typed value specification>

Subclause 6.32, "<datetime value function>":

<datetime value function> ::=
<current date value function>
| <current time value function>
| <current timestamp value function>
| <current local time value function>
| <current local timestamp value function>

<current date value function> ::=
CURRENT_DATE

<current time value function> ::=
CURRENT_TIME [ <left paren> <time precision> <right paren> ]

<current local time value function> ::=
LOCALTIME [ <left paren> <time precision> <right paren> ]

...

<current local timestamp value function> ::=
LOCALTIMESTAMP [ <left paren> <timestamp precision> <right paren> ]
```

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

Transact-SQL partially supports this feature. However, the **CURRENT_DATE**, **CURRENT_TIME**, **LOCALTIME**, and **LOCALTIMESTAMP** value functions are not supported in **DEFAULT** expressions.

2.1.1.18 E152, Basic SET TRANSACTION statement

V0018:

The specification states the following:

```
Subclause 17.2, "<set transaction statement>":

<set transaction statement> ::=
SET [ LOCAL ] TRANSACTION <transaction characteristics>

Subclause 17.3, "<transaction characteristics>":

<transaction characteristics> ::=
[ <transaction mode> [ { <comma> <transaction mode> }... ] ]

<transaction mode> ::=
<isolation level>
| <transaction access mode>
```

```
| <diagnostics size>

...

<isolation level> ::=
ISOLATION LEVEL <level of isolation>

<level of isolation> ::=
READ UNCOMMITTED
| READ COMMITTED
| REPEATABLE READ
| SERIALIZABLE
```

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

Transact-SQL partially supports this feature. Transact-SQL supports all four isolation levels, except for the **LOCAL** syntax.

2.1.1.19 E152-02, SET TRANSACTION statement: READ ONLY and READ WRITE clauses

V0019:

The specification states the following:

```
Subclause 17.2, "<set transaction statement>":
with <transaction access mode> of READ ONLY or READ WRITE

<set transaction statement> ::=
SET [ LOCAL ] TRANSACTION <transaction characteristics>

Subclause 17.3, "<transaction characteristics>":

<transaction characteristics> ::=
[ <transaction mode> [ { <comma> <transaction mode> }... ] ]

<transaction mode> ::=
<isolation level>
| <transaction access mode>
| <diagnostics size>

<transaction access mode> ::=
READ ONLY
| READ WRITE
```

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

Transact-SQL does not support this feature. Transact-SQL does not support transaction access modes, but equivalent functionality is available by using permissions.

See [ISO/IEC9075-1:2011] for the definition of "transaction".

2.1.1.20 E171, SQLSTATE support

V0020:

The specification states the following:

```
Subclause 24.1, "SQLSTATE":

The character string value returned in an SQLSTATE parameter comprises a 2-character class
value followed by a 3-character subclass value, each with an implementation-defined character
set that has a one-octet character encoding form and is restricted to <digit>s and <simple
Latin upper case letter>s. Table 33, "SQLSTATE class and subclass values", specifies the
class value for each condition and the subclass value or values for each class value.
```

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

Transact-SQL does not support this feature. However, Transact-SQL does support five-character parameters that are similar to **SQLSTATE** for indicating the successful completion of a statement or the raising of an exception.

2.1.1.21 E182, Module language

V0021:

The specification states the following:

```
Clause 13, "SQL-client modules"

Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

<module authorization clause> ::=
SCHEMA <schema name>
| AUTHORIZATION <module authorization identifier>
[ FOR STATIC { ONLY | AND DYNAMIC } ]
| SCHEMA <schema name> AUTHORIZATION <module authorization identifier>
[ FOR STATIC { ONLY | AND DYNAMIC } ]
```

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

Transact-SQL does not support this feature. Transact-SQL supports embedded language features instead.

2.1.1.22 F031, Basic schema manipulation

V0022:

The specification states the following:

```
Clause 11, "Schema definition and manipulation": Selected facilities as indicated by the
subfeatures of this Feature

Subclause 11.1, "<schema definition>":
```

```
<schema definition> ::=
CREATE SCHEMA <schema name clause>
[ <schema character set or path> ]
[ <schema element>... ]
```

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

Transact-SQL partially supports this feature. Transact-SQL supports all of feature F031's subfeatures except F031-13 (section 2.1.1.23), F031-16 (section 2.1.1.24), and F031-19 (section 2.1.1.25).

2.1.1.23 F031-13, DROP TABLE statement: RESTRICT clause

V0023:

The specification states the following:

```
Subclause 11.31, "<drop table statement>":
With a <drop behavior> of RESTRICT

<drop table statement> ::=
DROP TABLE <table name> <drop behavior>

Subclause 11.2, "<drop schema statement>":

<drop behavior> ::=
CASCADE
| RESTRICT
```

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

Transact-SQL does not support this feature. The **DROP TABLE** statement exists in Transact-SQL, but the statement does not support the **RESTRICT** keyword.

2.1.1.24 F031-16, DROP VIEW statement: RESTRICT clause

V0024:

The specification states the following:

```
Subclause 11.33, "<drop view statement>":
With a <drop behavior> of RESTRICT

<drop view statement> ::=
DROP VIEW <table name> <drop behavior>

Subclause 11.2, "<drop schema statement>":

<drop behavior> ::=
CASCADE
| RESTRICT
```

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

Transact-SQL does not support this feature. The **DROP VIEW** statement exists in Transact-SQL, but the statement does not support the **RESTRICT** keyword.

2.1.1.25 F031-19, REVOKE statement: RESTRICT clause

V0025:

The specification states the following:

```
Subclause 12.7, "<revoke statement>": With a <drop behavior> of RESTRICT, only where the use
of this statement can be restricted to the owner of the table being dropped

<revoke statement> ::=
<revoke privilege statement>
| <revoke role statement>

<revoke privilege statement> ::=
REVOKE [ <revoke option extension> ] <privileges>
FROM <grantee> [ { <comma> <grantee> }... ]
[ GRANTED BY <grantor> ]
<drop behavior>

...

<revoke role statement> ::=
REVOKE [ ADMIN OPTION FOR ] <role revoked> [ { <comma> <role revoked> }... ]
FROM <grantee> [ { <comma> <grantee> }... ]
[ GRANTED BY <grantor> ]
<drop behavior>

Subclause 11.2, "<drop schema statement>":

<drop behavior> ::=
CASCADE
| RESTRICT
```

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

Transact-SQL does not support this feature. The **REVOKE** statement exists in Transact-SQL, but the statement does not support the **RESTRICT** keyword.

2.1.1.26 F051, Basic date and time

V0026:

The specification states the following:

```
Subclause 6.1, "<data type>": <datetime type> including datetime literals, datetime
comparisons, and datetime conversions

<datetime type> ::=
DATE
| TIME [ <left paren> <time precision> <right paren> ] [ <with or without time zone> ]
| TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]
[ <with or without time zone> ]

<with or without time zone> ::=
```

```
WITH TIME ZONE
| WITHOUT TIME ZONE

<time precision> ::=
<time fractional seconds precision>

<timestamp precision> ::=
<time fractional seconds precision>

<time fractional seconds precision> ::=
<unsigned integer>

<interval type> ::=
INTERVAL <interval qualifier>
```

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

Transact-SQL partially supports this feature.

2.1.1.27 F051-03, TIMESTAMP data type (including support of TIMESTAMP literal) with fractional seconds precision of at least 0 and 6

V0027:

The specification states the following:

```
Subclause 5.3, "<literal>": The <timestamp literal> form of <datetime literal>, where the
value of <unquoted timestamp string> is simply <time value> that does not include the
optional <time zone interval>

<timestamp literal> ::=
TIMESTAMP <timestamp string>

...

<timestamp string> ::=
<quote> <unquoted timestamp string> <quote>

<time zone interval> ::=
<sign> <hours value> <colon> <minutes value>

...

<time value> ::=
<hours value> <colon> <minutes value> <colon> <seconds value>

Subclause 6.1, "<data type>": The TIMESTAMP <datetime type> without the <with or without time
zone> clause

<datetime type> ::=
DATE
...
| TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]...

Subclause 6.31, "<datetime value expression>":
For values of type TIMESTAMP
```

```
<datetime value expression> ::=
<datetime term>
| <interval value expression> <plus sign> <datetime term>
| <datetime value expression> <plus sign> <interval term>
| <datetime value expression> <minus sign> <interval term>
```

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

Transact-SQL does not support this feature. The Transact-SQL **timestamp** data type is different from the **timestamp** data type that is defined in [ISO/IEC9075-2:2011]. See the **datetime2** data type [MSDN-datetime2] for equivalent functionality.

2.1.1.28 F051-06, CURRENT_DATE

V0028:

The specification states the following:

```
Subclause 6.32, "<datetime value function>": The <current date value function>

<current date value function> ::=
CURRENT_DATE

Subclause 6.31, "<datetime value expression>":
When the value is a <current date value function>

<datetime value expression> ::=
<datetime term>
| <interval value expression> <plus sign> <datetime term>
| <datetime value expression> <plus sign> <interval term>
| <datetime value expression> <minus sign> <interval term>

<datetime term> ::=
<datetime factor>

<datetime factor> ::=
<datetime primary> [ <time zone> ]

<datetime primary> ::=
<value expression primary>
| <datetime value function>
```

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

Transact-SQL does not support this feature. See the **GETDATE** [MSDN-GETDATE], **SYSDATETIME** [MSDN-SYSDATETIME], and **CURRENT_TIMESTAMP** [MSDN-CURRENTTIMESTAMP] functions for equivalent functionality.

2.1.1.29 F051-07, LOCALTIME

V0029:

The specification states the following:

```
Subclause 6.32, "<datetime value function>": The <current local time value function>

<datetime value function> ::=
<current date value function>
| <current time value function>
| <current timestamp value function>
| <current local time value function>
| <current local timestamp value function>

...

<current local time value function> ::=
LOCALTIME [ <left paren> <time precision> <right paren> ]

Subclause 6.31, "<datetime value expression>": When the value is a <current local time value
function>

<datetime value expression> ::=
<datetime term>
| <interval value expression> <plus sign> <datetime term>
| <datetime value expression> <plus sign> <interval term>
| <datetime value expression> <minus sign> <interval term>

Subclause 11.5, "<default clause>": LOCALTIME option of <datetime value function>

<default clause> ::=
DEFAULT <default option>

<default option> ::=
<literal>
| <datetime value function>
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
| CURRENT_PATH
| <implicitly typed value specification>
```

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

Transact-SQL does not support this feature. See the **datetimeoffset** data type [MSDN-datetimeoffset] for equivalent functionality.

2.1.1.30 F051-08, LOCALTIMESTAMP

V0030:

The specification states the following:

```
Subclause 6.32, "<datetime value function>": The <current local timestamp value function>

<datetime value function> ::=
<current date value function>
| <current time value function>
```

```
| <current timestamp value function>
| <current local time value function>
| <current local timestamp value function>

...

<current local timestamp value function> ::=
LOCALTIMESTAMP [ <left paren> <timestamp precision> <right paren> ]

Subclause 6.31, "<datetime value expression>": When the value is a <current local timestamp
value function>

<datetime value expression> ::=
<datetime term>
| <interval value expression> <plus sign> <datetime term>
| <datetime value expression> <plus sign> <interval term>
| <datetime value expression> <minus sign> <interval term>

Subclause 11.5, "<default clause>": LOCALTIMESTAMP option of <datetime value function>

<default clause> ::=
DEFAULT <default option>

<default option> ::=
<literal>
| <datetime value function>
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
| CURRENT_PATH
| <implicitly typed value specification>
```

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

Transact-SQL does not support this feature. See the **datetimeoffset** data type [MSDN-datetimeoffset] for equivalent functionality.

2.1.1.31 F812, Basic flagging

V0031:

The [ISO/IEC9075-2:2011] specification states the following:

```
Part 1, Subclause 8.5, "SQL flagger": With "level of flagging" specified to be Core SQL
Flagging and "extent of checking" specified to be Syntax Only

NOTE — This form of flagging identifies vendor extensions and other non-standard SQL by
checking syntax only without requiring access to the catalog information.
```

The [ISO/IEC9075-1:2011] specification states the following:

```
Subclause 8.5, "SQL flagger":

An SQL Flagger is an implementation-provided facility that is able to identify SQL language
extensions, or other SQL processing alternatives, that may be provided by a conforming SQL-
implementation...

An SQL Flagger provides one or more of the following "level of flagging" options:
— Core SQL Flagging
— Part SQL Flagging

An SQL Flagger that provides one of these options shall be able to identify SQL language
constructs that violate the indicated subset of SQL language. The subset of SQL language used
by "Core SQL Flagging" is Core SQL....

An SQL Flagger provides one or more of the following "extent of checking" options:
— Syntax Only
— Catalog Lookup

Under the Syntax Only option, the SQL Flagger analyzes only the SQL language that is
presented; it checks for violations of any Syntax Rules that can be determined without access
to the Information Schema. It does not necessarily detect violations that depend on the data
type of syntactic elements, even if such violations are in principle deducible from the
syntax alone.
```

The [ISO/IEC9075-1:2008] specification excerpt differs from the [ISO/IEC9075-1:2011] specification as follows:

```
An SQL Flagger provides one or more of the following "level of flagging" options:
— Core SQL Flagging
— Part SQL Flagging
— Package SQL Flagging
```

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

Transact-SQL partially supports this feature. Transact-SQL supports the **SET FIPS_FLAGGER** statement, which specifies that a Transact-SQL program checks for compliance with the FIPS 127-2 standard [FIPS127], which is based on the ISO SQL-92 standard [SQL92].

2.1.1.32 S011, Distinct data types

V0032:

The specification states the following:

```
Subclause 11.51, "<user-defined type definition>": When <representation> is <predefined type>

<user-defined type definition> ::=
CREATE TYPE <user-defined type body>

<user-defined type body> ::=
<schema-resolved user-defined type name>
[ <subtype clause> ]
[ AS <representation> ]
[ <user-defined type option list> ]
[ <method specification list> ]
```

```
...

<representation> ::=
<predefined type>
| <member list>

Subclause 11.59, "<drop data type statement>":

<drop data type statement> ::=
DROP TYPE <schema-resolved user-defined type name> <drop behavior>
```

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 with some equivalent functionality. The user-defined types (UDTs) are not strongly typed, and only the operations that are supported by a UDT's base type are supported.

See [ISO/IEC9075-2:2011] for the definition of "user-defined type (UDT)".

2.1.1.33 T321, Basic SQL-invoked routines

V0033:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

NOTE — "Routine" is the collective term for functions, methods, and procedures. This feature
requires a conforming SQL-implementation to support both user-defined functions and user-
defined procedures. An SQL-implementation that conforms to Core SQL shall support at least
one language for writing routines; that language may be SQL. If the language is SQL, then the
basic specification capability in Core SQL is the ability to specify a one-statement routine.
Support for overloaded functions and procedures is not part of Core SQL.

<SQL-invoked routine> ::=
<schema routine>

<schema routine> ::=
<schema procedure>
| <schema function>

<schema procedure> ::=
CREATE <SQL-invoked procedure>

<schema function> ::=
CREATE <SQL-invoked function>

<SQL-invoked procedure> ::=
PROCEDURE <schema qualified routine name> <SQL parameter declaration list>
<routine characteristics>
<routine body>

...

<SQL parameter declaration list> ::=
<left paren> [ <SQL parameter declaration>
[ { <comma> <SQL parameter declaration> }... ] ] <right paren>
```

```
<SQL parameter declaration> ::=
[ <parameter mode> ] [ <SQL parameter name> ] <parameter type> [ RESULT ]

<parameter mode> ::=
IN
| OUT
| INOUT

Subclause 11.62, "<drop routine statement>" — If Feature T041, "Basic LOB data type support",
is supported, then the <locator indication> clause shall also be supported

<drop routine statement> ::=
DROP <specific routine designator> <drop behavior>
```

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

Transact-SQL partially supports this feature. The Transact-SQL **CREATE FUNCTION** and **CREATE PROCEDURE** statements do not support the **IN**, **INOUT**, or **LANGUAGE** keywords.

2.1.1.34 T321-01, User-defined functions with no overloading

V0034:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>": With <function specification>

<SQL-invoked routine> ::=
<schema routine>

<schema routine> ::=
<schema procedure>
| <schema function>

<schema procedure> ::=
CREATE <SQL-invoked procedure>

<schema function> ::=
CREATE <SQL-invoked function>

...

<SQL-invoked function> ::=
{ <function specification> | <method specification designator> } <routine body>

...

<function specification> ::=
FUNCTION <schema qualified routine name> <SQL parameter declaration list>
<returns clause>
<routine characteristics>
[ <dispatch clause> ]
```

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

Transact-SQL partially supports this feature.

2.1.1.35 T321-02, User-defined stored procedures with no overloading

V0035:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>": With <SQL-invoked procedure>

<SQL-invoked routine> ::=
<schema routine>

<schema routine> ::=
<schema procedure>
| <schema function>

<schema procedure> ::=
CREATE <SQL-invoked procedure>

<schema function> ::=
CREATE <SQL-invoked function>

<SQL-invoked procedure> ::=
PROCEDURE <schema qualified routine name> <SQL parameter declaration list>
<routine characteristics>
<routine body>
```

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

Transact-SQL partially supports this feature.

2.1.1.36 T321-04, CALL statement

V0036:

The specification states the following:

```
Subclause 16.1, "<call statement>":

<call statement> ::=
CALL <routine invocation>

Subclause 10.4, "<routine invocation>": Used by <call statement>s

<routine invocation> ::=
<routine name> <SQL argument list>
```

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

Transact-SQL partially supports this feature. See the **EXECUTE** function [MSDN-EXECUTE] for equivalent functionality.

2.1.2 Optional Features

2.1.2.1 B011, Embedded Ada

V0037:

The specification states the following:

```
Subclause 21.3, "<embedded SQL Ada program>":

<embedded SQL Ada program> ::=
!! See the Syntax Rules.

...

Syntax Rules

1) An <embedded SQL Ada program> is a compilation unit that consists of Ada text and SQL
text. The Ada text shall conform to [ISO8652]. The SQL text shall consist of one or more
<embedded SQL statement>s and, optionally, one or more <embedded SQL declare section>s.
```

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

Transact-SQL does not support this feature.

2.1.2.2 B012, Embedded C

V0038:

The specification states the following:

```
Subclause 21.4, "<embedded SQL C program>":

<embedded SQL C program> ::=
!! See the Syntax Rules.

...

Syntax Rules

1) An <embedded SQL C program> is a compilation unit that consists of C text and SQL text.
The C text shall conform to [ISO9899]. The SQL text shall consist of one or more <embedded
SQL statement>s and, optionally, one or more <embedded SQL declare section>s.
```

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

Transact-SQL does not support this feature.

2.1.2.3 B013, Embedded COBOL

V0039:

The specification states the following:

```
Subclause 21.5, "<embedded SQL COBOL program>":
```

```
<embedded SQL COBOL program> ::=
!! See the Syntax Rules.

...

Syntax Rules

1) An <embedded SQL COBOL program> is a compilation unit that consists of COBOL text and SQL
text. The COBOL text shall conform to [ISO1989]. The SQL text shall consist of one or more
<embedded SQL statement>s and, optionally, one or more <embedded SQL declare section>s.
```

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

Transact-SQL does not support this feature.

2.1.2.4 B014, Embedded Fortran

V0040:

The specification states the following:

```
<Subclause 21.6, "<embedded SQL Fortran program>":

<embedded SQL Fortran program> ::=
!! See the Syntax Rules.

...

Syntax Rules

1) An <embedded SQL Fortran program> is a compilation unit that consists of Fortran text and
SQL text. The Fortran text shall conform to [ISO1539]. The SQL text shall consist of one or
more <embedded SQL statement>s and, optionally, one or more <embedded SQL declare section>s.
```

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

Transact-SQL does not support this feature.

2.1.2.5 B015, Embedded MUMPS

V0041:

The specification states the following:

```
Subclause 21.7, "<embedded SQL MUMPS program>":

<embedded SQL MUMPS program> ::=
!! See the Syntax Rules.

...

Syntax Rules
```

```
1) An <embedded SQL MUMPS program> is a compilation unit that consists of M text and SQL
text. The M text shall conform to [ISO11756]. The SQL text shall consist of one or more
<embedded SQL statement>s and, optionally, one or more <embedded SQL declare section>s.
```

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

Transact-SQL does not support this feature.

2.1.2.6 B016, Embedded Pascal

V0042:

The specification states the following:

```
Subclause 21.8, "<embedded SQL Pascal program>":

<embedded SQL Pascal program> ::=
!! See the Syntax Rules.

...

Syntax Rules

1) An <embedded SQL Pascal program> is a compilation unit that consists of Pascal text and
SQL text. The Pascal text shall conform to one of [ISO7185] or [ISO10206]. The SQL text shall
consist of one or more <embedded SQL statement>s and, optionally, one or more <embedded SQL
declare section>s.
```

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

Transact-SQL does not support this feature.

2.1.2.7 B017, Embedded PL/I

V0043:

The specification states the following:

```
Subclause 21.9, "<embedded SQL PL/I program>":

<embedded SQL PL/I program> ::=
!! See the Syntax Rules.

...

Syntax Rules

1) An <embedded SQL PL/I program> is a compilation unit that consists of PL/I text and SQL
text. The PL/I text shall conform to [ISO6160]. The SQL text shall consist of one or more
<embedded SQL statement>s and, optionally, one or more <embedded SQL declare section>s.
```

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

Transact-SQL does not support this feature.

2.1.2.8 B021, Direct SQL

V0044:

The specification states the following:

```
Subclause 22.1, "<direct SQL statement>":

<direct SQL statement> ::=
<directly executable statement> <semicolon>

...

Conformance Rules

Without Feature B021, "Direct SQL", conforming SQL language shall not contain a <direct SQL
statement>.
```

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

Transact-SQL partially supports this feature. Some Transact-SQL statements can be issued directly by Open Database Connectivity applications. For more information, see [MSDN-MSODBC].

2.1.2.9 B031, Basic dynamic SQL

V0045:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<SQL statement name> ::=
<statement name>
| <extended statement name>


...

<dynamic cursor name> ::=
<cursor name>
| <extended cursor name>

...

<descriptor name> ::=
<non-extended descriptor name>
| <extended descriptor name>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain an <SQL
statement name>.

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain <dynamic
cursor name>.
```

```
Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<descriptor name>.

Subclause 6.4, "<value specification> and <target specification>":

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

<dynamic parameter specification> ::=
<question mark>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<general value specification> that contains a <dynamic parameter specification>.

Subclause 20.2, "<allocate descriptor statement>":

<allocate descriptor statement> ::=
ALLOCATE [ SQL ] DESCRIPTOR <descriptor name> [ WITH MAX <occurrences> ]

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain an
<allocate descriptor statement>.

Subclause 20.3, "<deallocate descriptor statement>":

<deallocate descriptor statement> ::=
DEALLOCATE [ SQL ] DESCRIPTOR <descriptor name>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<deallocate descriptor statement>.
```

```
Subclause 20.4, "<get descriptor statement>":

<get descriptor statement> ::=
GET [ SQL ] DESCRIPTOR <descriptor name> <get descriptor information>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a <get
descriptor statement>.

Subclause 20.5, "<set descriptor statement>":

<set descriptor statement> ::=
SET [ SQL ] DESCRIPTOR <descriptor name> <set descriptor information>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a <set
descriptor statement>.

Subclause 20.6, "<prepare statement>":

<prepare statement> ::=
PREPARE <SQL statement name> [ <attributes specification> ]
FROM <SQL statement variable>

<attributes specification> ::=
ATTRIBUTES <attributes variable>

<attributes variable> ::=
<simple value specification>

<SQL statement variable> ::=
<simple value specification>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<prepare statement>.

Subclause 20.9, "<describe statement>":

<describe output statement> ::=
DESCRIBE [ OUTPUT ] <described object> <using descriptor> [ <nesting option> ]

<nesting option> ::=
WITH NESTING
| WITHOUT NESTING

<using descriptor> ::=
USING [ SQL ] DESCRIPTOR <descriptor name>
```

```
<described object> ::=
<SQL statement name>
| CURSOR <extended cursor name> STRUCTURE

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<describe output statement>.

Subclause 20.10, "<input using clause>":

<input using clause> ::=
<using arguments>
| <using input descriptor>

<using arguments> ::=
USING <using argument> [ { <comma> <using argument> }... ]

<using argument> ::=
<general value specification>

<using input descriptor> ::=
<using descriptor>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain an
<input using clause>.

Subclause 20.11, "<output using clause>":

<output using clause> ::=
<into arguments>
| <into descriptor>

<into arguments> ::=
INTO <into argument> [ { <comma> <into argument> }... ]

<into argument> ::=
<target specification>

<into descriptor> ::=
INTO [ SQL ] DESCRIPTOR <descriptor name>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain an
<output using clause>.

Subclause 20.12, "<execute statement>":

<execute statement> ::=
EXECUTE <SQL statement name> [ <result using clause> ] [ <parameter using clause> ]
```

```
<result using clause> ::=
<output using clause>

<parameter using clause> ::=
<input using clause>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain an
<execute statement>.

Subclause 20.13, "<execute immediate statement>":

<execute immediate statement> ::=
EXECUTE IMMEDIATE <SQL statement variable>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain an
<execute immediate statement>.

Subclause 20.14, "<dynamic declare cursor>":

<dynamic declare cursor> ::=
DECLARE <cursor name>
<cursor properties>
FOR <statement name>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<dynamic declare cursor>.

Subclause 20.17, "<dynamic open statement>":

<dynamic open statement> ::=
OPEN <dynamic cursor name> [ <input using clause> ]

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<dynamic open statement>.

Subclause 20.18, "<dynamic fetch statement>":

<dynamic fetch statement> ::=
FETCH [ [ <fetch orientation> ] FROM ] <dynamic cursor name> <output using clause>

...
```

```
Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<dynamic fetch statement>.

Subclause 20.19, "<dynamic single row select statement>":

<dynamic single row select statement> ::=
<query specification>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<dynamic single row select statement>.

Subclause 20.20, "<dynamic close statement>":

<dynamic close statement> ::=
CLOSE <dynamic cursor name>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<dynamic close statement>.

Subclause 20.21, "<dynamic delete statement: positioned>":

<dynamic delete statement: positioned> ::=
DELETE FROM <target table> WHERE CURRENT OF <dynamic cursor name>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<dynamic delete statement: positioned>.

Subclause 20.22, "<dynamic update statement: positioned>":

<dynamic update statement: positioned> ::=
UPDATE <target table> SET <set clause list>
WHERE CURRENT OF <dynamic cursor name>

...

Conformance Rules

Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain a
<dynamic update statement: positioned>.
```

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

Transact-SQL does not support this feature. See the **EXECUTE**, **EXEC** [MSDN-EXECUTE], and **sp_executesql** [MSDN-spexecutesql] statements for equivalent functionality.

2.1.2.10 B032, Extended dynamic SQL

V0046:

The specification states the following:

```
Subclause 5.3, "<literal>":

<literal> ::=
<signed numeric literal>
| <general literal>

<unsigned literal> ::=
<unsigned numeric literal>
| <general literal>

<general literal> ::=
<character string literal>
| <national character string literal>
| <Unicode character string literal>
| <binary string literal>
| <datetime literal>
| <interval literal>
| <boolean literal>

Subclause 5.4, "Names and identifiers":

<extended statement name> ::=
[ <scope option> ] <simple value specification>

<dynamic cursor name> ::=
<cursor name>
| <extended cursor name>

<extended cursor name> ::=
[ <scope option> ] <simple value specification>

<descriptor name> ::=
<non-extended descriptor name>
| <extended descriptor name>

<non-extended descriptor name> ::=
<identifier>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain a
<extended statement name> or <extended cursor name>.

...

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain a
<descriptor name> that is not a <literal> or a <non-extended descriptor name>.

Subclause 20.2, "<allocate descriptor statement>":

<occurrences> ::=
```

```
<simple value specification>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain an
<occurrences> that is not a <literal>.

Subclause 20.8, "<deallocate prepared statement>":

<deallocate prepared statement> ::=
DEALLOCATE PREPARE <SQL statement name>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain a
<deallocate prepared statement>.

Subclause 20.9, "<describe statement>":

<describe input statement> ::=
DESCRIBE INPUT <SQL statement name> <using descriptor> [ <nesting option> ]

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain a
<describe input statement>.

Subclause 20.12, "<execute statement>":

<result using clause> ::=
<output using clause>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain a
<result using clause>.

Subclause 20.15, "<allocate extended dynamic cursor statement>":

<allocate extended dynamic cursor statement> ::=
ALLOCATE <extended cursor name>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain an
<allocate extended dynamic cursor statement>.

Subclause 20.23, "<preparable dynamic delete statement: positioned>":
```

```
<preparable dynamic delete statement: positioned> ::=
DELETE [ FROM <target table> ]
WHERE CURRENT OF <preparable dynamic cursor name>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain a
<preparable dynamic delete statement: positioned>.

Subclause 20.25, "<preparable dynamic update statement: positioned>":

<preparable dynamic update statement: positioned> ::=
UPDATE [ <target table> ] SET <set clause list>
WHERE CURRENT OF <preparable dynamic cursor name>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain a
<preparable dynamic update statement: positioned>.
```

The [ISO/IEC9075-2:2008] specification differs as follows:

```
Subclause 20.15, "<allocate cursor statement>":

<allocate cursor statement> ::=
ALLOCATE <extended cursor name> <cursor intent>

...

Conformance Rules

Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain an
<allocate cursor statement>.
```

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

Transact-SQL does not support this feature. See the **EXECUTE**, **EXEC** [MSDN-EXECUTE], and **sp_executesql** [MSDN-spexecutesql] statements for equivalent functionality.

2.1.2.11 B032-01, <describe input statement>

V0047:

The specification states the following:

```
Subclause 20.9, "<describe input statement>":

<describe input statement> ::=
DESCRIBE INPUT <SQL statement name> <using descriptor> [ <nesting option> ]
```

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

Transact-SQL does not support this feature.

2.1.2.12 B034, Dynamic specification of cursor attributes

V0048:

The specification states the following:

```
Subclause 20.6, "<prepare statement>”:

<attributes specification> ::=
ATTRIBUTES <attributes variable>

<attributes variable> ::=
<simple value specification>

...

Conformance Rules

Without Feature B034, "Dynamic specification of cursor attributes", conforming SQL language
shall not contain an <attributes specification>.
```

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

Transact-SQL does not support this feature.

2.1.2.13 B035, Non-extended descriptor names

V0049:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<descriptor name> ::=
<non-extended descriptor name>
| <extended descriptor name>

<non-extended descriptor name> ::=
<identifier>

<extended descriptor name> ::=
[ <scope option> ] <simple value specification>

...

Conformance Rules

Without Feature B035, "Non-extended descriptor names", conforming SQL language shall not
contain a <descriptor name> that is a <non-extended descriptor name>.
```

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

Transact-SQL does not support this feature.

2.1.2.14 B041, Extensions to embedded SQL exception declarations

V0050:

The specification states the following:

```
Subclause 21.2, "<embedded exception declaration>":

<embedded exception declaration> ::=
WHENEVER <condition> <condition action>

<condition> ::=
<SQL condition>

<SQL condition> ::=
<major category>
| SQLSTATE ( <SQLSTATE class value> [ , <SQLSTATE subclass value> ] )
| CONSTRAINT <constraint name>

...

Conformance Rules

Without Feature B041, "Extensions to embedded SQL exception declarations", conforming SQL
language shall not contain an <SQL condition> that contains either SQLSTATE or CONSTRAINT.
```

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

Transact-SQL does not support this feature.

2.1.2.15 B051, Enhanced execution rights

V0051:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

<module authorization clause> ::=
SCHEMA <schema name>
| AUTHORIZATION <module authorization identifier>
[ FOR STATIC { ONLY | AND DYNAMIC } ]
| SCHEMA <schema name> AUTHORIZATION <module authorization identifier>
[ FOR STATIC { ONLY | AND DYNAMIC } ]

...

Conformance Rules
```

```
Without Feature B051, "Enhanced execution rights", conforming SQL language shall not contain
a <module authorization clause> that immediately contains FOR STATIC ONLY or FOR STATIC AND
DYNAMIC.

Subclause 21.1, "<embedded SQL host program>":

<statement or declaration> ::=
<declare cursor>
| <dynamic declare cursor>
| <temporary table declaration>
| <embedded authorization declaration>
| <embedded path specification>
| <embedded transform group specification>
| <embedded collation specification>
| <embedded exception declaration>
| <SQL procedure statement>

...

<embedded authorization declaration> ::=
DECLARE <embedded authorization clause>

<embedded authorization clause> ::=
SCHEMA <schema name>
| AUTHORIZATION <embedded authorization identifier>
[ FOR STATIC { ONLY | AND DYNAMIC } ]
| SCHEMA <schema name> AUTHORIZATION <embedded authorization identifier>
[ FOR STATIC { ONLY | AND DYNAMIC } ]

...

Conformance Rules

Without Feature B051, "Enhanced execution rights", conforming SQL language shall not contain
an <embedded authorization declaration>.
```

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

Transact-SQL does not support this feature.

2.1.2.16 B111, Module language Ada

V0052:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

...
```

```
Conformance Rules

Without Feature B111, "Module language Ada", conforming SQL language shall not contain an
<SQL-client module definition> that contains a <language clause> that contains ADA.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.17 B112, Module language C

V0053:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

...

Conformance Rules

Without Feature B112, "Module language C", conforming SQL language shall not contain an <SQL-
client module definition> that contains a <language clause> that contains C.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.18 B113, Module language COBOL

V0054:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
```

```
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

...

Conformance Rules

Without Feature B113, "Module language COBOL", conforming SQL language shall not contain an
<SQL-client module definition> that contains a <language clause> that contains COBOL.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.19 B114, Module language Fortran

V0055:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

...

Conformance Rules

Without Feature B114, "Module language Fortran", conforming SQL language shall not contain an
<SQL-client module definition> that contains a <language clause> that contains FORTRAN.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.20 B115, Module language MUMPS

V0056:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

...

Conformance Rules

Without Feature B115, "Module language MUMPS", conforming SQL language shall not contain an
<SQL-client module definition> that contains a <language clause> that contains M.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.21 B116, Module language Pascal

V0057:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

...

Conformance Rules

Without Feature B116, "Module language Pascal", conforming SQL language shall not contain an
<SQL-client module definition> that contains a <language clause> that contains PASCAL.

Subclause 10.2, "<language clause>":
```

```
<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.22 B117, Module language PL/I

V0058:

The specification states the following:

```
Subclause 13.1, "<SQL-client module definition>":

<SQL-client module definition> ::=
<module name clause> <language clause> <module authorization clause>
[ <module path specification> ]
[ <module transform group specification> ]
[ <module collations> ]
[ <temporary table declaration>... ]
<module contents>...

...

Conformance Rules

Without Feature B117, "Module language PL/I", conforming SQL language shall not contain an
<SQL-client module definition> that contains a <language clause> that contains PLI.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.23 B121, Routine language Ada

V0059:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| <savepoint level indication>
```

```
...

Conformance Rules

Without Feature B121, "Routine language Ada", conforming SQL language shall not contain a
<routine characteristic> that contains a <language clause> that contains ADA.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.24 B122, Routine language C

V0060:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| <savepoint level indication>

...

Conformance Rules

Without Feature B122, "Routine language C", conforming SQL language shall not contain a
<routine characteristic>that contains a <language clause> that contains C.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature. See SQL Server extended stored procedures [MSDN-ExtdStorProc] and CLR stored procedures [MSDN-CLRStorProc] for equivalent functionality.

2.1.2.25 B123, Routine language COBOL

V0061:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| <savepoint level indication>

...

Conformance Rules

Without Feature B123, "Routine language COBOL", conforming SQL language shall not contain a
<routine characteristic> that contains a <language clause> that contains COBOL.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.26 B124, Routine language Fortran

V0062:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| <savepoint level indication>

...

Conformance Rules

Without Feature B124, "Routine language Fortran", conforming SQL language shall not contain a
<routine characteristic> that contains a <language clause> that contains FORTRAN.

Subclause 10.2, "<language clause>":
```

```
<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.27 B125, Routine language MUMPS

V0063:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| <savepoint level indication>

...

Conformance Rules

Without Feature B125, "Routine language MUMPS", conforming SQL language shall not contain a
<routine characteristic> that contains a <language clause> that contains M.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.28 B126, Routine language Pascal

V0064:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
```

```
| <returned result sets characteristic>
| <savepoint level indication>

...

Conformance Rules

Without Feature B126, "Routine language Pascal", conforming SQL language shall not contain a
<routine characteristic> that contains a <language clause> that contains PASCAL.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.29 B127, Routine language PL/I

V0065:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| <savepoint level indication>

...

Conformance Rules

Without Feature B127, "Routine language PL/I", conforming SQL language shall not contain a
<routine characteristic> that contains a <language clause> that contains PLI.

Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name>
```

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

Transact-SQL does not support this feature.

2.1.2.30 B221, Routine language Ada: VARCHAR and NUMERIC support

V0066

The specification states the following:

```
Subclause 10.2, "<language clause>":

<language clause> ::=
LANGUAGE <language name> []

Subclause 11.60, "<SQL-invoked routine>":

<SQL-invoked routine> ::=
<schema routine>

...

<SQL parameter declaration> ::=
[ <parameter mode> ]
[ <SQL parameter name> ]
<parameter type> [ RESULT ]
[ DEFAULT <parameter default> ]

<parameter default> ::=
<value expression>
| <contextually typed value specification>

<parameter mode> ::=
IN
| OUT
| INOUT

<parameter type> ::=
<data type> [ <locator indication> ]

...

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules:

Without Feature B221, "Routine language Ada: VARCHAR and NUMERIC support", if the <language
clause> of the <SQL-invoked routine> specifies ADA, then <parameter type> and <returns data
type> shall not specify CHARACTER VARYING or NUMERIC.

Subclause 13.3, "<externally-invoked procedure>":

<externally-invoked procedure> ::=
PROCEDURE <procedure name> <host parameter declaration list> <semicolon>
<SQL procedure statement> <semicolon>

<host parameter declaration list> ::=
<left paren> <host parameter declaration>
[ { <comma> <host parameter declaration> }... ] <right paren>

<host parameter declaration> ::=
<host parameter name> <host parameter data type>
| <status parameter>
```

```
<host parameter data type> ::=
<data type> [ <locator indication> ]

<status parameter> ::=
SQLSTATE

...

Conformance Rules:

Without Feature B221, "Routine language Ada: VARCHAR and NUMERIC support", if the caller
language of the <externally-invoked procedure> is Ada, then a <host parameter data type>
shall not be CHARACTER VARYING or NUMERIC.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.31 F032, CASCADE drop behavior

V0067:

The specification states the following:

```
Subclause 11.2, "<drop schema statement>":

<drop behavior> ::=
CASCADE
| RESTRICT

Subclause 11.31, "<drop table statement>":

<drop table statement> ::=
DROP TABLE <table name> <drop behavior>

...

Conformance Rules

Without Feature F032, "CASCADE drop behavior", conforming SQL language shall not contain a
<drop table statement> that contains <drop behavior> that contains CASCADE.

Subclause 11.33, "<drop view statement>":

<drop view statement> ::=
DROP VIEW <table name> <drop behavior>

...

Conformance Rules

Without Feature F032, "CASCADE drop behavior", conforming SQL language shall not contain a
<drop view statement> that contains a <drop behavior> that contains CASCADE.
```

```
Subclause 11.59, "<drop data type statement>":

<drop data type statement> ::=
DROP TYPE <schema-resolved user-defined type name> <drop behavior>

Subclause 11.62, "<drop routine statement>":

<drop routine statement> ::=
DROP <specific routine designator> <drop behavior>

...

Conformance Rules

Without Feature F032, "CASCADE drop behavior", conforming SQL language shall not contain a
<drop routine statement> that contains a <drop behavior> that contains CASCADE.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **RESTRICT** keyword, but SQL Server provides the **RESTRICT** functionality implicitly by default.

2.1.2.32 F034, Extended REVOKE statement

V0068

The specification states the following:

```
Subclause 11.2, "<drop schema statement>":

<drop behavior> ::=
CASCADE
| RESTRICT

Subclause 12.3, "<privileges>":

<privileges> ::=
<object privileges> ON <object name>

<object name> ::=
[ TABLE ] <table name>
| DOMAIN <domain name>
| COLLATION <collation name>
| CHARACTER SET <character set name>
| TRANSLATION <transliteration name>
| TYPE <schema-resolved user-defined type name>
| SEQUENCE <sequence generator name>
| <specific routine designator>

<object privileges> ::=
ALL PRIVILEGES
| <action> [ { <comma> <action> }... ]

<action> ::=
SELECT
| SELECT <left paren> <privilege column list> <right paren>
| SELECT <left paren> <privilege method list> <right paren>
```

```
| DELETE
| INSERT [ <left paren> <privilege column list> <right paren> ]
| UPDATE [ <left paren> <privilege column list> <right paren> ]
| REFERENCES [ <left paren> <privilege column list> <right paren> ]
| USAGE
| TRIGGER
| UNDER
| EXECUTE

...

<grantee> ::=
PUBLIC
| <authorization identifier>

Subclause 12.7, "<revoke statement>":

<revoke statement> ::=
<revoke privilege statement>
| <revoke role statement>

<revoke privilege statement> ::=
REVOKE [ <revoke option extension> ] <privileges>
FROM <grantee> [ { <comma> <grantee> }... ]
[ GRANTED BY <grantor> ]
<drop behavior>

<revoke option extension> ::=
GRANT OPTION FOR
| HIERARCHY OPTION FOR

<revoke role statement> ::=
REVOKE [ ADMIN OPTION FOR ] <role revoked> [ { <comma> <role revoked> }... ]
FROM <grantee> [ { <comma> <grantee> }... ]
[ GRANTED BY <grantor> ]
<drop behavior>

...

Conformance Rules

...

Without Feature F034, "Extended REVOKE statement", conforming SQL language shall not contain
a <revoke statement> that contains a <drop behavior> that contains CASCADE.

Without Feature F034, "Extended REVOKE statement", conforming SQL language shall not contain
a <revoke option extension> that contains GRANT OPTION FOR.

Without Feature F034, "Extended REVOKE statement", conforming SQL language shall not contain
a <revoke statement> that contains a <privileges> that contains an <object name> where the
owner of the SQL-schema that is specified explicitly or implicitly in the <object name> is
not the current authorization identifier.

Without Feature F034, "Extended REVOKE statement", conforming SQL language shall not contain
a <revoke statement> such that there exists a privilege descriptor PD that satisfies all the
following conditions:
a) PD identifies the object identified by <object name> simply contained in <privileges>
contained in the <revoke statement>.
```

```
b) PD identifies the <grantee> identified by any <grantee> simply contained in <revoke
statement> and that <grantee> does not identify the owner of the SQL-schema that is specified
explicitly or implicitly in the <object name> simply contained in <privileges> contained in
the <revoke statement>.
c) PD identifies the action identified by the <action> simply contained in <privileges>
contained in the <revoke statement>.
d) PD indicates that the privilege is grantable.
```

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

Transact-SQL partially supports this feature. The **REVOKE** statement in Transact-SQL supports an **AS** clause that specifies a principal from which the principal that is executing the query derives its right to revoke the permission.

See [ISO/IEC9075-1:2011] for the definition of "query".

2.1.2.33 F052, Intervals and datetime arithmetic

V0069

The specification states the following:

```
Subclause 5.3, "<literal>":

<interval literal> ::=
INTERVAL [ <sign> ] <interval string> <interval qualifier>

...

Conformance Rules

Without Feature F052, "Intervals and datetime arithmetic", conforming SQL language shall not
contain an <interval literal>.

Subclause 6.1, "<data type>":

<interval type> ::=
INTERVAL <interval qualifier>

...

Conformance Rules

Without Feature F052, "Intervals and datetime arithmetic", conforming SQL language shall not
contain an <interval type>.

Subclause 6.28, "<numeric value function>":

<extract expression> ::=
EXTRACT <left paren> <extract field> FROM <extract source> <right paren>

...

Conformance Rules

Without Feature F052, "Intervals and datetime arithmetic", conforming SQL language shall not
contain an <extract expression>.
```

```
Subclause 6.31, "<datetime value expression>":

<datetime value expression> ::=
<datetime term>
| <interval value expression> <plus sign> <datetime term>
| <datetime value expression> <plus sign> <interval term>
| <datetime value expression> <minus sign> <interval term>

...

Conformance Rules

Without Feature F052, "Intervals and datetime arithmetic", conforming SQL language shall not
contain <datetime value expression> that immediately contains a <plus sign> or a <minus
sign>.

Subclause 6.32, "<interval value expression>":

<interval value expression> ::=
<interval term>
| <interval value expression 1> <plus sign> <interval term 1>
| <interval value expression 1> <minus sign> <interval term 1>
| <left paren> <datetime value expression> <minus sign> <datetime term> <right paren>
<interval qualifier>

...

Conformance Rules

Without Feature F052, "Intervals and datetime arithmetic", conforming SQL language shall not
contain an <interval value expression>.

Subclause 6.34, "<interval value function>":

<interval value function> ::=
<interval absolute value function>
<interval absolute value function> ::=
ABS <left paren> <interval value expression> <right paren>

...

Conformance Rules

Without Feature F052, "Intervals and datetime arithmetic", conforming SQL shall not contain
an <interval value function>.

g) Subclause 10.1, "<interval qualifier>":

<interval qualifier> ::=
<start field> TO <end field>
| <single datetime field>

...

Conformance Rules

Without Feature F052, "Intervals and datetime arithmetic", conforming SQL language shall not
contain an <interval qualifier>.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **INTERVAL** data type, although calculation of intervals and temporal data is generally supported. Transact-SQL does not support the **EXTRACT** function; for equivalent functionality, see the **DATEPART** function [MSDN-DATEPART]. Transact-SQL does not support interval qualifiers.

2.1.2.34 F053, OVERLAPS predicate

V0070

The specification states the following:

```
Subclause 8.14, "<overlaps predicate>":

<overlaps predicate> ::=
<overlaps predicate part 1> <overlaps predicate part 2>

<overlaps predicate part 1> ::=
<row value predicand 1>

<overlaps predicate part 2> ::=
OVERLAPS <row value predicand 2>

...

Conformance Rules

Without Feature F053, "OVERLAPS predicate", conforming SQL language shall not contain an
<overlaps predicate>.
```

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

Transact-SQL does not support this feature.

2.1.2.35 F054, TIMESTAMP in DATE type precedence list

V0071

The specification states the following:

```
Subclause 9.7, "Type precedence list determination":

Conformance Rules:

Without Feature F054, "TIMESTAMP in DATE type precedence list", the type precedence list of
DATE is

DATE
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.36 F121, Basic diagnostics management

V0072

The specification states the following:

```
Subclause 17.3, "<transaction characteristics>":

<transaction characteristics> ::=
[ <transaction mode> [ { <comma> <transaction mode> }... ] ]

<transaction mode> ::=
<isolation level>
| <transaction access mode>
| <diagnostics size>

...

<diagnostics size> ::=
DIAGNOSTICS SIZE <number of conditions>

<number of conditions> ::=
<simple value specification>

...

Conformance Rules

Without Feature F121, "Basic diagnostics management", conforming SQL language shall not
contain a <diagnostics size>.

Subclause 13.4, "<SQL procedure statement>":

<SQL executable statement> ::=
<SQL schema statement>
| <SQL data statement>
| <SQL control statement>
| <SQL transaction statement>
| <SQL connection statement>
| <SQL session statement>
| <SQL diagnostics statement>
| <SQL dynamic statement>

...

<SQL diagnostics statement> ::=
<get diagnostics statement>

Subclause 23.1, "<get diagnostics statement>":

<get diagnostics statement> ::=
```

```
GET DIAGNOSTICS <SQL diagnostics information>

<SQL diagnostics information> ::=
<statement information>
| <condition information>
| <all information>

...

Conformance Rules

Without Feature F121, "Basic diagnostics management", conforming SQL language shall not
contain a <get diagnostics statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.37 F121-01, GET DIAGNOSTICS statement

V0073

The specification states the following:

```
Subclause 23.1, "<get diagnostics statement>":

Get exception or completion condition information from a diagnostics area.

...

<get diagnostics statement> ::=
GET DIAGNOSTICS <SQL diagnostics information>
```

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

Transact-SQL does not support this feature.

2.1.2.38 F121-02, SET TRANSACTION statement: DIAGNOSTICS SIZE clause

V0074

The specification states the following:

```
Subclause 17.2, "<set transaction statement>":

<set transaction statement> ::=
SET [ LOCAL ] TRANSACTION <transaction characteristics>

Subclause 17.3, "<transaction characteristics>":

<transaction characteristics> ::=
[ <transaction mode> [ { <comma> <transaction mode> }... ] ]

<transaction mode> ::=
<isolation level>
```

```
| <transaction access mode>
| <diagnostics size>

...

<diagnostics size> ::=
DIAGNOSTICS SIZE <number of conditions>
```

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

Transact-SQL does not support this feature.

2.1.2.39 F122, Enhanced diagnostics management

V0075

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<simple target specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <column reference>
| <embedded variable name>

Subclause 23.1, "<get diagnostics statement>":

<get diagnostics statement> ::=
GET DIAGNOSTICS <SQL diagnostics information>

<SQL diagnostics information> ::=
<statement information>
| <condition information>
| <all information>

<statement information> ::=
<statement information item> [ { <comma> <statement information item> }... ]

<statement information item> ::=
<simple target specification> <equals operator> <statement information item name>

<statement information item name> ::=
NUMBER
| MORE
| COMMAND_FUNCTION
| COMMAND_FUNCTION_CODE
| DYNAMIC_FUNCTION
| DYNAMIC_FUNCTION_CODE
| ROW_COUNT
| TRANSACTIONS_COMMITTED
| TRANSACTIONS_ROLLED_BACK
| TRANSACTION_ACTIVE

...

<condition information item name> ::=
```

```
CATALOG_NAME
CLASS_ORIGIN
COLUMN_NAME
CONDITION_NUMBER
CONNECTION_NAME
CONSTRAINT_CATALOG
CONSTRAINT_NAME
CONSTRAINT_SCHEMA
CURSOR_NAME
MESSAGE_LENGTH
MESSAGE_OCTET_LENGTH
MESSAGE_TEXT
PARAMETER_MODE
PARAMETER_NAME
PARAMETER_ORDINAL_POSITION
RETURNED_SQLSTATE
ROUTINE_CATALOG
ROUTINE_NAME
ROUTINE_SCHEMA
SCHEMA_NAME
SERVER_NAME
SPECIFIC_NAME
SUBCLASS_ORIGIN
TABLE_NAME
TRIGGER_CATALOG
TRIGGER_NAME
TRIGGER_SCHEMA

...

Conformance Rules

Without Feature F122, "Enhanced diagnostics management", conforming SQL language shall not
contain a <get diagnostics statement> containing a <simple target specification> whose
declared type is different from that of the corresponding <statement information item name>
or <condition information item name>.
```

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

Transact-SQL does not support this feature.

2.1.2.40 F123, All diagnostics

V0076

The specification states the following:

```
Subclause 23.1, "<get diagnostics statement>":

<get diagnostics statement> ::=
GET DIAGNOSTICS <SQL diagnostics information>

<SQL diagnostics information> ::=
<statement information>
| <condition information>
| <all information>
```

```
...

<all information> ::=
<all info target> <equals operator> ALL [ <all qualifier> ]

<all info target> ::=
<simple target specification>

<all qualifier> ::=
STATEMENT
| CONDITION [ <condition number> ]

<condition number> ::=
<simple value specification>

...

Conformance Rules

Without Feature F123, "All diagnostics", conforming SQL language shall not contain an <all
information>.
```

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

Transact-SQL does not support this feature.

2.1.2.41 F191, Referential delete actions

V0077

The specification states the following:

```
Subclause 11.8, "<referential constraint definition>":

<delete rule> ::=
ON DELETE <referential action>

<referential action> ::=
CASCADE
| SET NULL
| SET DEFAULT
| RESTRICT
| NO ACTION

...

Conformance Rules

Without Feature F191, "Referential delete actions", conforming SQL language shall not contain
a <delete rule>.
```

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

Transact-SQL partially supports this feature. The **ON DELETE RESTRICT** rule is not supported.

2.1.2.42 F202, TRUNCATE TABLE: identity column restart option

V0078

The specification states the following:

```
Subclause 14.10, "<truncate table statement>":

<truncate table statement> ::=
TRUNCATE TABLE <target table> [ <identity column restart option> ]

<identity column restart option> ::=
CONTINUE IDENTITY
| RESTART IDENTITY

...

Conformance Rules:

Without Feature F202, "TRUNCATE TABLE: identity column restart option", conforming SQL
language shall not contain an <identity column restart option>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.43 F231, Privilege tables

V0079

The [ISO/IEC9075-11:2011] specification states the following:

```
Subclause 5.19, "COLUMN_PRIVILEGES view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.COLUMN_PRIVILEGES.

Subclause 5.25, "DATA_TYPE_PRIVILEGES view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.DATA_TYPE_PRIVILEGES.

Subclause 5.41, "ROLE_COLUMN_GRANTS view":

Without Feature F231, "Privilege tables", and Feature T331, "Basic roles", conforming SQL
language shall not reference INFORMATION_SCHEMA.ROLE_COLUMN_GRANTS.

Subclause 5.42, "ROLE_ROUTINE_GRANTS view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROLE_ROUTINE_GRANTS.

Subclause 5.43, "ROLE_TABLE_GRANTS view":
```

```
Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROLE_TABLE_GRANTS.

Subclause 5.46, "ROLE_UDT_GRANTS view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROLE_UDT_GRANTS.

Subclause 5.49, "ROUTINE_PRIVILEGES view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUTINE_PRIVILEGES.

Subclause 5.62, "TABLE_PRIVILEGES view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TABLE_PRIVILEGES.

Subclause 5.73, "UDT_PRIVILEGES view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.UDT_PRIVILEGES.

Subclause 5.74, "USAGE_PRIVILEGES view":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.USAGE_PRIVILEGES.

Subclause 5.81, "Short name views":

Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROLE_ROUT_GRANTS.
```

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

Transact-SQL partially supports this feature. Transact-SQL supports the **TABLE_PRIVILEGES** and **COLUMN_PRIVILEGES** views.

See [ISO/IEC9075-1:2011] for the definition of "view".

2.1.2.44 F231-03, USAGE_PRIVILEGES view

V0080

The [ISO/IEC9075-11:2011] specification states the following:

```
Subclause 5.74, "USAGE_PRIVILEGES view":
Without Feature F231, "Privilege tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.USAGE_PRIVILEGES.
```

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

Transact-SQL does not support this feature.

2.1.2.45 F251, Domain support

V0081

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<domain name> ::=
<schema qualified name>

...

<schema qualified name> ::=
[ <schema name> <period> ] <qualified identifier>

...

Conformance Rules

Without Feature F251, "Domain support", conforming SQL language shall not contain a <domain
name>.

Subclause 6.4, "<value specification> and <target specification>":

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

Conformance Rules

Without Feature F251, "Domain support", conforming SQL language shall not contain a <general
value specification> that contains VALUE.

Subclause 11.34, "<domain definition>":

<domain definition> ::=
CREATE DOMAIN <domain name> [ AS ] <predefined type>
[ <default clause> ]
[ <domain constraint>... ]
[ <collate clause> ]

<domain constraint> ::=
```

```
[ <constraint name definition> ] <check constraint definition> [
<constraint characteristics> ]
...

Conformance Rules

Without Feature F251, "Domain support", conforming SQL language shall not contain a <domain
definition>.

Subclause 11.40, "<drop domain statement>":

<drop domain statement> ::=
DROP DOMAIN <domain name> <drop behavior>
...

Conformance Rules

Without Feature F251, "Domain support", conforming SQL language shall not contain a <drop
domain statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.46 F271, Compound character literals

V0082

The specification states the following:

```
Subclause 5.3, "<literal>":

<character string literal> ::=
[ <introducer><character set specification> ]
<quote> [ <character representation>... ] <quote>
[ { <separator> <quote> [ <character representation>... ] <quote> }... ]

<introducer> ::=
<underscore>

<character representation> ::=
<nonquote character>
| <quote symbol>

...

Conformance Rules

...

Without Feature F271, "Compound character literals", in conforming SQL language, a <character
string literal> shall contain exactly one repetition of <character representation> (that is,
it shall contain exactly one sequence of "<quote> <character representation>... <quote>").
```

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

Transact-SQL does not support this feature.

2.1.2.47 F291, UNIQUE predicate

V0083

The specification states the following:

```
Subclause 8.11, "<unique predicate>":

<unique predicate> ::=
UNIQUE <table subquery>

...

Conformance Rules

Without Feature F291, "UNIQUE predicate", conforming SQL language shall not contain a <unique
predicate>.

NOTE — The Conformance Rules of Subclause 9.12, "Grouping operations", also apply.

Subclause 9.12, "Grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of a grouping operation shall not be ST-ordered.
```

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

Transact-SQL does not support this feature.

2.1.2.48 F301, CORRESPONDING in query expressions

V0084

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<query expression body> ::=
<query term>
| <query expression body> UNION [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>
| <query expression body> EXCEPT [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>

...

<corresponding spec> ::=
```

```
CORRESPONDING [ BY <left paren> <corresponding column list> <right paren> ]

...

Conformance Rules

Without Feature F301, "CORRESPONDING in query expressions", conforming SQL language shall not
contain a <query expression> that contains CORRESPONDING.
```

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

Transact-SQL does not support this feature.

2.1.2.49 F302, INTERSECT table operator

V0085

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<query expression body> ::=
<query term>
| <query expression body> UNION [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>
| <query expression body> EXCEPT [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>

<query term> ::=
<query primary>
| <query term> INTERSECT [ ALL | DISTINCT ]
[ <corresponding spec> ] <query primary>

...

Conformance Rules

Without Feature F302, "INTERSECT table operator", conforming SQL language shall not contain a
<query term> that contains INTERSECT.
```

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

Transact-SQL partially supports this feature.

2.1.2.50 F302-01, INTERSECT DISTINCT table operator

V0086

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<query expression body> ::=
<query term>
| <query expression body> UNION [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>
| <query expression body> EXCEPT [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>

<query term> ::=
<query primary>
| <query term> INTERSECT [ ALL | DISTINCT ]
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **DISTINCT** keyword, but SQL Server provides the **DISTINCT** functionality implicitly by default.

2.1.2.51 F304, EXCEPT ALL table operator

V0087

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

<with clause> ::=
WITH [ RECURSIVE ] <with list>

...

<query expression body> ::=
<query term>
| <query expression body> UNION [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>
| <query expression body> EXCEPT [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>

<query term> ::=
<query primary>
| <query term> INTERSECT [ ALL | DISTINCT ]
```

```
[ <corresponding spec> ] <query primary>

...

Conformance Rules

Without Feature F304, "EXCEPT ALL table operator", conforming SQL language shall not contain
a <query expression> that contains EXCEPT ALL.

NOTE — If DISTINCT, INTERSECT or EXCEPT is specified, then the Conformance Rules of Subclause
9.12, "Grouping operations", apply.

Subclause 9.12, "Grouping operations":

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of a grouping operation shall not be ST-ordered.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **DISTINCT** keyword, but SQL Server provides the **DISTINCT** functionality implicitly by default.

2.1.2.52 F314, MERGE statement with DELETE branch

V0088

The specification states the following:

```
Subclause 14.12, "<merge statement>":

<merge when matched clause> ::=
WHEN MATCHED [ AND <search condition> ]
THEN <merge update or delete specification>

<merge update or delete specification> ::=
<merge update specification>
| <merge delete specification>
...

<merge delete specification> ::=
DELETE

...

Conformance Rules:
Without Feature F314, "MERGE statement with DELETE branch", in conforming SQL language, a
<merge when matched clause> shall not immediately contain a <merge delete specification>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.53 F321, User authorization

V0089

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<value specification> ::=
<literal>
| <general value specification>

<unsigned value specification> ::=
<unsigned literal>
| <general value specification>

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

Conformance Rules

Without Feature F321, "User authorization", conforming SQL language shall not contain a
<general value specification> that contains CURRENT_USER, SYSTEM_USER, or SESSION_USER.

NOTE — Although CURRENT_USER and USER are semantically the same, without Feature F321, "User
authorization", CURRENT_USER shall be specified as USER.

Subclause 11.5, "<default clause>":

<default clause> ::=
DEFAULT <default option>

<default option> ::=
<literal>
| <datetime value function>
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
```

```
| CURRENT_PATH
| <implicitly typed value specification>

...

Conformance Rules

Without Feature F321, "User authorization", conforming SQL language shall not contain a
<default option> that contains CURRENT_USER, SESSION_USER, or SYSTEM_USER.

NOTE — Although CURRENT_USER and USER are semantically the same, without Feature F321, "User
authorization", CURRENT_USER shall be specified as USER.

Subclause 19.2, "<set session user identifier statement>":

<set session user identifier statement> ::=
SET SESSION AUTHORIZATION <value specification>

...

Conformance Rules

Without Feature F321, "User authorization", conforming SQL language shall not contain a <set
session user identifier statement>.
```

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

Transact-SQL partially supports this feature. Transact-SQL supports the **CURRENT_USER**, **SESSION_USER**, and **SYSTEM_USER** functions, but does not support a **SET SESSION AUTHORIZATION** statement.

2.1.2.54 F341, Usage tables

V0090

The [ISO/IEC9075-11:2011] specification states the following:

```
Subclause 5.13, "CHECK_CONSTRAINT_ROUTINE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.CHECK_CONSTRAINT_ROUTINE_USAGE.

Subclause 5.17, "COLUMN_COLUMN_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.COLUMN_COLUMN_USAGE.

Subclause 5.18, "COLUMN_DOMAIN_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.COLUMN_DOMAIN_USAGE.

Subclause 5.20, "COLUMN_UDT_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.COLUMN_UDT_USAGE.

Subclause 5.22, "CONSTRAINT_COLUMN_USAGE view":
```

```
Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE.

Subclause 5.23, "CONSTAINT_PERIOD_USAGE view":

ithout Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.CONSTRAINT_PERIOD_USAGE.

Subclause 5.24, "CONSTRAINT_TABLE_USAGE view":

ithout Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE.

Subclause 5.33, "KEY_COLUMN_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.KEY_COLUMN_USAGE.

Subclause 5.34, "KEY_PERIOD_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.KEY_PERIOD_USAGE.

Subclause 5.45, "ROLE_USAGE_GRANTS view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROLE_USAGE_GRANTS.

Subclause 5.47, "ROUTINE_COLUMN_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUTINE_COLUMN_USAGE.

Subclause 5.48, "ROUTINE_PERIOD_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUTINE_PERIOD_USAGE.

Subclause 5.50, "ROUTINE_ROUTINE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUTINE_ROUTINE_USAGE.

Subclause 5.51, "ROUTINE_SEQUENCE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUTINE_SEQUENCE_USAGE.

Subclause 5.52, "ROUTINE_TABLE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUTINE_TABLE_USAGE.

Subclause 5.67, "TRIGGER_COLUMN_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIGGER_COLUMN_USAGE.

Subclause 5.68, "TRIGGER_PERIOD_USAGE view":
```

```
Without Feature F341, “Usage tables”, conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIGGER_PERIOD_USAGE.

Subclause 5.69, "TRIGGER_ROUTINE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIGGER_ROUTINE_USAGE.

Subclause 5.70, "TRIGGER_SEQUENCE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIGGER_SEQUENCE_USAGE.

Subclause 5.71, "TRIGGER_TABLE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIGGER_TABLE_USAGE.

Subclause 5.76, "VIEW_COLUMN_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.VIEW_COLUMN_USAGE.

Subclause 5.77, "VIEW_PERIOD_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.VIEW_PERIOD_USAGE.

Subclause 5.78, "VIEW_ROUTINE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.VIEW_ROUTINE_USAGE.

Subclause 5.79, "VIEW_TABLE_USAGE view":

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.VIEW_TABLE_USAGE.

Subclause 5.81, "Short name views":

Without Feature F341, "Usage tables", conforming SQL language shall not reference the
INFORMATION_SCHEMA.TRIG_TABLE_USAGE view.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIG_UPDATE_COLS.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.COL_DOMAIN_USAGE.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.CONST_COL_USAGE.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.CONST_TABLE_USAGE.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.KEY_COLUMN_USAGE_S.
```

```
Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUTINE_COL_USAGE.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUT_TABLE_USAGE.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUT_ROUT_USAGE_S.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.CONSTR_ROUT_USE_S.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIG_ROUT_USAGE_S.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.ROUT_SEQ_USAGE_S.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIG_COLUMN_USAGE.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.TRIG_SEQ_USAGE_S.

Without Feature F341, "Usage tables", conforming SQL language shall not reference
INFORMATION_SCHEMA.COL_COL_USAGE.
```

The [ISO/IEC9075-11:2008] specification excerpt differs from the [ISO/IEC9075-11:2011] specification in that the following subclauses are absent from the [ISO/IEC9075-11:2008] specification:

```
Subclause 5.23, "CONSTAINT_PERIOD_USAGE view"
Subclause 5.34, "KEY_PERIOD_USAGE view"
Subclause 5.48, "ROUTINE_PERIOD_USAGE view"
Subclause 5.68, "TRIGGER_PERIOD_USAGE view"
Subclause 5.77, "VIEW_PERIOD_USAGE view"
```

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

Transact-SQL does not support this feature.

2.1.2.55 F361, Subprogram support

V0091

The specification states the following:

```
Subclause 21.1, "<embedded SQL host program>":

<host variable definition> ::=
<Ada variable definition>
| <C variable definition>
| <COBOL variable definition>
| <Fortran variable definition>
| <MUMPS variable definition>
| <Pascal variable definition>
| <PL/I variable definition>
```

```
...

Conformance Rules

Without Feature F361, "Subprogram support", conforming SQL language shall not contain two
<host variable definition>s that specify the same variable name.
```

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

Transact-SQL does not support this feature.

2.1.2.56 F381, Extended schema manipulation

V0092

The specification states the following:

```
Subclause 11.2, "<drop schema statement>":

<drop schema statement> ::=
DROP SCHEMA <schema name> <drop behavior>

<drop behavior> ::=
CASCADE
| RESTRICT

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain a <drop schema statement>.

Subclause 11.12, "<alter column definition>":

<alter column definition> ::=
ALTER [ COLUMN ] <column name> <alter column action>

<alter column action> ::=
<set column default clause>
| <drop column default clause>
| <add column scope clause>
| <drop column scope clause>
| <alter column data type clause>
| <alter identity column specification>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain an <alter column definition>.

Subclause 11.13, "<set column default clause>":
```

```
<set column default clause> ::=
SET <default clause>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain a <set column default clause>.

Subclause 11.14, "<drop column default clause>":

<drop column default clause> ::=
DROP DEFAULT

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain a <drop column default clause>.

Subclause 11.17, "<add column scope clause>":

<add column scope clause> ::=
ADD <scope clause>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain an <add column scope clause>.

Subclause 11.18, "<drop column scope clause>":

<drop column scope clause> ::=
DROP SCOPE <drop behavior>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain a <drop column scope clause>.

Subclause 11.24, "<add table constraint definition>":

<add table constraint definition> ::=
ADD <table constraint definition>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain an <add table constraint definition>.

Subclause 11.26, "<drop table constraint definition>":
```

```
<drop table constraint definition> ::=
DROP CONSTRAINT <constraint name> <drop behavior>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain a <drop table constraint definition>.

Subclause 11.61, "<alter routine statement>":

<alter routine statement> ::=
ALTER <specific routine designator>
<alter routine characteristics> <alter routine behavior>

<alter routine characteristics> ::=
<alter routine characteristic>...

<alter routine characteristic> ::=
<language clause>
| <parameter style clause>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| NAME <external routine name>

<alter routine behavior> ::=
RESTRICT

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain an <alter routine statement>.
```

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

Transact-SQL partially supports this feature.

2.1.2.57 F381-01, ALTER TABLE statement: ALTER COLUMN clause

V0093

The specification states the following:

```
Subclause 11.12, "<alter column definition>":

<alter column definition> ::=
ALTER [ COLUMN ] <column name> <alter column action>

<alter column action> ::=
<set column default clause>
| <drop column default clause>
| <add column scope clause>
```

```
| <drop column scope clause>
| <alter column data type clause>
| <alter identity column specification>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain an <alter column definition>.
```

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

Transact-SQL partially supports this feature. An **ALTER COLUMN** clause does exist in Transact-SQL, but the syntax of the clause is significantly different from the **ALTER COLUMN** clause in the ISO standard.

2.1.2.58 F381-03, ALTER TABLE statement: DROP CONSTRAINT clause

V0093

The specification states the following:

```
Subclause 11.26, "<drop table constraint definition>":

<drop table constraint definition> ::=
DROP CONSTRAINT <constraint name> <drop behavior>

...

Conformance Rules

Without Feature F381, "Extended schema manipulation", conforming SQL language shall not
contain a <drop table constraint definition>.
```

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

Transact-SQL partially supports this feature. The **DROP CONSTRAINT** clause in Transact-SQL does not support drop behavior.

2.1.2.59 F383, Set column not null clause

V0095

The specification states the following:

```
Subclause 11.15, "<set column not null clause>":

<set column not null clause> ::=
SET NOT NULL

...

Conformance Rules:
```

```
Without Feature F383, "Set column not null clause", conforming SQL language shall not contain
a <set column not null clause>.

Subclause 11.16, "<drop column not null clause>":

<drop column not null clause> ::=
DROP NOT NULL

...

Conformance Rules:

Without Feature F383, "Set column not null clause", conforming SQL language shall not contain
a <drop column not null clause>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQLdoes not support this feature. See the **ALTER TABLE** function for equivalent functionality.

2.1.2.60 F384, Drop identity property clause

V0096

The specification states the following:

```
Subclause 11.21, "<drop identity property clause>":

<drop identity property clause> ::=
DROP IDENTITY

...

Conformance Rules:

Without Feature F384, "Drop identity property clause", conforming SQL language shall not
contain a <drop identity property clause>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.61 F385, Drop column generation expression clause

V0097

The specification states the following:

```
Subclause 11.22, "<drop column generation expression clause>":

<drop column generation expression clause> ::=
DROP EXPRESSION

...

Conformance Rules:

Without Feature F385, "Drop column generation expression clause", conforming SQL language
shall not contain a <drop column generation expression clause>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.62 F386, Set identity column generation clause

V0098

The specification states the following:

```
Subclause 11.20, "<alter identity column specification>":

<alter identity column specification> ::=
<set identity column generation clause> [ <alter identity column option>... ]
| <alter identity column option>...

<set identity column generation clause> ::=
SET GENERATED { ALWAYS | BY DEFAULT }

<alter identity column option> ::=
<alter sequence generator restart option>
| SET <basic sequence generator option>

...

Conformance Rules:

Without Feature F386, "Set identity column generation clause", conforming SQL language shall
not contain a <set identity column generation clause>.
```

Microsoft SQL Server 2008 R2varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.63 F392, Unicode escapes in identifiers

V0099

The specification states the following:

```
Subclause 5.2, "<token> and <separator>":

<Unicode delimited identifier> ::=
U<ampersand><double quote> <Unicode delimiter body> <double quote>
<Unicode escape specifier>

<Unicode escape specifier> ::=
[ UESCAPE <quote><Unicode escape character><quote> ]

<Unicode delimiter body> ::=
<Unicode identifier part>...

<Unicode identifier part> ::=
<delimited identifier part>
| <Unicode escape value>

<Unicode escape value> ::=
<Unicode 4 digit escape value>
| <Unicode 6 digit escape value>
| <Unicode character escape value>

...

Conformance Rules

Without Feature F392, "Unicode escapes in identifiers", conforming SQL language shall not
contain a <Unicode delimited identifier>.
```

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

Transact-SQL does not support this feature. However, Transact-SQL does support Unicode characters in identifiers.

2.1.2.64 F393, Unicode escapes in literals

V0100

The specification states the following:

```
Subclause 5.3, "<literal>":

<Unicode character string literal> ::=
[ <introducer><character set specification> ]
U<ampersand><quote> [ <Unicode representation>... ] <quote>
[ { <separator> <quote> [ <Unicode representation>... ] <quote> }... ]
<Unicode escape specifier>

<Unicode representation> ::=
<character representation>
| <Unicode escape value>
```

```
...

Conformance Rules

Without Feature F393, "Unicode escapes in literals", conforming SQL language shall not
contain a <Unicode character string literal>.
```

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

Transact-SQL does not support this feature. However, Transact-SQL does support Unicode characters in string literals.

2.1.2.65 F394, Optional normal form specification

V0101

The specification states the following:

```
Subclause 6.30, "<string value function>":

<string value function> ::=
<character value function>
| <binary value function>

<character value function> ::=
<character substring function>
| <regular expression substring function>
| <regex substring function>
| <fold>
| <transcoding>
| <character transliteration>
| <regex transliteration>
| <trim function>
| <character overlay function>
| <normalize function>
| <specific type method>

...

<normalize function> ::=
NORMALIZE <left paren> <character value expression>
[ <comma> <normal form> [ <comma> <normalize function result length> ] ] <right paren>

<normal form> ::=
NFC
| NFD
| NFKC
| NFKD

...

Conformance Rules
```

```
Without Feature F394, "Optional normal form specification", conforming SQL language shall not
contain <normal form>.

Subclause 8.12, "<normalized predicate>":

<normalized predicate> ::=
<row value predicand> <normalized predicate part 2>

<normalized predicate part 2> ::=
IS [ NOT ] [ <normal form> ] NORMALIZED

...

Conformance Rules

Without Feature F394, "Optional normal form specification", conforming SQL language shall not
contain <normal form>.
```

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

Transact-SQL does not support this feature.

2.1.2.66 F401, Extended joined table

V0102

The specification states the following:

```
Subclause 7.7, "<joined table>":

<joined table> ::=
<cross join>
| <qualified join>
| <natural join>

<cross join> ::=
<table reference> CROSS JOIN <table factor>
...

<natural join> ::=
{ <table reference> | <partitioned join table> }
NATURAL [ <join type> ] JOIN
{ <table factor> | <partitioned join table> }

...

<join type> ::=
INNER
| <outer join type> [ OUTER ]

<outer join type> ::=
LEFT
| RIGHT
| FULL

...
```

```
Conformance Rules

Without Feature F401, "Extended joined table", conforming SQL language shall not contain a
<cross join>.

Without Feature F401, "Extended joined table", conforming SQL language shall not contain a
<natural join>.

Without Feature F401, "Extended joined table", conforming SQL language shall not contain
FULL.
```

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

Transact-SQL partially supports this feature.

2.1.2.67 F401-01, NATURAL JOIN

V0103

The specification states the following:

```
Subclause 7.7, "<joined table>":

<natural join> ::=
{ <table reference> | <partitioned join table> }
NATURAL [ <join type> ] JOIN
{ <table factor> | <partitioned join table> }
```

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

Transact-SQL does not support this feature.

2.1.2.68 F402, Named column joins for LOBs, arrays, and multisets

V0104

The specification states the following:

```
Subclause 7.7, "<joined table>":

<joined table> ::=
<cross join>
| <qualified join>
| <natural join>

...

<qualified join> ::=
{ <table reference> | <partitioned join table> }
[ <join type> ] JOIN
{ <table reference> | <partitioned join table> }
<join specification>

...
```

```
<natural join> ::=
{ <table reference> | <partitioned join table> }
NATURAL [ <join type> ] JOIN
{ <table factor> | <partitioned join table> }

<join specification> ::=
<join condition>
| <named columns join>

...

<named columns join> ::=
USING <left paren> <join column list> <right paren>

...

Conformance Rules

Without Feature F402, "Named column joins for LOBs, arrays, and multisets", conforming SQL
language shall not contain a <joined table> that simply contains either <natural join> or
<named columns join> in which, if C is a corresponding join column, the declared type of C is
LOB-ordered, array-ordered, or multiset-ordered.

NOTE — If C is a corresponding join column, then the Conformance Rules of Subclause 9.11,
"Equality operations", also apply.

Subclause 9.11, "Equality operations":

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an equality operation shall not be ST-ordered.

Without Feature T042, "Extended LOB data type support", in conforming SQL language, the
declared type of an operand of an equality operation shall not be LOB-ordered.

Without Feature S275, "Advanced multiset support", in conforming SQL language, the declared
type of an operand of an equality operation shall not be multiset-ordered.

NOTE — If the declared type of an operand OP of an equality operation is a multiset type,
then OP is a multiset operand of a multiset element grouping operation. The Conformance Rules
of Subclause 9.13, "Multiset element grouping operations", apply.

Subclause 9.13, "Multiset element grouping operations":

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
element type of a multiset operand of a multiset element grouping operation shall not be ST-
ordered.
```

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

Transact-SQL does not support this feature.

2.1.2.69 F403, Partitioned join tables

V0105

The specification states the following:

```
Subclause 7.7, "<joined table>":
```

```
<joined table> ::=
<cross join>
| <qualified join>
| <natural join>

...

<qualified join> ::=
{ <table reference> | <partitioned join table> }
[ <join type> ] JOIN
{ <table reference> | <partitioned join table> }
<join specification>

<partitioned join table> ::=
<table factor> PARTITION BY
<partitioned join column reference list>

...

Conformance Rules

Without Feature F403, "Partitioned join tables", conforming SQL language shall not contain
<partitioned join table>.
```

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

Transact-SQL does not support this feature.

2.1.2.70 F411, Time zone specification

V0106

The specification states the following:

```
Subclause 5.3, "<literal>":

<time zone interval> ::=
<sign> <hours value> <colon> <minutes value>

...

Conformance Rules

Without Feature F411, "Time zone specification", conforming SQL language shall not contain a
<time zone interval>.

Subclause 6.1, "<data type>":

<with or without time zone> ::=
WITH TIME ZONE
| WITHOUT TIME ZONE

...

Conformance Rules
```

```
Without Feature F411, "Time zone specification", conforming SQL language shall not contain
<with or without time zone>.

Subclause 6.28, "<numeric value function>":

<extract expression> ::=
EXTRACT <left paren> <extract field> FROM <extract source> <right paren>

<extract field> ::=
<primary datetime field>
| <time zone field>

...

Conformance Rules

Feature F411, "Time zone specification", conforming SQL language shall not contain an
<extract expression> that specifies a <time zone field>.


Subclause 6.31, "<datetime value expression>":

<time zone> ::=
AT <time zone specifier>

...

Conformance Rules

Without Feature F411, "Time zone specification", conforming SQL language shall not contain a
<time zone>.

Subclause 6.32, "<datetime value function>":

<current time value function> ::=
CURRENT_TIME [ <left paren> <time precision> <right paren> ]

...

<current timestamp value function> ::=
CURRENT_TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]

...

Conformance Rules

Without Feature F411, "Time zone specification", conforming SQL language shall not contain a
<current time value function>.

Without Feature F411, "Time zone specification", conforming SQL language shall not contain a
<current timestamp value function>.

Subclause 19.4, "<set local time zone statement>":

<set local time zone statement> ::=
SET TIME ZONE <set time zone value>

<set time zone value> ::=
```

```
<interval value expression>
| LOCAL

...

Conformance Rules

Without Feature F411, "Time zone specification", conforming SQL language shall not contain a
<set local time zone statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.71 F421, National character

V0107

The specification states the following:

```
Subclause 5.3, "<literal>":

<national character string literal> ::=
N <quote> [ <character representation>... ]
<quote> [ { <separator> <quote> [ <character representation>... ] <quote> }... ]

...

Conformance Rules

Without Feature F421, "National character", conforming SQL language shall not contain a
<national character string literal>.

Subclause 6.1, "<data type>":

<national character string type> ::=
NATIONAL CHARACTER [ <left paren> <character length> <right paren> ]
| NATIONAL CHAR [ <left paren> <character length> <right paren> ]
| NCHAR [ <left paren> <character length> <right paren> ]
| NATIONAL CHARACTER VARYING <left paren> <character length> <right paren>
| NATIONAL CHAR VARYING <left paren> <character length> <right paren>
| NCHAR VARYING <left paren> <character length> <right paren>
| <national character large object type>

<national character large object type> ::=
NATIONAL CHARACTER LARGE OBJECT [ <left paren> <character large object length> <right
paren> ]
| NCHAR LARGE OBJECT [ <left paren> <character large object length> <right paren> ]
| NCLOB [ <left paren> <character large object length> <right paren> ]

...

Conformance Rules

Without Feature F421, "National character", conforming SQL language shall not contain a
<national character string type>.
```

```
Subclause 6.13, "<cast specification>":

<cast specification> ::=
CAST <left paren> <cast operand> AS <cast target> <right paren>

<cast operand> ::=
<value expression>
| <implicitly typed value specification>

<cast target> ::=
<domain name>
| <data type>

...

Conformance Rules

Without Feature F421, "National character", conforming SQL language shall not contain a <cast
operand> whose declared type is NATIONAL CHARACTER LARGE OBJECT.

Subclause 6.28, "<numeric value function>":

<length expression> ::=
<char length expression>
| <octet length expression>

<char length expression> ::=
{ CHAR_LENGTH | CHARACTER_LENGTH } <left paren> <character value expression>
[ USING <char length units> ] <right paren>

<octet length expression> ::=
OCTET_LENGTH <left paren> <string value expression> <right paren>

...

Conformance Rules

Without Feature F421, "National character", conforming SQL language shall not contain a
<length expression> that simply contains a <string value expression> that has a declared type
of NATIONAL CHARACTER LARGE OBJECT.

Subclause 6.29, "<string value expression>":

<string value expression> ::=
<character value expression>
| <binary value expression>

<character value expression> ::=
<concatenation>
| <character factor>

Subclause 8.5, "<like predicate>":

<like predicate> ::=
<character like predicate>
| <octet like predicate>

<character like predicate> ::=
<row value predicand> <character like predicate part 2>
```

```
<character like predicate part 2> ::=
[ NOT ] LIKE <character pattern> [ ESCAPE <escape character> ]

<character pattern> ::=
<character value expression>

<escape character> ::=
<character value expression>

...

Conformance Rules

Without Feature F421, "National character", and Feature T042, "Extended LOB data type
support", in conforming SQL language, a <character value expression> simply contained in a
<like predicate> shall not be of declared type NATIONAL CHARACTER LARGE OBJECT.
```

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

Transact-SQL partially supports this feature, including type declarations with ISO syntax, and national character string literals. See the **NCHAR** [MSDN-NCHAR], **NVARCHAR**, and **NTEXT** [MSDN-ntexttextimage] data types for equivalent functionality. Note that **NTEXT** is deprecated in Transact-SQL.

2.1.2.72 F431, Read-only scrollable cursors

V0108:

The specification states the following:

```
Subclause 14.2, "<cursor properties>":

<cursor properties> ::=
[ <cursor sensitivity> ] [ <cursor scrollability> ] CURSOR
[ <cursor holdability> ]
[ <cursor returnability> ]

...

<cursor scrollability> ::=
SCROLL
| NO SCROLL

...

Conformance Rules

Without Feature F431, "Read-only scrollable cursors", conforming SQL language shall not
contain a <cursor scrollability>.

Subclause 14.5, "<fetch statement>":

<fetch statement> ::=
FETCH [ [ <fetch orientation> ] FROM ] <cursor name> INTO <fetch target list>

<fetch orientation> ::=
```

```
NEXT
| PRIOR
| FIRST
| LAST
| { ABSOLUTE | RELATIVE } <simple value specification>

...

Conformance Rules

Without Feature F431, "Read-only scrollable cursors", a <fetch statement> shall not contain a
<fetch orientation>.
```

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

Transact-SQL partially supports this feature. The cursor scrollability argument **NO SCROLL** is not explicitly supported.

See [ISO/IEC9075-2:2011] for the definition of "cursor".

2.1.2.73 F441, Extended set function support

V0109

The specification states the following:

```
Subclause 6.26, "<value expression>":

<value expression> ::=
<common value expression>
| <boolean value expression>
| <row value expression>

Subclause 7.8, "<where clause>":

<where clause> ::=
WHERE <search condition>

...

Conformance Rules

Without Feature F441, "Extended set function support", conforming SQL language shall not
contain a <value expression> directly contained in a <where clause> that contains a <column
reference> that references a <derived column> that generally contains a <set function
specification> without an intervening <routine invocation>.

Subclause 6.7, "<column reference>":

<column reference> ::=
<basic identifier chain>
| MODULE <period> <qualified identifier> <period> <column name>

Subclause 7.12, "<query specification>":

<derived column> ::=
<value expression> [ <as clause> ]
```

```
Subclause 6.9, "<set function specification>":

<set function specification> ::=
<aggregate function>
| <grouping operation>

Subclause 10.4, "<routine invocation>":

<routine invocation> ::=
<routine name> <SQL argument list>

Subclause 10.9, "<aggregate function>":

<aggregate function> ::=
COUNT <left paren> <asterisk> <right paren> [ <filter clause> ]
| <general set function> [ <filter clause> ]
| <binary set function> [ <filter clause> ]
| <ordered set function> [ <filter clause> ]
| <array aggregate function> [ <filter clause> ]

<general set function> ::=
<set function type> <left paren> [ <set quantifier> ]
<value expression> <right paren>

...

<computational operation> ::=
AVG
| MAX
| MIN
| SUM
| EVERY
| ANY
| SOME
| COUNT
| STDDEV_POP
| STDDEV_SAMP
| VAR_SAMP
| VAR_POP
| COLLECT
| FUSION
| INTERSECTION

<set quantifier> ::=
DISTINCT
| ALL


...

<binary set function> ::=
<binary set function type> <left paren> <dependent variable expression> <comma>
<independent variable expression> <right paren>

...

<dependent variable expression> ::=
<numeric value expression>
```

```
<independent variable expression> ::=
<numeric value expression>

...

Conformance Rules

Without Feature F441, "Extended set function support", conforming SQL language shall not
contain a <general set function> that contains a <computational operation> that immediately
contains COUNT and does not contain a <set quantifier> that immediately contains DISTINCT.

Without Feature F441, "Extended set function support", conforming SQL language shall not
contain a <general set function> that does not contain a <set quantifier> that immediately
contains DISTINCT and that contains a <value expression> that contains a column reference
that does not reference a column of T.

Without Feature F441, "Extended set function support", conforming SQL language shall not
contain a <binary set function> that does not contain either a <dependent variable
expression> or an <independent variable expression> that contains a column reference that
references a column of T.

Without Feature F441, "Extended set function support", conforming SQL language shall not
contain a <value expression> simply contained in a <general set function> that contains a
column reference that is an outer reference where the <value expression> is not a column
reference.

Without Feature F441, "Extended set function support", conforming SQL language shall not
contain a <numeric value expression> simply contained in a <dependent variable expression> or
an <independent variable expression> that contains a column reference that is an outer
reference and in which the <numeric value expression> is not a column reference.

Without Feature F441, "Extended set function support", conforming SQL language shall not
contain a column reference contained in an <aggregate function> that contains a reference to
a column derived from a <value expression> that generally contains an <aggregate function>
SFS2 without an intervening <routine invocation>.

Subclause 6.27, "<numeric value expression>":

<numeric value expression> ::=
<term>
| <numeric value expression> <plus sign> <term>
| <numeric value expression> <minus sign> <term>
```

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

Transact-SQL partially supports this feature. For more details, see feature T621 (section 2.1.2.256).

2.1.2.74 F442, Mixed column references in set functions

V0110

The specification states the following:

```
Subclause 10.9, "<aggregate function>":

<aggregate function> ::=
COUNT <left paren> <asterisk> <right paren> [ <filter clause> ]
| <general set function> [ <filter clause> ]
```

```
| <binary set function> [ <filter clause> ]
| <ordered set function> [ <filter clause> ]
| <array aggregate function> [ <filter clause> ]

<general set function> ::=
<set function type> <left paren> [ <set quantifier> ]
<value expression> <right paren>

...
<binary set function> ::=
<binary set function type> <left paren> <dependent variable expression> <comma>
<independent variable expression> <right paren>

...

<dependent variable expression> ::=
<numeric value expression>

<independent variable expression> ::=
<numeric value expression>

...

<hypothetical set function value expression list> ::=
<value expression> [ { <comma> <value expression> }... ]

<inverse distribution function> ::=
<inverse distribution function type> <left paren>
<inverse distribution function argument> <right paren>
<within group specification>

<inverse distribution function argument> ::=
<numeric value expression>

<inverse distribution function type> ::=
PERCENTILE_CONT
| PERCENTILE_DISC

...

Conformance Rules

Without Feature F442, "Mixed column references in set functions", conforming SQL language
shall not contain a <hypothetical set function value expression list> or a <sort
specification list> that simply contains a <value expression> that contains more than one
column reference, one of which is an outer reference.

Without Feature F442, "Mixed column references in set functions", conforming SQL language
shall not contain an <inverse distribution function> that contains an <inverse distribution
function argument> or a <sort specification> that contains more than one column reference,
one of which is an outer reference.

Without Feature F442, "Mixed column references in set functions", conforming SQL language
shall not contain an <aggregate function> that contains a <general set function> whose simply
contained <value expression> contains more than one column reference, one of which is an
outer reference.

Without Feature F442, "Mixed column references in set functions", conforming SQL language
shall not contain an <aggregate function> that contains a <binary set function> whose simply
```

```
contained <dependent variable expression> or <independent variable expression> contains more
than one column reference, one of which is an outer reference.


Subclause 10.10, "<sort specification list>":

<sort specification list> ::=
<sort specification> [ { <comma> <sort specification> }... ]

<sort specification> ::=
<sort key> [ <ordering specification> ] [ <null ordering> ]

<sort key> ::=
<value expression>

<ordering specification> ::=
ASC
| DESC

<null ordering> ::=
NULLS FIRST
| NULLS LAST
```

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

Transact-SQL does not support this feature.

2.1.2.75 F451, Character set definition

V0111

The specification states the following:

```
Subclause 11.41, "<character set definition>":

<character set definition> ::=
CREATE CHARACTER SET <character set name> [ AS ]
<character set source> [ <collate clause> ]

<character set source> ::=
GET <character set specification>


...

Conformance Rules

Without Feature F451, "Character set definition", conforming SQL language shall not contain a
<character set definition>.

Subclause 11.42, "<drop character set statement>":

<drop character set statement> ::=
DROP CHARACTER SET <character set name>

...
```

```
Conformance Rules

Without Feature F451, "Character set definition", conforming SQL language shall not contain a
<drop character set statement>.
```

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

Transact-SQL partially supports this feature.

2.1.2.76 F461, Named character sets

V0112

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<character set name> ::=
[ <schema name> <period> ] <SQL language identifier>

...

Conformance Rules

Without Feature F461, "Named character sets", conforming SQL language shall not contain a
<character set name>.

Subclause 10.5, "<character set specification>":

<character set specification> ::=
<standard character set name>
| <implementation-defined character set name>
| <user-defined character set name>

<standard character set name> ::=
<character set name>

...

Conformance Rules

Without Feature F461, "Named character sets", conforming SQL language shall not contain a
<character set specification>.

Subclause 11.1, "<schema definition>":

<schema character set specification> ::=
DEFAULT CHARACTER SET <character set specification>

...

Conformance Rules

Without Feature F461, "Named character sets", conforming SQL language shall not contain a
<schema character set specification>.

Subclause 13.2, "<module name clause>":
```

```
<module character set specification> ::=
NAMES ARE <character set specification>

...

Conformance Rules

Without Feature F461, "Named character sets", conforming SQL language shall not contain a
<module character set specification>.

Subclause 19.7, "<set names statement>":

<set names statement> ::=
SET <character set name characteristic>

<character set name characteristic> ::=
NAMES <value specification>

...

Conformance Rules

Without Feature F461, "Named character sets", conforming SQL language shall not contain a
<set names statement>.

Subclause 21.1, "<embedded SQL host program>":

<embedded character set declaration> ::=
SQL NAMES ARE <character set specification>

...

Conformance Rules

Without Feature F461, "Named character sets", conforming SQL language shall not contain an
<embedded character set declaration>.
```

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

Transact-SQL does not support this feature.

2.1.2.77 F492, Optional table constraint enforcement

V0113

The specification states the following:

```
Subclause 10.8, "<constraint name definition> and <constraint characteristics>”:

<constraint name definition> ::=
CONSTRAINT <constraint name>

<constraint characteristics> ::=
<constraint check time> [ [ NOT ] DEFERRABLE ] [ <constraint enforcement> ]
| [ NOT ] DEFERRABLE [ <constraint check time> ] [ <constraint enforcement> ]
| <constraint enforcement>
```

```
<constraint check time> ::=
INITIALLY DEFERRED
| INITIALLY IMMEDIATE

<constraint enforcement> ::=
[ NOT ] ENFORCED

...

Conformance Rules:

Without Feature F492, "Optional table constraint enforcement", conforming SQL language shall
not contain a <constraint characteristics>, that specifies <constraint enforcement>.

Subclause 11.25, "<alter table constraint definition>":

<alter table constraint definition> ::=
ALTER CONSTRAINT <constraint name> <constraint enforcement>

...

Conformance Rules:

Without Feature F492, "Optional table constraint enforcement", conforming SQL language shall
not contain an <alter table constraint definition> that contains a <constraint enforcement>.
```

Microsoft SQL Server 2008 R2varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.78 F502, Enhanced documentation tables

V0114

The [ISO/IEC9075-11:2011] specification states the following:

```
Subclause 5.57, "SQL_IMPLEMENTATION_INFO view":

Function

List the SQL-implementation information items defined in this standard and, for each of
these, indicate the value supported by the SQL-implementation.

Definition

CREATE VIEW SQL_IMPLEMENTATION_INFO AS
SELECT IMPLEMENTATION_INFO_ID, IMPLEMENTATION_INFO_NAME,
INTEGER_VALUE, CHARACTER_VALUE, COMMENTS
FROM DEFINITION_SCHEMA.SQL_IMPLEMENTATION_INFO;

GRANT SELECT ON TABLE SQL_IMPLEMENTATION_INFO
TO PUBLIC WITH GRANT OPTION;
```

```
Conformance Rules

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_IMPLEMENTATION_INFO.

Subclause 5.58, "SQL_PARTS view":

Function

List the parts of this standard, and indicate which of these the SQL-implementation supports.

Definition

CREATE VIEW SQL_PARTS AS
SELECT ID AS PART, NAME, IS_SUPPORTED, IS_VERIFIED_BY, COMMENTS
FROM DEFINITION_SCHEMA.SQL_CONFORMANCE
WHERE TYPE = 'PART';

GRANT SELECT ON TABLE SQL_PARTS
TO PUBLIC WITH GRANT OPTION;

Conformance Rules

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_PARTS.

Subclause 5.81, "Short name views":

CREATE VIEW SQL_IMPL_INFO
( IMPL_INFO_ID, IMPL_INFO_NAME, INTEGER_VALUE,
CHARACTER_VALUE, COMMENTS ) AS
SELECT IMPLEMENTATION_INFO_ID, IMPLEMENTATION_INFO_NAME, INTEGER_VALUE,
CHARACTER_VALUE, COMMENTS
FROM INFORMATION_SCHEMA.SQL_IMPLEMENTATION_INFO;

GRANT SELECT ON TABLE SQL_IMPL_INFO
TO PUBLIC WITH GRANT OPTION;

CREATE VIEW SQL_SIZING_PROFS
( SIZING_ID,   SIZING_NAME,   PROFILE_ID,
PROFILE_NAME,  REQUIRED_VALUE,  COMMENTS ) AS
SELECT SIZING_ID, SIZING_NAME, PROFILE_ID,
PROFILE_NAME, REQUIRED_VALUE, COMMENTS
FROM INFORMATION_SCHEMA.SQL_SIZING_PROFILES;

GRANT SELECT ON TABLE SQL_SIZING_PROFS
TO PUBLIC WITH GRANT OPTION;

...

Conformance Rules

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_IMPL_INFO.

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_SIZING_PROFS.
```

The following additional subclauses are present in the [ISO/IEC9075-11:2008] specification:

```
Subclause 5.54, "SQL_PACKAGES view":

Function

List the packages of this standard, and indicate which of these the SQL-implementation
supports.

Definition

CREATE VIEW SQL_PACKAGES AS
SELECT ID, NAME, IS_SUPPORTED, IS_VERIFIED_BY, COMMENTS
FROM DEFINITION_SCHEMA.SQL_CONFORMANCE
WHERE TYPE = 'PACKAGE';

GRANT SELECT ON TABLE SQL_PACKAGES
TO PUBLIC WITH GRANT OPTION;

Conformance Rules

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_PACKAGES.

Subclause 5.57, "SQL_SIZING_PROFILES view":

Function

List the sizing items defined in this standard and, for each of these, indicate the size
required by one or more profiles of the standard.

Definition

CREATE VIEW SQL_SIZING_PROFILES AS
SELECT SIZING_ID, SS.SIZING_NAME, SSP.PROFILE_ID,
SSP.PROFILE_NAME, SSP.REQUIRED_VALUE, SSP.COMMENTS
FROM DEFINITION_SCHEMA.SQL_SIZING_PROFILES AS SSP
JOIN
DEFINITION_SCHEMA.SQL_SIZING AS SS
USING ( SIZING_ID );

GRANT SELECT ON TABLE SQL_SIZING_PROFILES
TO PUBLIC WITH GRANT OPTION;

Conformance Rules

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_SIZING_PROFILES.
```

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

Transact-SQL does not support this feature.

2.1.2.79 F502-01, SQL_SIZING_PROFILES view

V0115

The [ISO/IEC9075-11:2008] specification states the following:

```
Subclause 5.57, "SQL_SIZING_PROFILES view":

Function

List the sizing items defined in this standard and, for each of these, indicate the size
required by one or more profiles of the standard.

Definition

CREATE VIEW SQL_SIZING_PROFILES AS
SELECT SIZING_ID, SS.SIZING_NAME, SSP.PROFILE_ID,
SSP.PROFILE_NAME, SSP.REQUIRED_VALUE, SSP.COMMENTS
FROM DEFINITION_SCHEMA.SQL_SIZING_PROFILES AS SSP
JOIN
DEFINITION_SCHEMA.SQL_SIZING AS SS
USING ( SIZING_ID );

GRANT SELECT ON TABLE SQL_SIZING_PROFILES
TO PUBLIC WITH GRANT OPTION;

Conformance Rules

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_SIZING_PROFILES.
```

Microsoft SQL Server 2008 R2 varies as follows:

Transact-SQL does not support this feature.

Microsoft SQL Server 2012 varies as follows:

This feature is absent in the [ISO/IEC9075-11:2011] specification.

2.1.2.80 F502-02, SQL_IMPLEMENTATION_INFO view

V0116:

The [ISO/IEC9075-11:2011] specification states the following:

```
Subclause 5.57, "SQL_IMPLEMENTATION_INFO view":

Function

List the SQL-implementation information items defined in this standard and, for each of
these, indicate the value supported by the SQL-implementation.

Definition

CREATE VIEW SQL_IMPLEMENTATION_INFO AS
SELECT IMPLEMENTATION_INFO_ID, IMPLEMENTATION_INFO_NAME,
INTEGER_VALUE, CHARACTER_VALUE, COMMENTS
FROM DEFINITION_SCHEMA.SQL_IMPLEMENTATION_INFO;

GRANT SELECT ON TABLE SQL_IMPLEMENTATION_INFO
TO PUBLIC WITH GRANT OPTION;

Conformance Rules
```

```
Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_IMPLEMENTATION_INFO.
```

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

Transact-SQL does not support this feature.

2.1.2.81 F502-03, SQL_PACKAGES view

V0117:

The [ISO/IEC9075-11:2008] specification states the following:

```
Subclause 5.54, "SQL_PACKAGES view":

Function

List the packages of this standard, and indicate which of these the SQL-implementation
supports.

Definition

CREATE VIEW SQL_PACKAGES AS
SELECT ID, NAME, IS_SUPPORTED, IS_VERIFIED_BY, COMMENTS
FROM DEFINITION_SCHEMA.SQL_CONFORMANCE
WHERE TYPE = 'PACKAGE';

GRANT SELECT ON TABLE SQL_PACKAGES
TO PUBLIC WITH GRANT OPTION;

Conformance Rules

Without Feature F502, "Enhanced documentation tables", conforming SQL language shall not
reference INFORMATION_SCHEMA.SQL_PACKAGES.
```

Microsoft SQL Server 2008 R2 varies as follows:

Transact-SQL does not support this feature.

Microsoft SQL Server 2012 varies as follows:

This feature is absent in the [ISO/IEC9075-11:2011] specification.

2.1.2.82 F521, Assertions

V0118:

The specification states the following:

```
Subclause 11.47, "<assertion definition>":

<assertion definition> ::=
CREATE ASSERTION <constraint name>
CHECK <left paren> <search condition> <right paren>
[ <constraint characteristics> ]
```

```
...

Conformance Rules

Without Feature F521, "Assertions", conforming SQL language shall not contain an <assertion
definition>.

Subclause 11.48, "<drop assertion statement>":

<drop assertion statement> ::=
DROP ASSERTION <constraint name> [ <drop behavior> ]

...

Conformance Rules

Without Feature F521, "Assertions", conforming SQL language shall not contain a <drop
assertion statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.83 F531, Temporary tables

V0119:

The specification states the following:

```
Subclause 11.3, "<table definition>":

<table definition> ::=
CREATE [ <table scope> ] TABLE <table name> <table contents source>
[ ON COMMIT <table commit action> ROWS ]

<table contents source> ::=
<table element list>
| <typed table clause>
| <as subquery clause>

<table scope> ::=
<global or local> TEMPORARY

<global or local> ::=
GLOBAL
| LOCAL

...

Conformance Rules

Without Feature F531, "Temporary tables", conforming SQL language shall not contain a <table
scope> and shall not reference any global or local temporary table.

Subclause 14.16, "<temporary table declaration>":

<temporary table declaration> ::=
```

```
DECLARE LOCAL TEMPORARY TABLE <table name> <table element list>
[ ON COMMIT <table commit action> ROWS ]

...

Conformance Rules

Without Feature F531, "Temporary tables", conforming SQL language shall not contain a
<temporary table declaration>.
```

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

Transact-SQL does not support this feature. See the **#** and **##** table name prefixes for equivalent functionality.

See [ISO/IEC9075-1:2011] for the definition of "table".

2.1.2.84 F555, Enhanced seconds precision

V0120:

The specification states the following:

```
Subclause 5.3, "<literal>":

<unsigned integer> ::=
<digit>...

...

<time literal> ::=
TIME <time string>

<timestamp literal> ::=
TIMESTAMP <timestamp string>

...

<time string> ::=
<quote> <unquoted time string> <quote>

...

<seconds fraction> ::=
<unsigned integer>

...

Conformance Rules

Without Feature F555, "Enhanced seconds precision", in conforming SQL language, an <unsigned
integer> that is a <seconds fraction> that is contained in a <timestamp literal> shall not
contain more than 6 <digit>s.

Without Feature F555, "Enhanced seconds precision", in conforming SQL language, a <time
literal> shall not contain a <seconds fraction>.
```

```
Subclause 6.1, "<data type>":

<time precision> ::=
<time fractional seconds precision>

<timestamp precision> ::=
<time fractional seconds precision>

<time fractional seconds precision> ::=
<unsigned integer>

...

Conformance Rules

Without Feature F555, "Enhanced seconds precision", conforming SQL language shall not contain
a <time precision> that does not specify 0 (zero).

Without Feature F555, "Enhanced seconds precision", conforming SQL language shall not contain
a <timestamp precision> that does not specify either 0 (zero) or 6.

Subclause 6.32, "<datetime value function>":

<current local time value function> ::=
LOCALTIME [ <left paren> <time precision> <right paren> ]

...

<current local timestamp value function> ::=
LOCALTIMESTAMP [ <left paren> <timestamp precision> <right paren> ]

...

Conformance Rules

Without Feature F555, "Enhanced seconds precision", conforming SQL language shall not contain
a <current local time value function> that contains a <time precision> that is not 0 (zero).

Without Feature F555, "Enhanced seconds precision", conforming SQL language shall not contain
a <current local timestamp value function> that contains a <timestamp precision> that is
neither 0 (zero) nor 6.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **TIME** or **TIMESTAMP** keywords for literals. However, the seconds fractions in Transact-SQL date and time data types and functions have more than the precision that is required by [ISO/IEC9075-2:2011].

2.1.2.85 F561, Full value expressions

V0121:

The specification states the following:

```
Subclause 8.4, "<in predicate>":

<in value list> ::=
<row value expression> [ { <comma> <row value expression> }... ]
```

```
...

Conformance Rules

Without Feature F561, "Full value expressions", conforming SQL language shall not contain a
<row value expression> immediately contained in an <in value list> that is not a <value
specification>.

NOTE — Since <in predicate> is an equality operation, the Conformance Rules of Subclause
9.11, "Equality operations", also apply.

Subclause 9.11, "Equality operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an equality operation shall not be ST-ordered.

Without Feature T042, "Extended LOB data type support", in conforming SQL language, the
declared type of an operand of an equality operation shall not be LOB-ordered.

Without Feature S275, "Advanced multiset support", in conforming SQL language, the declared
type of an operand of an equality operation shall not be multiset-ordered.

NOTE — If the declared type of an operand OP of an equality operation is a multiset type,
then OP is a multiset operand of a multiset element grouping operation. The Conformance Rules
of Subclause 9.13, "Multiset element grouping operations", apply.

Subclause 9.13, "Multiset element grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
element type of a multiset operand of a multiset element grouping operation shall not be ST-
ordered.

Subclause 7.2, "<row value expression>":

<row value expression> ::=
<row value special case>
| <explicit row value constructor>

Subclause 6.4, "<value specification> and <target specification>":

<value specification> ::=
<literal>
| <general value specification>

...

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
```

```
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

Subclause 10.9, "<aggregate function>":

<general set function> ::=
<set function type> <left paren> [ <set quantifier> ]
<value expression> <right paren>

...

Conformance Rules

Without Feature F561, "Full value expressions", or Feature F801, "Full set function",
conforming SQL language shall not contain a <general set function> that immediately contains
DISTINCT and contains a <value expression> that is not a column reference.

Subclause 6.26, "<value expression>":

<value expression> ::=
<common value expression>
| <boolean value expression>
| <row value expression>
```

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

Transact-SQL does not support this feature.

2.1.2.86 F571, Truth value tests

V0122:

The specification states the following:

```
Subclause 6.35, "<boolean value expression>":

<boolean test> ::=
<boolean primary> [ IS [ NOT ] <truth value> ]

<truth value> ::=
TRUE
| FALSE
| UNKNOWN

<boolean primary> ::=
<predicate>
| <boolean predicand>

...
```

```
Conformance Rules

Without Feature F571, "Truth value tests", conforming SQL language shall not contain a
<boolean test> that simply contains a <truth value>.
```

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

Transact-SQL does not support this feature.

2.1.2.87 F611, Indicator data types

V0123:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<indicator variable> ::=
[ INDICATOR ] <embedded variable name>

<indicator parameter> ::=
[ INDICATOR ] <host parameter name>

...

Conformance Rules

Without Feature F611, "Indicator data types", in conforming SQL language, the declared types
of <indicator parameter>s and <indicator variable>s shall be the same implementation-defined
data type.
```

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

Transact-SQL does not support this feature.

2.1.2.88 F641, Row and table constructors

V0124:

The specification states the following:

```
Subclause 7.1, "<row value constructor>":

<explicit row value constructor> ::=
<left paren> <row value constructor element> <comma>
<row value constructor element list> <right paren>
| ROW <left paren> <row value constructor element list> <right paren>
| <row subquery>

...

<row value constructor element> ::=
<value expression>

<contextually typed row value constructor> ::=
```

```
<common value expression>
| <boolean value expression>
| <contextually typed value specification>
| <left paren> <contextually typed value specification> <right paren>
| <left paren> <contextually typed row value constructor element> <comma>
<contextually typed row value constructor element list> <right paren>
| ROW <left paren> <contextually typed row value constructor element list> <right paren>

<contextually typed row value constructor element list> ::=
<contextually typed row value constructor element>
[ { <comma> <contextually typed row value constructor element> }... ]
...

Conformance Rules

Without Feature F641, "Row and table constructors", conforming SQL language shall not contain
an <explicit row value constructor> that is not simply contained in a <table value
constructor> and that contains more than one <row value constructor element>.

Without Feature F641, "Row and table constructors", conforming SQL language shall not contain
an <explicit row value constructor> that is a <row subquery>.

Without Feature F641, "Row and table constructors", conforming SQL language shall not contain
a <contextually typed row value constructor> that is not simply contained in a <contextually
typed table value constructor> and that contains more than one <row value constructor
element>.

Without Feature F641, "Row and table constructors", conforming SQL language shall not contain
a <contextually typed row value constructor> that is a <row subquery>.

Subclause 7.15, "<subquery>":

<row subquery> ::=
<subquery>

...

<subquery> ::=
<left paren> <query expression> <right paren>


Subclause 7.3, "<table value constructor>":

<table value constructor> ::=
VALUES <row value expression list>

<row value expression list> ::=
<table row value expression> [ { <comma> <table row value expression> }... ]

<contextually typed table value constructor> ::=
VALUES <contextually typed row value expression list>

<contextually typed row value expression list> ::=
<contextually typed row value expression>
[ { <comma> <contextually typed row value expression> }... ]

...

Conformance Rules
```

```
Without Feature F641, "Row and table constructors", in conforming SQL language, the
<contextually typed row value expression list> of a <contextually typed table value
constructor> shall contain exactly one <contextually typed row value constructor> RVE. RVE
shall be of the form "(<contextually typed row value constructor element list>)".

Without Feature F641, "Row and table constructors", conforming SQL language shall not contain
a <table value constructor>.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support simple tables and has limited support for table constructors in general.

See [ISO/IEC9075-1:2011] for the definition of "table".

2.1.2.89 F651, Catalog name qualifiers

V0125:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<schema name> ::=
[ <catalog name> <period> ] <unqualified schema name>

<unqualified schema name> ::=
<identifier>

<catalog name> ::=
<identifier>

...

Conformance Rules

Without Feature F651, "Catalog name qualifiers", conforming SQL language shall not contain a
<catalog name>.

Subclause 19.5, "<set catalog statement>":

<set catalog statement> ::=
SET <catalog name characteristic>

<catalog name characteristic> ::=
CATALOG <value specification>

...

Conformance Rules

Without Feature F651, "Catalog name qualifiers", conforming SQL language shall not contain a
<set catalog statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.90 F661, Simple tables

V0126:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query primary> ::=
<simple table>
| <left paren> <query expression body>
[ <order by clause> ] [ <fetch first clause> ] <right paren>

<simple table> ::=
<query specification>
| <table value constructor>
| <explicit table>

<explicit table> ::=
TABLE <table or query name>

...

Conformance Rules

Without Feature F661, "Simple tables", conforming SQL language shall not contain a <simple
table> that immediately contains a <table value constructor> except in an <insert statement>.

Without Feature F661, "Simple tables", conforming SQL language shall not contain an <explicit
table>.

Subclause 7.3, "<table value constructor>":

<table value constructor> ::=
VALUES <row value expression list>

<row value expression list> ::=
<table row value expression> [ { <comma> <table row value expression> }... ]
```

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

Transact-SQL does not support this feature.

2.1.2.91 F672, Retrospective check constraints

V0127:

The specification states the following:

```
Subclause 11.9, "<check constraint definition>":

<check constraint definition> ::=
```

```
CHECK <left paren> <search condition> <right paren>

...

Conformance Rules

Without Feature F672, "Retrospective check constraints", conforming SQL language shall not
contain a <check constraint definition> that generally contains CURRENT_DATE,
CURRENT_TIMESTAMP, or LOCALTIMESTAMP.

Subclause 11.4, "<column definition>":

<column constraint> ::=
NOT NULL
| <unique specification>
| <references specification>
| <check constraint definition>

Subclause 11.6, "<table constraint definition>":

<table constraint> ::=
<unique constraint definition>
| <referential constraint definition>
| <check constraint definition>

Subclause 11.34, "<domain definition>":

<domain constraint> ::=
[ <constraint name definition> ] <check constraint definition> [
<constraint characteristics> ]


Subclause 11.47, "<assertion definition>":

<assertion definition> ::=
CREATE ASSERTION <constraint name>
CHECK <left paren> <search condition> <right paren>
[ <constraint characteristics> ]

...

Conformance Rules

Without Feature F672, "Retrospective check constraints", conforming SQL language shall not
contain an <assertion definition> that generally contains CURRENT_DATE, CURRENT_TIMESTAMP, or
LOCALTIMESTAMP.

Subclause 8.21, "<search condition>":

<search condition> ::=
<boolean value expression>
```

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

Transact-SQL does not support this feature.

2.1.2.92 F673, Reads SQL-data routine invocations in CHECK constraints

V0128:

The specification states the following:

```
Subclause 8.21, "<search condition>":

<search condition> ::=
<boolean value expression>

Subclause 10.4, "<routine invocation>":

<routine invocation> ::=
<routine name> <SQL argument list>

Subclause 11.9, "<check constraint definition>":

<check constraint definition> ::=
CHECK <left paren> <search condition> <right paren>

...

Conformance Rules:

Without Feature F673, "Reads SQL-data routine invocations in CHECK constraints", conforming
SQL language shall not contain a <check constraint definition> that contains a <search
condition> that generally contains a <routine invocation> whose subject routine is an SQL
routine that possibly reads SQL-data.
```

Microsoft SQL Server 2008 R2varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008]standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL partially supports this feature. The syntax differs from the standard.

2.1.2.93 F690, Collation support

V0129:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<collation name> ::=
<schema qualified name>

...

Conformance Rules

Without Feature F690, "Collation support", conforming SQL language shall not contain a
<collation name>.

Subclause 10.7, "<collate clause>":
```

```
<collate clause> ::=
COLLATE <collation name>

...

Conformance Rules

Without Feature F690, "Collation support", conforming SQL language shall not contain a
<collate clause>.

Subclause 11.43, "<collation definition>":

<collation definition> ::=
CREATE COLLATION <collation name> FOR <character set specification>
FROM <existing collation name> [ <pad characteristic> ]

<existing collation name> ::=
<collation name>

<pad characteristic> ::=
NO PAD
| PAD SPACE

...

Conformance Rules

Without Feature F690, "Collation support", conforming SQL language shall not contain a
<collation definition>.

Subclause 11.44, "<drop collation statement>":

<drop collation statement> ::=
DROP COLLATION <collation name> <drop behavior>

...

Conformance Rules

Without Feature F690, "Collation support", conforming SQL language shall not contain a <drop
collation statement>.
```

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

Transact-SQL partially supports this feature. Transact-SQL supports a **COLLATE** clause, but does not support the **CREATE COLLATION** and **DROP COLLATION** statements.

2.1.2.94 F692, Enhanced collation support

V0130:

The specification states the following:

```
Subclause 10.7, "<collate clause>":

<collate clause> ::=
COLLATE <collation name>
```

```
Subclause 11.4, "<column definition>":

<column definition> ::=
<column name> [ <data type or domain name> ]
[ <default clause> | <identity column specification> | <generation clause> ]
[ <column constraint definition>... ]
[ <collate clause> ]

...

Conformance Rules

Without Feature F692, "Extended collation support", conforming SQL language shall not contain
a <column definition> that immediately contains a <collate clause>.

Subclause 11.34, "<domain definition>":

<domain definition> ::=
CREATE DOMAIN <domain name> [ AS ] <predefined type>
[ <default clause> ]
[ <domain constraint>... ]
[ <collate clause> ]

...

Conformance Rules

Without Feature F692, "Extended collation support", conforming SQL language shall not contain
a <domain definition> that immediately contains a <collate clause>.

Subclause 11.52, "<attribute definition>":

<attribute definition> ::=
<attribute name> <data type>
[ <attribute default> ]
[ <collate clause> ]

...

Conformance Rules

Without Feature F692, "Extended collation support", conforming SQL language shall not contain
an <attribute definition> that immediately contains a <collate clause>.
```

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

Transact-SQL partially supports this feature. Transact-SQL supports the **COLLATE** clause in column definitions, but not in domain definitions or attribute definitions, which do not exist in Transact-SQL.

See [ISO/IEC9075-2:2011] for the definition of "column".

2.1.2.95 F693, SQL-session and client module collations

V0131:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<current collation specification> ::=
COLLATION FOR <left paren> <string value expression> <right paren>

...

Conformance Rules

Without Feature F693, "SQL-session and client module collations", conforming SQL language
shall not contain <current collation specification>.

Subclause 13.1, "<SQL-client module definition>":

<module collation specification> ::=
COLLATION <collation name> [ FOR <character set specification list> ]

...

Conformance Rules

Without Feature F693, "SQL-session and client module collations", conforming SQL language
shall not contain a <module collation specification>.

Subclause 19.10, "<set session collation statement>":

<set session collation statement> ::=
SET COLLATION <collation specification> [ FOR <character set specification list> ]
| SET NO COLLATION [ FOR <character set specification list> ]

<collation specification> ::=
<value specification>

...

Conformance Rules

Without Feature F693, "SQL-session and client module collations", conforming SQL language
shall not contain a <set session collation statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.96 F695, Translation support

V0132:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<transliteration name> ::=
<schema qualified name>

<transcoding name> ::=
```

```
<schema qualified name>

...

Conformance Rules

Without Feature F695, "Translation support", conforming SQL language shall not contain a
<transliteration name>.

Without Feature F695, "Translation support", conforming SQL language shall not contain a
<transcoding name>.

Subclause 6.30, "<string value function>":

<transcoding> ::=
CONVERT <left paren> <character value expression>
USING <transcoding name> <right paren>

<character transliteration> ::=
TRANSLATE <left paren> <character value expression>
USING <transliteration name> <right paren>

...

Conformance Rules

Without Feature F695, "Translation support", conforming SQL language shall not contain a
<character transliteration>.

Without Feature F695, "Translation support", conforming SQL language shall not contain a
<transcoding>.

Subclause 11.45, "<transliteration definition>":

<transliteration definition> ::=
CREATE TRANSLATION <transliteration name> FOR <source character set specification>
TO <target character set specification> FROM <transliteration source>

...

Conformance Rules

Without Feature F695, "Translation support", conforming SQL language shall not contain a
<transliteration definition>.

Subclause 11.46, "<drop transliteration statement>":

<drop transliteration statement> ::=
DROP TRANSLATION <transliteration name>

...

Conformance Rules

Without Feature F695, "Translation support", conforming SQL language shall not contain a
<drop transliteration statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.97 F701, Referential update actions

V0133:

The specification states the following:

```
Subclause 11.8, "<referential constraint definition>":

<referential triggered action> ::=
<update rule> [ <delete rule> ]
| <delete rule> [ <update rule> ]

<update rule> ::=
ON UPDATE <referential action>

<delete rule> ::=
ON DELETE <referential action>

...

Conformance Rules

Without Feature F701, "Referential update actions", conforming SQL language shall not contain
an <update rule>.
```

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

Transact-SQL partially supports this feature. See the **ON UPDATE** clause and related clauses of the **CREATE TRIGGER** [MSDN-CREATETRIGGER] and **ALTER TRIGGER** [MSDN-ALTERTRIGGER] statements for equivalent functionality.

2.1.2.98 F711, ALTER domain

V0134:

The specification states the following:

```
Subclause 11.35, "<alter domain statement>":

<alter domain statement> ::=
ALTER DOMAIN <domain name> <alter domain action>

<alter domain action> ::=
<set domain default clause>
| <drop domain default clause>
| <add domain constraint definition>
| <drop domain constraint definition>

...

Conformance Rules

Without Feature F711, "ALTER domain", conforming SQL language shall not contain an <alter
domain statement>.
```

```
Subclause 11.36, "<set domain default clause>":

<set domain default clause> ::=
SET <default clause>

...

Conformance Rules

Without Feature F711, "ALTER domain", conforming SQL language shall not contain a <set domain
default clause>.

Subclause 11.37, "<drop domain default clause>":

<drop domain default clause> ::=
DROP DEFAULT

...

Conformance Rules

Without Feature F711, "ALTER domain", conforming SQL language shall not contain a <drop
domain default clause>.

Subclause 11.38, "<add domain constraint definition>":

<add domain constraint definition> ::=
ADD <domain constraint>

...

Conformance Rules

Without Feature F711, "ALTER domain", conforming SQL language shall not contain an <add
domain constraint definition>.

Subclause 11.39, "<drop domain constraint definition>":

<drop domain constraint definition> ::=
DROP CONSTRAINT <constraint name>

...

Conformance Rules

Without Feature F711, "ALTER domain", conforming SQL language shall not contain a <drop
domain constraint definition>.
```

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

Transact-SQL does not support this feature.

2.1.2.99 F721, Deferrable constraints

V0135:

The specification states the following:

```
Subclause 10.8, "<constraint name definition> and <constraint characteristics>":

<constraint characteristics> ::=
<constraint check time> [ [ NOT ] DEFERRABLE ]
| [ NOT ] DEFERRABLE [ <constraint check time> ]

<constraint check time> ::=
INITIALLY DEFERRED
| INITIALLY IMMEDIATE

...

Conformance Rules

Without Feature F721, "Deferrable constraints", conforming SQL language shall not contain a
<constraint characteristics>, other than a <constraint enforcement>.

NOTE — This means that INITIALLY IMMEDIATE NOT DEFERRABLE is implicit.

Subclause 17.4, "<set constraints mode statement>":

<set constraints mode statement> ::=
SET CONSTRAINTS <constraint name list> { DEFERRED | IMMEDIATE }

<constraint name list> ::=
ALL
| <constraint name> [ { <comma> <constraint name> }... ]

...

Conformance Rules

Without Feature F721, "Deferrable constraints", conforming SQL language shall not contain a
<set constraints mode statement>.
```

The [ISO/IEC 9075-2:2008] specification differs as follows:

```
Subclause 10.8, "<constraint name definition> and <constraint characteristics>":

Conformance Rules

Without Feature F721, "Deferrable constraints", conforming SQL language shall not contain a
<constraint characteristics>.
```

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

Transact-SQL does not support this feature.

2.1.2.100 F741, Referential MATCH types

V0136:

The specification states the following:

```
Subclause 8.13, "<match predicate>":

<match predicate> ::=
<row value predicand> <match predicate part 2>

<match predicate part 2> ::=
MATCH [ UNIQUE ] [ SIMPLE | PARTIAL | FULL ] <table subquery>

Conformance Rules

1) Without Feature F741, "Referential MATCH types", conforming SQL language shall not contain
a <match predicate>.

NOTE — The Conformance Rules of Subclause 9.11, "Equality operations", also apply.

Subclause 9.11, "Equality operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an equality operation shall not be ST-ordered.

Without Feature T042, "Extended LOB data type support", in conforming SQL language, the
declared type of an operand of an equality operation shall not be LOB-ordered.

Without Feature S275, "Advanced multiset support", in conforming SQL language, the declared
type of an operand of an equality operation shall not be multiset-ordered.

NOTE — If the declared type of an operand OP of an equality operation is a multiset type,
then OP is a multiset operand of a multiset element grouping operation. The Conformance Rules
of Subclause 9.13, "Multiset element grouping operations", apply.

Subclause 9.13, "Multiset element grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
element type of a multiset operand of a multiset element grouping operation shall not be ST-
ordered.

Subclause 11.8, "<referential constraint definition>":

<references specification> ::=
REFERENCES <referenced table and columns>
[ MATCH <match type> ] [ <referential triggered action> ]

...

Conformance Rules

...

Without Feature F741, "Referential MATCH types", conforming SQL language shall not contain a
<references specification> that contains MATCH.
```

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

Transact-SQL does not support this feature.

2.1.2.101 F751, View CHECK enhancements

V0137:

The specification states the following:

```
Subclause 11.32, "<view definition>":

<view definition> ::=
CREATE [ RECURSIVE ] VIEW <table name> <view specification>
AS <query expression> [ WITH [ <levels clause> ] CHECK OPTION ]

...

<levels clause> ::=
CASCADED
| LOCAL

...

Conformance Rules

Without Feature F751, "View CHECK enhancements", conforming SQL language shall not contain a
<levels clause>.

...

Without Feature F751, "View CHECK enhancements", conforming SQL language shall not contain
<view definition> that contains a <query expression> that contains a <query expression> and
contains WITH CHECK OPTION.
```

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

Transact-SQL partially supports this feature. The **CASCADED** and **LOCAL** keywords are not supported, but the **WITH CHECK OPTION** clause is supported by Transact-SQL.

2.1.2.102 F761, Session management

V0138:

The specification states the following:

```
Subclause 19.1, "<set session characteristics statement>":

<set session characteristics statement> ::=
SET SESSION CHARACTERISTICS AS <session characteristic list>

<session characteristic list> ::=
<session characteristic> [ { <comma> <session characteristic> }... ]

<session characteristic> ::=
<session transaction characteristics>

<session transaction characteristics> ::=
TRANSACTION <transaction mode> [ { <comma> <transaction mode> }... ]
```

```
...

Conformance Rules

Without Feature F761, "Session management", conforming SQL language shall not contain a <set
session characteristics statement>.

Subclause 19.5, "<set catalog statement>":

<set catalog statement> ::=
SET <catalog name characteristic>

<catalog name characteristic> ::=
CATALOG <value specification>

...

Conformance Rules

Without Feature F761, "Session management", conforming SQL language shall not contain a <set
catalog statement>.

Subclause 19.6, "<set schema statement>":

<set schema statement> ::=
SET <schema name characteristic>

<schema name characteristic> ::=
SCHEMA <value specification>

...

Conformance Rules

Without Feature F761, "Session management", conforming SQL language shall not contain a <set
schema statement>.

Subclause 19.7, "<set names statement>":

<set names statement> ::=
SET <character set name characteristic>

<character set name characteristic> ::=
NAMES <value specification>

...

Conformance Rules

Without Feature F761, "Session management", conforming SQL language shall not contain a <set
names statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.103 F762, CURRENT_CATALOG

V0139:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

Conformance Rules

Without Feature F762, "CURRENT_CATALOG", conforming SQL language shall not contain a <general
value specification> that contains CURRENT_CATALOG.

Subclause 11.5, "<default clause>":

<default option> ::=
<literal>
| <datetime value function>
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
| CURRENT_PATH
| <implicitly typed value specification>

...

Conformance Rules

Without Feature F762, "CURRENT_CATALOG", conforming SQL language shall not contain a <default
option> that contains CURRENT_CATALOG.
```

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

Transact-SQL does not support this feature.

2.1.2.104 F763, CURRENT_SCHEMA

V0140:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

Conformance Rules

Without Feature F763, "CURRENT_SCHEMA", conforming SQL language shall not contain a <general
value specification> that contains CURRENT_SCHEMA.

Subclause 11.5, "<default clause>":

<default option> ::=
<literal>
| <datetime value function>
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
| CURRENT_PATH
| <implicitly typed value specification>

...

Conformance Rules

Without Feature F763, "CURRENT_SCHEMA", conforming SQL language shall not contain a <default
option> that contains CURRENT_SCHEMA.
```

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

Transact-SQL does not support this feature.

2.1.2.105 F771, Connection management

V0141:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<connection name> ::=
<simple value specification>

...

Conformance Rules

Without Feature F771, "Connection management", conforming SQL language shall not contain an
explicit <connection name>.

Subclause 13.4, "<SQL procedure statement>":

<SQL connection statement> ::=
<connect statement>
| <set connection statement>
| <disconnect statement>

Subclause 18.1, "<connect statement>":

<connect statement> ::=
CONNECT TO <connection target>

<connection target> ::=
<SQL-server name> [ AS <connection name> ] [ USER <connection user name> ]
| DEFAULT

...

Conformance Rules

Without Feature F771, "Connection management", conforming SQL language shall not contain a
<connect statement>.

Subclause 18.2, "<set connection statement>":

<set connection statement> ::=
SET CONNECTION <connection object>

<connection object> ::=
DEFAULT
| <connection name>

...

Conformance Rules
```

```
Without Feature F771, "Connection management", conforming SQL language shall not contain a
<set connection statement>.

Subclause 18.3, "<disconnect statement>":

<disconnect statement> ::=
DISCONNECT <disconnect object>

<disconnect object> ::=
<connection object>
| ALL
| CURRENT

...

Conformance Rules

Without Feature F771, "Connection management", conforming SQL language shall not contain a
<disconnect statement>.
```

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

Transact-SQL partially supports this feature. **Connection management** is supported only within embedded SQL programs. See the **USE** statement [MSDN-USE] for equivalent functionality.

2.1.2.106 F791, Insensitive cursors

V0142:

The specification states the following:

```
Subclause 14.2, "<cursor properties>":

<cursor properties> ::=
[ <cursor sensitivity> ] [ <cursor scrollability> ] CURSOR
[ <cursor holdability> ]
[ <cursor returnability> ]

<cursor sensitivity> ::=
SENSITIVE
| INSENSITIVE
| ASENSITIVE

...

Conformance Rules

Without Feature F791, "Insensitive cursors", conforming SQL language shall not contain a
<cursor sensitivity> that immediately contains INSENSITIVE.

Without Feature F791, "Insensitive cursors", or Feature T231, "Sensitive cursors", conforming
SQL language shall not contain a <cursor sensitivity> that immediately contains ASENSITIVE.
```

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

Transact-SQL partially supports this feature. The **INSENSITIVE** cursor sensitivity parameter is supported, but the **ASENSITIVE** cursor sensitivity parameter is not supported.

See [ISO/IEC9075-2:2011] for the definition of "cursor".

2.1.2.107 F813, Extended flagging

V0143:

The [ISO/IEC9075-2:2011] specification states the following:

```
Part 1, Subclause 8.5, "SQL flagger": With "level of flagging" specified to be Core SQL
Flagging and "extent of checking" specified to be Catalog Lookup
```

The [ISO/IEC9075-1:2011] specification states the following:

```
Subclause 8.5, "SQLflagger":

An SQL Flagger provides one or more of the following "level of flagging" options:

— Core SQL Flagging
— Part SQL Flagging

An SQL Flagger that provides one of these options shall be able to identify SQL language
constructs that violate the indicated subset of SQL language. The subset of SQL language used
by "Core SQL Flagging" is Core SQL....

An SQL Flagger provides one or more of the following "extent of checking" options:

— Syntax Only
— Catalog Lookup

...

Under the Catalog Lookup option, the SQL Flagger assumes the availability of Definition
Schema information and checks for violations of all Syntax Rules. For example, some Syntax
Rules place restrictions on data types; this flagger option would identify extensions that
relax such restrictions. In order to avoid security breaches, this option shall view the
Definition Schema only through the eyes of a specific Information Schema. The flagger does
not necessarily execute or simulate the execution of any <schema definition statement> or
<schema manipulation statement>.
```

The [ISO/IEC9075-1:2008] specification differs as follows:

```
An SQL Flagger provides one or more of the following "level of flagging" options:

— Core SQL Flagging
— Part SQL Flagging
— Package SQL Flagging
```

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

Transact-SQL does not support this feature.

2.1.2.108 F821, Local table references

V0144:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<local or schema qualified name> ::=
[ <local or schema qualifier> <period> ] <qualified identifier>

<local or schema qualifier> ::=
<schema name>
| <local qualifier>

...

<local qualifier> ::=
MODULE

...

Conformance Rules

Without Feature F821, "Local table references", conforming SQL language shall not contain a
<local or schema qualifier> that contains a <local qualifier>.

Subclause 6.7, "<column reference>":

<column reference> ::=
<basic identifier chain>
| MODULE <period> <qualified identifier> <period> <column name>

...

Conformance Rules

Without Feature F821, "Local table references", conforming SQL language shall not contain a
<column reference> that simply contains MODULE.
```

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

Transact-SQL does not support this feature.

2.1.2.109 F841, LIKE_REGEX predicate

V0145:

The specification states the following:

```
Subclause 8.7, "<regex like predicate>":

<regex like predicate> ::=
<row value predicand> <regex like predicate part 2>

<regex like predicate part 2> ::=
```

```
[ NOT ] LIKE_REGEX <XQuery pattern> [ FLAG <XQuery option flag> ]

...

Conformance Rules

Without Feature F841, "LIKE_REGEX predicate", conforming SQL language shall not contain
<regex like predicate>.
```

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

Transact-SQL does not support this feature.

2.1.2.110 F842, OCCURRENCES_REGEX function

V0146:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<regex occurrences function> ::=
OCCURRENCES_REGEX <left paren>
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
[ FROM <start position> ]
[ USING <char length units> ]
<right paren>

...

Conformance Rules

Without Feature F842, "OCCURENCES_REGEX function", conforming SQL language shall not contain
<regex occurrences function>.
```

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

Transact-SQL does not support this feature.

2.1.2.111 F843, POSITION_REGEX function

V0147:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<regex position expression> ::=
POSITION_REGEX <left paren>
[ <regex position start or after> ]
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
[ FROM <start position> ]
[ USING <char length units> ]
```

```
[ OCCURRENCE <regex occurrence> ]
[ GROUP <regex capture group> ]
<right paren>

...

Conformance Rules

Without Feature F843, "POSITION_REGEX function", conforming SQL language shall not contain
<regex position expression>.
```

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

Transact-SQL does not support this feature.

2.1.2.112 F844, SUBSTRING_REGEX function

V0148:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<regex substring function> ::=
SUBSTRING_REGEX <left paren>
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
[ FROM <start position> ]
[ USING <char length units> ]
[ OCCURRENCE <regex occurrence> ]
[ GROUP <regex capture group> ]
<right paren>

...

Conformance Rules

Without Feature F844, "SUBSTRING_REGEXP", conforming SQL language shall not contain <regex
substring function>.
```

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

Transact-SQL does not support this feature.

2.1.2.113 F845, TRANSLATE_REGEX function

V0149:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<regex transliteration> ::=
TRANSLATE_REGEX <left paren>
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
```

```
[ WITH <XQuery replacement string> ]
[ FROM <start position> ]
[ USING <char length units> ]
[ OCCURRENCE <regex transliteration occurrence> ]
<left paren>

...

Conformance Rules

Without Feature F845, "TRANSLATE_REGEX", conforming SQL language shall not contain <regex
transliteration>.
```

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

Transact-SQL does not support this feature.

2.1.2.114 F846, Octet support in regular expression operators

V0150:

The specification states the following:

```
Subclause 6.1, "<data type>"

<char length units> ::=
CHARACTERS
| OCTETS

Subclause 6.28, "<numeric value function>":

<regex occurrences function> ::=
OCCURRENCES_REGEX <left paren>
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
[ FROM <start position> ]
[ USING <char length units> ]
<right paren>

...

<regex position expression> ::=
POSITION_REGEX <left paren>
[ <regex position start or after> ]
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
[ FROM <start position> ]
[ USING <char length units> ]
[ OCCURRENCE <regex occurrence> ]
[ GROUP <regex capture group> ]
<right paren>

...

Conformance Rules
```

```
Without Feature F846, "Octet support in regular expression operators", in conforming SQL
language, <regex occurrences function> shall not contain <char length units>.

...

Without Feature F846, "Octet support in regular expression operators", in conforming SQL
language , <regex position expression> shall not contain <char length units>.

Subclause 6.30, "<string value function>":

<regex substring function> ::=
SUBSTRING_REGEX <left paren>
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
[ FROM <start position> ]
[ USING <char length units> ]
[ OCCURRENCE <regex occurrence> ]
[ GROUP <regex capture group> ]
<right paren>

...

<regex transliteration> ::=
TRANSLATE_REGEX <left paren>
<XQuery pattern> [ FLAG <XQuery option flag> ]
IN <regex subject string>
[ WITH <XQuery replacement string> ]
[ FROM <start position> ]
[ USING <char length units> ]
[ OCCURRENCE <regex transliteration occurrence> ]
<left paren>

...

Conformance Rules

Without Feature F846, "Octet support in regular expression operators", in conforming SQL
language, <regex substring function> shall not contain <char length units>.

...

Without Feature F846, "Octet support in regular expression operators", in conforming SQL
language, <regex transliteration> shall not contain <char length units>.
```

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

Transact-SQL does not support this feature.

2.1.2.115 F847, Nonconstant regular expressions

V0151:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<value specification> ::=
<literal>
```

```
| <general value specification>

...

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

Subclause 6.28, "<numeric value function>":

<XQuery pattern> ::=
<character value expression>

<XQuery option flag> ::=
<character value expression>

...

Conformance Rules

Without Feature F847, "Nonconstant regular expression", in conforming SQL language, <XQuery
pattern> and <XQuery option flag> shall be <value specification>s.

Subclause 6.30, "<string value function>":

<XQuery replacement string> ::=
<character value expression>

...

Conformance Rules

Without Feature F847, "Nonconstant regular expressions", in conforming SQL language, <XQuery
pattern>, <XQuery option flag>, and <XQuery replacement string> shall be <value
specification>s.

Subclause 8.7, "<regex like predicate>":

<regex like predicate> ::=
<row value predicand> <regex like predicate part 2>

<regex like predicate part 2> ::=
[ NOT ] LIKE_REGEX <XQuery pattern> [ FLAG <XQuery option flag> ]

...
```

```
Conformance Rules

Without Feature F847, "Nonconstant regular expressions", in conforming SQL language, <XQuery
pattern> and <XQuery option flag> shall be <value specification>s.
```

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

Transact-SQL does not support this feature.

2.1.2.116 F851, <order by clause> in subqueries

V0152:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<order by clause> ::=
ORDER BY <sort specification list>

...

Conformance Rules

Without Feature F851, "<order by clause> in subqueries", in conforming SQL language, a <query
expression> contained in another <query expression> shall not immediately contain an <order
by clause>.
```

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

Transact-SQL partially supports this feature. The **ORDER BY** clause is not valid in subqueries unless a **TOP** clause is also specified.

2.1.2.117 F852, Top-level <order by clause> in views

V0153:

The specification states the following:

```
Subclause 11.32, "<view definition>":

<view definition> ::=
CREATE [ RECURSIVE ] VIEW <table name> <view specification>
AS <query expression> [ WITH [ <levels clause> ] CHECK OPTION ]

...

Conformance Rules
```

```
Without Feature F852, "Top-level <order by clause> in views", in conforming SQL language, a
<query expression> immediately contained in a <view definition> shall not immediately contain
an <order by clause>.

Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<order by clause> ::=
ORDER BY <sort specification list>
```

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

Transact-SQL partially supports this feature. The **ORDER BY** clause is not valid in views unless a **TOP** clause is also specified.

See [ISO/IEC9075-1:2011] for the definition of "view".

2.1.2.118 F856, Nested <fetch first clause> in <query expression>

V0154:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query primary> ::=
<simple table>
| <left paren> <query expression body>
[ <order by clause> ] [ <fetch first clause> ] <right paren>

...

<fetch first clause> ::=
FETCH FIRST [ <unsigned integer> ] { ROW | ROWS } ONLY

...

Conformance Rules

Without Feature F856, "Nested <fetch first clause> in <query expression>", in conforming SQL
language, a <query primary> shall not immediately contain a <fetch first clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.119 F857, Top-level <fetch first clause> in <query expression>

V0155:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<fetch first clause> ::=
FETCH FIRST [ <unsigned integer> ] { ROW | ROWS } ONLY

...

Conformance Rules

Without Feature F857, "Top-level <fetch first clause> in <query expression>", in conforming
SQL language, a <query expression> shall not immediately contain a <fetch first clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.120 F858, <fetch first clause> in subqueries

V0156:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<fetch first clause> ::=
FETCH FIRST [ <unsigned integer> ] { ROW | ROWS } ONLY

...

Conformance Rules

Without Feature F858, "<fetch first clause> in subqueries", in conforming SQL language, a
<query expression> contained in another <query expression> shall not immediately contain a
<fetch first clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.121 F859, Top-level <fetch first clause> in views

V0157:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

...

<fetch first clause> ::=
FETCH FIRST [ <unsigned integer> ] { ROW | ROWS } ONLY

Subclause 11.32, "<view definition>":

Conformance Rules

Without Feature F859, "Top-level <fetch first clause> in views", in conforming SQL language,
a <query expression> immediately contained in a <view definition> shall not immediately
contain a <fetch first clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.122 F866, FETCH FIRST clause: PERCENT option

V0158:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<result offset clause> ::=
OFFSET <offset row count> { ROW | ROWS }

...

<offset row count> ::=
<simple value specification>

...

Conformance Rules:

Without Feature F865, "dynamic <offset row count> in <result offset clause>", in conforming
SQL language, a <result offset clause> shall not contain a <offset row count> that is not an
<unsigned integer>.
```

Microsoft SQL Server 2008 R2varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.123 F867, FETCH FIRST clause: WITH TIES option

V0159:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<fetch first clause> ::=
FETCH { FIRST | NEXT } [ <fetch first quantity> ] { ROW | ROWS } { ONLY | WITH TIES }

...

Conformance Rules:

Without Feature F867, "FETCH FIRST clause: WITH TIES option", <fetch first clause> shall not
contain WITH TIES.
```

Microsoft SQL Server 2008 R2varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.124 S023, Basic structured types

V0160:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<schema-resolved user-defined type name> ::=
<user-defined type name>

<user-defined type name> ::=
[ <schema name> <period> ] <qualified identifier>

<attribute name> ::=
<identifier>

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<attribute name>.

Subclause 6.1, "<data type>":
```

```
<path-resolved user-defined type name> ::=
<user-defined type name>

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<path-resolved user-defined type name> that identifies a structured type.

Subclause 6.17, "<method invocation>":

<method invocation> ::=
<direct invocation>
| <generalized invocation>

<direct invocation> ::=
<value expression primary> <period> <method name> [ <SQL argument list> ]

<generalized invocation> ::=
<left paren> <value expression primary> AS <data type> <right paren>
<period> <method name> [ <SQL argument list> ]

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<method invocation>.

Subclause 6.19, "<new specification>":

<new specification> ::=
NEW <path-resolved user-defined type name> <SQL argument list>

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<new specification>.

Subclause 10.4, "<routine invocation>":

<generalized expression> ::=
<value expression> AS <path-resolved user-defined type name>

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<generalized expression>.

Subclause 11.51, "<user-defined type definition>":

<member list> ::=
<left paren> <member> [ { <comma> <member> }... ] <right paren>
```

```
...

<method specification list> ::=
<method specification> [ { <comma> <method specification> }... ]

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<member list>.

...

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<method specification list>.

Subclause 11.52, "<attribute definition>":

<attribute definition> ::=
<attribute name> <data type>
[ <attribute default> ]
[ <collate clause> ]

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain an
<attribute definition>.

Subclause 11.60, "<SQL-invoked routine>":

<method specification designator> ::=
SPECIFIC METHOD <specific method name>
| [ INSTANCE | STATIC | CONSTRUCTOR ]
METHOD <method name> <SQL parameter declaration list>
[ <returns clause> ]
FOR <schema-resolved user-defined type name>

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<method specification designator>.

Subclause 12.3, "<privileges>":

<privileges> ::=
<object privileges> ON <object name>

<object name> ::=
[ TABLE ] <table name>
| DOMAIN <domain name>
| COLLATION <collation name>
| CHARACTER SET <character set name>
| TRANSLATION <transliteration name>
| TYPE <schema-resolved user-defined type name>
```

```
| SEQUENCE <sequence generator name>
| <specific routine designator>

<object privileges> ::=
ALL PRIVILEGES
| <action> [ { <comma> <action> }... ]

<action> ::=
SELECT
| SELECT <left paren> <privilege column list> <right paren>
| SELECT <left paren> <privilege method list> <right paren>
| DELETE
| INSERT [ <left paren> <privilege column list> <right paren> ]
| UPDATE [ <left paren> <privilege column list> <right paren> ]
| REFERENCES [ <left paren> <privilege column list> <right paren> ]
| USAGE
| TRIGGER
| UNDER
| EXECUTE

...

Conformance Rules

Without Feature S023, "Basic structured types", conforming SQL language shall not contain a
<privileges> that contains an <action> that contains UNDER and that contains an <object name>
that contains a <schema-resolved user-defined type name> that identifies a structured type.
```

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

Transact-SQL does not support this feature. The **CREATE TYPE** command in Transact-SQL can create only scalar types.

2.1.2.125 S024, Enhanced structured types

V0161:

The specification states the following:

```
Subclause 6.18, "<static method invocation>":

<static method invocation> ::=
<path-resolved user-defined type name> <double colon> <method name>
[ <SQL argument list> ]

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <static method invocation>.

Subclause 9.11, "Equality operations":

Conformance Rules
```

```
Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an equality operation shall not be ST-ordered.

Subclause 9.12, "Grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of a grouping operation shall not be ST-ordered.

Subclause 9.13, "Multiset element grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
element type of a multiset operand of a multiset element grouping operation shall not be ST-
ordered.

Subclause 9.14, "Ordering operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an ordering operation shall not be ST-ordered.

Subclause 10.6, "<specific routine designator>":

<specific routine designator> ::=
SPECIFIC <routine type> <specific name>
| <routine type> <member name> [ FOR <schema-resolved user-defined type name> ]

<routine type> ::=
ROUTINE
| FUNCTION
| PROCEDURE
| [ INSTANCE | STATIC | CONSTRUCTOR ] METHOD

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <specific routine designator> that contains a <routine type> that immediately contains
METHOD.

Subclause 11.51, "<user-defined type definition>":

<instantiable clause> ::=
INSTANTIABLE
| NOT INSTANTIABLE

...

<original method specification> ::=
<partial method specification> [ SELF AS RESULT ] [ SELF AS LOCATOR ]
[ <method characteristics> ]

...
```

```
<partial method specification> ::=
[ INSTANCE | STATIC | CONSTRUCTOR ]
METHOD <method name> <SQL parameter declaration list>
<returns clause>
[ SPECIFIC <specific method name> ]

...

<method characteristics> ::=
<method characteristic>...

<method characteristic> ::=
<language clause>
| <parameter style clause>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
an <instantiable clause> that contains NOT INSTANTIABLE.

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
an <original method specification> that immediately contains SELF AS RESULT.

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <method characteristics> that contains a <parameter style> that contains GENERAL.

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
an <original method specification> that contains an <SQL-data access indication> that
immediately contains NO SQL.

...

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <partial method specification> that contains INSTANCE or STATIC.

Subclause 11.52, "<attribute definition>":

<attribute default> ::=
<default clause>

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
an <attribute default>.

Subclause 11.53, "<alter type statement>":

<alter type statement> ::=
ALTER TYPE <schema-resolved user-defined type name> <alter type action>

...
```

```
Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
an <alter type statement>.

Subclause 11.60, "<SQL-invoked routine>":

<SQL parameter declaration> ::=
[ <parameter mode> ] [ <SQL parameter name> ] <parameter type> [ RESULT ]

...

<parameter style> ::=
SQL
| GENERAL

...

<SQL-data access indication> ::=
NO SQL
| CONTAINS SQL
| READS SQL DATA
| MODIFIES SQL DATA

...

Conformance Rules

Without Feature S024, "Enhanced structured types", an <SQL parameter declaration> shall not
contain RESULT.

Subclause 11.62, "<drop routine statement>":

<drop routine statement> ::=
DROP <specific routine designator> <drop behavior>

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <drop routine statement> that contains a <specific routine designator> that identifies a
method.

Subclause 12.2, "<grant privilege statement>":

<grant privilege statement> ::=
GRANT <privileges> TO <grantee> [ { <comma> <grantee> }... ]
[ WITH HIERARCHY OPTION ]
[ WITH GRANT OPTION ]
[ GRANTED BY <grantor> ]

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <specific routine designator> contained in a <grant privilege statement> that identifies a
method.
```

```
Subclause 12.3, "<privileges>":

<privileges> ::=
<object privileges> ON <object name>

<object name> ::=
[ TABLE ] <table name>
| DOMAIN <domain name>
| COLLATION <collation name>
| CHARACTER SET <character set name>
| TRANSLATION <transliteration name>
| TYPE <schema-resolved user-defined type name>
| SEQUENCE <sequence generator name>
| <specific routine designator>

<object privileges> ::=
ALL PRIVILEGES
| <action> [ { <comma> <action> }... ]

<action> ::=
SELECT
| SELECT <left paren> <privilege column list> <right paren>
| SELECT <left paren> <privilege method list> <right paren>
| DELETE
| INSERT [ <left paren> <privilege column list> <right paren> ]
| UPDATE [ <left paren> <privilege column list> <right paren> ]
| REFERENCES [ <left paren> <privilege column list> <right paren> ]
| USAGE
| TRIGGER
| UNDER
| EXECUTE

<privilege method list> ::=
<specific routine designator> [ { <comma> <specific routine designator> }... ]

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <privileges> that contains an <action> that contains USAGE and that contains an <object
name> that contains a <schema-resolved user-defined type name> that identifies a structured
type.

...

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <privilege method list>.

Subclause 14.11, "<insert statement>":

<insert column list> ::=
<column name list>

...

Conformance Rules
```

```
Without Feature S024, "Enhanced structured types", in conforming SQL language, for each
column C identified in the explicit or implicit <insert column list>, if the declared type of
C is a structured type TY, then the declared type of the corresponding column of the <query
expression> or <contextually typed table value constructor> shall be TY.

Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

<with clause> ::=
WITH [ RECURSIVE ] <with list>

Subclause 7.3, "<table value constructor>":

<contextually typed table value constructor> ::=
VALUES <contextually typed row value expression list>

Subclause 7.1, "<row value constructor>":

<contextually typed row value constructor element list> ::=
<contextually typed row value constructor element>
[ { <comma> <contextually typed row value constructor element> }... ]

Subclause 14.12, "<merge statement>":

<merge statement> ::=
MERGE INTO <target table> [ [ AS ] <merge correlation name> ]
USING <table reference>
ON <search condition> <merge operation specification>

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <merge statement> that does not satisfy the condition: for each column C identified in the
explicit or implicit <insert column list>, if the declared type of C is a structured type TY,
then the declared type of the corresponding column of the <query expression> or <contextually
typed table value constructor> is TY.

Subclause 14.15, "<set clause list>":

<set clause> ::=
<multiple column assignment>
| <set target> <equals operator> <update source>

...

<assigned row> ::=
<contextually typed row value expression>

<update target> ::=
<object column>
| <object column>
<left bracket or trigraph> <simple value specification> <right bracket or trigraph>

<mutated set clause> ::=
```

```
<mutated target> <period> <method name>

...

<update source> ::=
<value expression>
| <contextually typed value specification>

...

Conformance Rules

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <set clause> in which the declared type of the <update target> in the <set clause> is a
structured type TY and the declared type of the <update source> or corresponding field of the
<assigned row> contained in the <set clause> is not TY.

Without Feature S024, "Enhanced structured types", conforming SQL language shall not contain
a <set clause> that contains a <mutated set clause> and in which the declared type of the
last <method name> identifies a structured type TY, and the declared type of the <update
source> contained in the <set clause> is not TY.

Subclause 5.4, "Names and identifiers":

<method name> ::=
<identifier>
```

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

Transact-SQL does not support this feature.

2.1.2.126 S025, Final structured types

V0162:

The specification states the following:

```
Subclause 11.51, "<user-defined type definition>":

<user-defined type definition> ::=
CREATE TYPE <user-defined type body>

<user-defined type body> ::=
<schema-resolved user-defined type name>
[ <subtype clause> ]
[ AS <representation> ]
[ <user-defined type option list> ]
[ <method specification list> ]

<user-defined type option list> ::=
<user-defined type option> [ <user-defined type option>... ]

<user-defined type option> ::=
<instantiable clause>
| <finality>
| <reference type specification>
| <cast to ref>
| <cast to type>
```

```
| <cast to distinct>
| <cast to source>

...

<finality> ::=
FINAL
| NOT FINAL

...

Conformance Rules

Without Feature S025, "Final structured types", in conforming SQL language, a <user-defined
type definition> that defines a structured type shall contain a <finality> that is NOT FINAL.
```

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

Transact-SQL does not support this feature.

2.1.2.127 S026, Self-referencing structured types

V0163:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<reference type> ::=
REF <left paren> <referenced type> <right paren> [ <scope clause> ]

...

<referenced type> ::=
<path-resolved user-defined type name>

Subclause 5.4, "Names and identifiers":

<schema-resolved user-defined type name> ::=
<user-defined type name>

<user-defined type name> ::=
[ <schema name> <period> ] <qualified identifier>

Subclause 11.51, "<user-defined type definition>":

<user-defined type definition> ::=
CREATE TYPE <user-defined type body>
```

```
Subclause 11.52, "<attribute definition>":

<attribute definition> ::=
<attribute name> <data type>
[ <attribute default> ]
[ <collate clause> ]

<attribute default> ::=
<default clause>

...

Conformance Rules

Without Feature S026, "Self-referencing structured types", conforming SQL language shall not
contain a <data type> simply contained in an <attribute definition> that is a <reference
type> whose <referenced type> is equivalent to the <schema-resolved user-defined type name>
simply contained in the <user-defined type definition> that contains <attribute definition>.
```

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

Transact-SQL does not support this feature.

2.1.2.128 S027, Create method by specific method name

V0164:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<method specification designator> ::=
SPECIFIC METHOD <specific method name>
| [ INSTANCE | STATIC | CONSTRUCTOR ]
METHOD <method name> <SQL parameter declaration list>
[ <returns clause> ]
FOR <schema-resolved user-defined type name>

...

Conformance Rules

Without Feature S027, "Create method by specific method name", conforming SQL language shall
not contain a <method specification designator> that contains SPECIFIC METHOD.
```

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

Transact-SQL does not support this feature.

2.1.2.129 S028, Permutable UDT options list

V0165:

The specification states the following:

```
Subclause 11.51, "<user-defined type definition>":

<user-defined type definition> ::=
CREATE TYPE <user-defined type body>

<user-defined type body> ::=
<schema-resolved user-defined type name>
[ <subtype clause> ]
[ AS <representation> ]
[ <user-defined type option list> ]
[ <method specification list> ]

<user-defined type option list> ::=
<user-defined type option> [ <user-defined type option>... ]

<user-defined type option> ::=
<instantiable clause>
| <finality>
| <reference type specification>
| <cast to ref>
| <cast to type>
| <cast to distinct>
| <cast to source>

...

Conformance Rules

Without Feature S028, "Permutable UDT options list", conforming SQL language shall not
contain a <user-defined type option list> in which <instantiable clause>, if specified,
<finality>, <reference type specification>, if specified, <cast to ref>, if specified, <cast
to type>, if specified, <cast to distinct>, if specified, and <cast to source>, if specified,
do not appear in that sequence.
```

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

Transact-SQL does not support this feature.

2.1.2.130 S041, Basic reference types

V0166:

The specification states the following:

```
Subclause 6.1, "<data type>":

<reference type> ::=
REF <left paren> <referenced type> <right paren> [ <scope clause> ]

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<reference type>.

Subclause 6.20, "<attribute or method reference>":
```

```
<attribute or method reference> ::=
<value expression primary> <dereference operator> <qualified identifier>
[ <SQL argument list> ]

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain an
<attribute or method reference>.

Subclause 6.21, "<dereference operation>":

<dereference operation> ::=
<reference value expression> <dereference operator> <attribute name>

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<dereference operation>.

Subclause 6.26, "<value expression>":

<reference value expression> ::=
<value expression primary>

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<reference value expression>.

Subclause 21.3, "<embedded SQL Ada program>":

<Ada REF variable> ::=
SQL TYPE IS <reference type>

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain an
<Ada REF variable>.

Subclause 21.4, "<embedded SQL C program>":

<C REF variable> ::=
SQL TYPE IS <reference type> <C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [ <C initial value> ] }... ]

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a <C
REF variable>.
```

```
Subclause 21.5, "<embedded SQL COBOL program>":

<COBOL REF variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS <reference type>

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<COBOL REF variable>.

Subclause 21.6, "<embedded SQL Fortran program>":

<Fortran REF variable> ::=
SQL TYPE IS <reference type>

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<Fortran REF variable>.

Subclause 21.7, "<embedded SQL MUMPS program>":

<MUMPS REF variable> ::=
SQL TYPE IS <reference type>

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<MUMPS REF variable>.

Subclause 21.8, "<embedded SQL Pascal program>":

<Pascal REF variable> ::=
SQL TYPE IS <reference type>

...

Conformance Rules

Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<Pascal REF variable>.

Subclause 21.9, "<embedded SQL PL/I program>":

<PL/I REF variable> ::=
SQL TYPE IS <reference type>

...

Conformance Rules
```

```
Without Feature S041, "Basic reference types", conforming SQL language shall not contain a
<PL/I REF variable>.
```

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

Transact-SQL does not support this feature.

2.1.2.131 S043, Enhanced reference types

V0167:

The specification states the following:

```
Subclause 11.4, "<column definition>":

<column definition> ::=
<column name> [ <data type or domain name> ]
[ <default clause> | <identity column specification> | <generation clause> ]
[ <column constraint definition>... ]
[ <collate clause> ]

Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<scope clause> ::=
SCOPE <table name>

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain a
<scope clause> that is not simply contained in a <data type> that is simply contained in a
<column definition>.

Subclause 6.13, "<cast specification>":

<cast operand> ::=
<value expression>
| <implicitly typed value specification>

<cast target> ::=
<domain name>
| <data type>

...

Conformance Rules
```

```
Without Feature S043, "Enhanced reference types", in conforming SQL language, if the declared
type of <cast operand> is a reference type, then <cast target> shall contain a <data type>
that is a reference type.

Subclause 6.22, "<method reference>":

<method reference> ::=
<value expression primary> <dereference operator> <method name> <SQL argument list>

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain a
<method reference>.

Subclause 6.23, "<reference resolution>":

<reference resolution> ::=
DEREF <left paren> <reference value expression> <right paren>

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain a
<reference resolution>.

Subclause 11.3, "<table definition>":

<reference generation> ::=
SYSTEM GENERATED
| USER GENERATED
| DERIVED

...

<column option list> ::=
[ <scope clause> ] [ <default clause> ] [ <column constraint definition>... ]

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain a
<column option list> that contains a <scope clause>.

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain
<reference generation> that does not contain SYSTEM GENERATED.

Subclause 11.17, "<add column scope clause>":

<add column scope clause> ::=
ADD <scope clause>

...

Conformance Rules
```

```
Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain
an <add column scope clause>.

Subclause 11.18, "<drop column scope clause>":

<drop column scope clause> ::=
DROP SCOPE <drop behavior>

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain a
<drop column scope clause>.

Subclause 11.32, "<view definition>":

<referenceable view specification> ::=
OF <path-resolved user-defined type name> [ <subview clause> ]
[ <view element list> ]

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain a
<referenceable view specification>.

Subclause 11.51, "<user-defined type definition>":

<reference type specification> ::=
<user-defined representation>
| <derived representation>
| <system-generated representation>

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain a
<reference type specification>.

Subclause 14.11, "<insert statement>":

<override clause> ::=
OVERRIDING USER VALUE
| OVERRIDING SYSTEM VALUE

...

Conformance Rules

Without Feature S043, "Enhanced reference types", conforming SQL language shall not contain
an <override clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.132 S051, Create table of type

V0168:

The specification states the following:

```
Subclause 11.3, "<table definition>":

<typed table clause> ::=
OF <path-resolved user-defined type name> [ <subtable clause> ]
[ <typed table element list> ]

...

Conformance Rules

Without Feature S051, "Create table of type", conforming SQL language shall not contain "OF
<path-resolved user-defined type name>".

Subclause 11.32, "<view definition>":

<referenceable view specification> ::=
OF <path-resolved user-defined type name> [ <subview clause> ]
[ <view element list> ]

Subclause 6.1, "<data type>":

<path-resolved user-defined type name> ::=
<user-defined type name>

Subclause 5.4, "Names and identifiers":

<user-defined type name> ::=
[ <schema name> <period> ] <qualified identifier>
```

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

Transact-SQL does not support this feature. For more details, see feature S023 (section 2.1.2.124).

2.1.2.133 S071, SQL paths in function and type name resolution

V0169:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
```

```
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

Conformance Rules

Without Feature S071, "SQL paths in function and type name resolution", conforming SQL
language shall not contain a <general value specification> that contains CURRENT_PATH.

Subclause 10.3, "<path specification>":

<path specification> ::=
PATH <schema name list>

...

Conformance Rules

Without Feature S071, "SQL paths in function and type name resolution", conforming SQL
language shall not contain a <path specification>.

Subclause 11.1, "<schema definition>":

<schema path specification> ::=
<path specification>

...

Conformance Rules

Without Feature S071, "SQL paths in function and type name resolution", conforming SQL
language shall not contain a <schema path specification>.

Subclause 11.5, "<default clause>":

<default option> ::=
<literal>
| <datetime value function>
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
| CURRENT_PATH
| <implicitly typed value specification>

...

Conformance Rules
```

```
Without Feature S071, "SQL paths in function and type name resolution", conforming SQL
language shall not contain a <default option> that contains CURRENT_PATH.

Subclause 13.1, "<SQL-client module definition>":

<module path specification> ::=
<path specification>

...

Conformance Rules

Without Feature S071, "SQL paths in function and type name resolution", conforming SQL
language shall not contain a <module path specification>.

Subclause 19.8, "<set path statement>":

<set path statement> ::=
SET <SQL-path characteristic>

<SQL-path characteristic> ::=
PATH <value specification>

...

Conformance Rules

Without Feature S071, "SQL paths in function and type name resolution", Conforming SQL
language shall not contain a <set path statement>.

Subclause 21.1, "<embedded SQL host program>":

<embedded path specification> ::=
<path specification>

...

Conformance Rules

Without Feature S071, "SQL paths in function and type name resolution", conforming SQL
language shall not contain an <embedded path specification>.
```

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

Transact-SQL does not support this feature.

2.1.2.134 S081, Subtables

V0170:

The specification states the following:

```
Subclause 11.3, "<table definition>":

<subtable clause> ::=
UNDER <supertable clause>
```

```
<supertable clause> ::=
<supertable name>

<supertable name> ::=
<table name>

...

Conformance Rules

Without Feature S081, "Subtables", conforming SQL language shall not contain a <subtable
clause>.

Subclause 12.2, "<grant privilege statement>":

<grant privilege statement> ::=
GRANT <privileges> TO <grantee> [ { <comma> <grantee> }... ]
[ WITH HIERARCHY OPTION ]
[ WITH GRANT OPTION ]
[ GRANTED BY <grantor> ]

...

Conformance Rules

Without Feature S081, "Subtables", conforming SQL language shall not contain a <grant
privilege statement> that contains WITH HIERARCHY OPTION.

Subclause 12.3, "<privileges>":

<privileges> ::=
<object privileges> ON <object name>

<object name> ::=
[ TABLE ] <table name>
| DOMAIN <domain name>
| COLLATION <collation name>
| CHARACTER SET <character set name>
| TRANSLATION <transliteration name>
| TYPE <schema-resolved user-defined type name>
| SEQUENCE <sequence generator name>
| <specific routine designator>

<object privileges> ::=
ALL PRIVILEGES
| <action> [ { <comma> <action> }... ]

<action> ::=
SELECT
| SELECT <left paren> <privilege column list> <right paren>
| SELECT <left paren> <privilege method list> <right paren>
| DELETE
| INSERT [ <left paren> <privilege column list> <right paren> ]
| UPDATE [ <left paren> <privilege column list> <right paren> ]
| REFERENCES [ <left paren> <privilege column list> <right paren> ]
| USAGE
| TRIGGER
| UNDER
| EXECUTE
```

```
...

Conformance Rules

Without Feature S081, "Subtables", conforming SQL language shall not contain a <privileges>
that contains an <action> that contains UNDER and that contains an <object name> that
contains a <table name>.

Subclause 12.7, "<revoke statement>":

<revoke privilege statement> ::=
REVOKE [ <revoke option extension> ] <privileges>
FROM <grantee> [ { <comma> <grantee> }... ]
[ GRANTED BY <grantor> ]
<drop behavior>

<revoke option extension> ::=
GRANT OPTION FOR
| HIERARCHY OPTION FOR

...

Conformance Rules

Without Feature S081, "Subtables", conforming SQL language shall not contain a <revoke option
extension> that contains HIERARCHY OPTION FOR.
```

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

Transact-SQL does not support this feature.

2.1.2.135 S091, Basic array support

V0171:

The specification states the following:

```
Subclause 6.1, "<data type>":

<array type> ::=
<data type> ARRAY
[ <left bracket or trigraph> <maximum cardinality> <right bracket or trigraph> ]

...

Conformance Rules

Without Feature S091, "Basic array support", conforming SQL language shall not contain an
<array type>.

Subclause 6.5, "<contextually typed value specification>":

<contextually typed value specification> ::=
<implicitly typed value specification>
| <default specification>
```

```
<implicitly typed value specification> ::=
<null specification>
| <empty specification>

<null specification> ::=
NULL

<empty specification> ::=
ARRAY <left bracket or trigraph> <right bracket or trigraph>
| MULTISET <left bracket or trigraph> <right bracket or trigraph>

...

Conformance Rules

Without Feature S091, "Basic array support", conforming SQL language shall not contain an
<empty specification> that simply contains ARRAY.

Subclause 6.24, "<array element reference>":

<array element reference> ::=
<array value expression>
<left bracket or trigraph> <numeric value expression> <right bracket or trigraph>

...

Conformance Rules

Without Feature S091, "Basic array support", conforming SQL language shall not contain an
<array element reference>.

Subclause 6.28, "<numeric value function>":

<cardinality expression> ::=
CARDINALITY <left paren> <collection value expression> <right paren>

...

Conformance Rules

Without Feature S091, "Basic array support", or Feature S271, "Basic multiset support",
conforming SQL language shall not contain a <cardinality expression>.

Subclause 6.36, "<array value expression>":

<array value expression> ::=
<array concatenation>
| <array primary>

<array concatenation> ::=
<array value expression 1> <concatenation operator> <array primary>

<array value expression 1> ::=
<array value expression>

<array primary> ::=
<value expression primary>

...
```

```
Conformance Rules

Without Feature S091, "Basic array support", conforming SQL language shall not contain an
<array value expression>.

Subclause 6.38, "<array value constructor>":

<array value constructor> ::=
<array value constructor by enumeration>
| <array value constructor by query>

<array value constructor by enumeration> ::=
ARRAY <left bracket or trigraph> <array element list> <right bracket or trigraph>

...

Conformance Rules

Without Feature S091, "Basic array support", conforming SQL language shall not contain an
<array value constructor by enumeration>.

Subclause 7.6, "<table reference>":

<collection derived table> ::=
UNNEST <left paren> <collection value expression>
[ { <comma> <collection value expression> }... ] <right paren>
[ WITH ORDINALITY ]

...

Conformance Rules

Without Feature S091, "Basic array support", or Feature S271, "Basic multiset support",
conforming SQL language shall not contain a <collection derived table>.

Subclause 14.15, "<set clause list>":

<update target> ::=
<object column>
| <object column>
<left bracket or trigraph> <simple value specification> <right bracket or trigraph>

...

Conformance Rules

Without Feature S091, "Basic array support", conforming SQL language shall not contain an
<update target> that immediately contains a <simple value specification>.

Subclause 6.4, "<value specification> and <target specification>":

<simple value specification> ::=
<literal>
| <host parameter name>
| <SQL parameter reference>
| <embedded variable name>
```

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

Transact-SQL does not support this feature. However, table-valued parameters provide equivalent functionality.

See [ISO/IEC9075-1:2011] for the definition of "table".

2.1.2.136 S091-01, Arrays of built-in data types

V0172:

The specification states the following:

```
Subclause 6.38, "<array value constructor>":

<array value constructor by enumeration> ::=
ARRAY <left bracket or trigraph> <array element list> <right bracket or trigraph>

<array element list> ::=
<array element> [ { <comma> <array element> }... ]

<array element> ::=
<value expression>

Subclause 6.26, "<value expression>":

<value expression> ::=
<common value expression>
| <boolean value expression>
| <row value expression>

<common value expression> ::=
|<numeric value expression>
|<string value expression>
|<datetime value expression>
|<interval value expression>
|<user-defined type value expression>
|<reference value expression>
|<collection value expression>
```

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

Transact-SQL does not support this feature.

2.1.2.137 S091-02, Arrays of distinct types

V0173:

The specification states the following:

```
Subclause 6.38, "<array value constructor>":

<array value constructor by enumeration> ::=
ARRAY <left bracket or trigraph> <array element list> <right bracket or trigraph>

<array element list> ::=
<array element> [ { <comma> <array element> }... ]
```

```
<array element> ::=
<value expression>

Subclause 6.26, "<value expression>":

<value expression> ::=
<common value expression>
| <boolean value expression>
| <row value expression>

<common value expression> ::=
|<numeric value expression>
|<string value expression>
|<datetime value expression>
|<interval value expression>
|<user-defined type value expression>
|<reference value expression>
|<collection value expression>
```

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

Transact-SQL does not support this feature.

2.1.2.138 S091-03, Array expressions

V0174:

The specification states the following:

```
Subclause 6.36, "<array value expression>":

<array value expression> ::=
<array concatenation>
| <array primary>

<array concatenation> ::=
<array value expression 1> <concatenation operator> <array primary>

<array value expression 1> ::=
<array value expression>

<array primary> ::=
<value expression primary>

...

Conformance Rules

Without Feature S091, "Basic array support", conforming SQL language shall not contain an
<array value expression>.
```

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

Transact-SQL does not support this feature.

2.1.2.139 S092, Arrays of user-defined types

V0175:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<path-resolved user-defined type name> ::=
<user-defined type name>

...

<array type> ::=
<data type> ARRAY
[ <left bracket or trigraph> <maximum cardinality> <right bracket or trigraph> ]

...

Conformance Rules

Without Feature S092, "Arrays of user-defined types", conforming SQL language shall not
contain an <array type> that is based on a <data type> that contains a <path-resolved user-
defined type name>.
```

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

Transact-SQL does not support this feature.

2.1.2.140 S094, Arrays of reference types

V0176:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<reference type> ::=
```

```
REF <left paren> <referenced type> <right paren> [ <scope clause> ]

...

<array type> ::=
<data type> ARRAY
[ <left bracket or trigraph> <maximum cardinality> <right bracket or trigraph> ]

...

Conformance Rules

Without Feature S094, "Arrays of reference types", conforming SQL language shall not contain
an <array type> that is based on a <data type> that contains a <reference type>.
```

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

Transact-SQL does not support this feature.

2.1.2.141 S095, Array constructors by query

V0177:

The specification states the following:

```
Subclause 6.38, "<array value constructor>":

<array value constructor> ::=
<array value constructor by enumeration>
| <array value constructor by query>

...

<array value constructor by query> ::=
ARRAY <left paren> <query expression> <right paren>

...

Conformance Rules

Without Feature S095, "Array constructors by query", conforming SQL language shall not
contain an <array value constructor by query>.
```

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

Transact-SQL does not support this feature.

2.1.2.142 S096, Optional array bounds

V0178:

The specification states the following:

```
Subclause 6.1, "<data type>":
```

```
<array type> ::=
<data type> ARRAY
[ <left bracket or trigraph> <maximum cardinality> <right bracket or trigraph> ]

<maximum cardinality> ::=
<unsigned integer>

...

Conformance Rules

Without Feature S096, "Optional array bounds", conforming SQL language shall not contain an
<array type> that does not immediately contain <maximum cardinality>.
```

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

Transact-SQL does not support this feature.

2.1.2.143 S097, Array element assignment

V0179:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<target array element specification> ::=
<target array reference>
<left bracket or trigraph> <simple value specification> <right bracket or trigraph>

<target array reference> ::=
<SQL parameter reference>
| <column reference>

...

Conformance Rules

Without Feature S097, "Array element assignment", conforming SQL language shall not contain a
<target array element specification>.
```

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

Transact-SQL does not support this feature.

2.1.2.144 S098, ARRAY_AGG

V0180:

The specification states the following:

```
Subclause 10.9, "<aggregate function>":

<array aggregate function> ::=
```

```
ARRAY_AGG
<left paren> <value expression> [ ORDER BY <sort specification list> ] <right paren>

...

Conformance Rules

Without Feature S098, "ARRAY_AGG", conforming SQL language shall not contain an <array
aggregate function>.
```

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

Transact-SQL does not support this feature.

2.1.2.145 S111, ONLY in query expressions

V0181:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<table reference> ::=
<table factor>
| <joined table>

<table factor> ::=
<table primary> [ <sample clause> ]

...

<table primary> ::=
<table or query name> [ [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ] ]
| <derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <lateral derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <collection derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <table function derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <only spec> [ [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ] ]
| <parenthesized joined table>

<only spec> ::=
ONLY <left paren> <table or query name> <right paren>

...

Conformance Rules

Without Feature S111, "ONLY in query expressions", conforming SQL language shall not contain
a <table reference> that contains an <only spec>.
```

```
Subclause 14.8, "<delete statement: positioned>":

<target table> ::=
<table name>
| ONLY <left paren> <table name> <right paren>

...

Conformance Rules

Without Feature S111, "ONLY in query expressions", conforming SQL language shall not contain
a <target table> that contains ONLY.
```

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

Transact-SQL does not support this feature.

2.1.2.146 S151, Type predicate

V0182:

The specification states the following:

```
Subclause 8.19, "<type predicate>":

<type predicate> ::=
<row value predicand> <type predicate part 2>

<type predicate part 2> ::=
IS [ NOT ] OF <left paren> <type list> <right paren>

...

Conformance Rules

Without Feature S151, "Type predicate", conforming SQL language shall not contain a <type
predicate>.
```

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

Transact-SQL does not support this feature.

2.1.2.147 S161, Subtype treatment

V0183:

The specification states the following:

```
Subclause 6.16, "<subtype treatment>":

<subtype treatment> ::=
TREAT <left paren> <subtype operand> AS <target subtype> <right paren>

<subtype operand> ::=
```

```
<value expression>

<target subtype> ::=
<path-resolved user-defined type name>
| <reference type>

...

Conformance Rules

Without Feature S161, "Subtype treatment", conforming SQL Language shall not contain a
<subtype treatment>.
```

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

Transact-SQL does not support this feature.

2.1.2.148 S162, Subtype treatment for references

V0184:

The specification states the following:

```
Subclause 6.16, "<subtype treatment>":

<subtype treatment> ::=
TREAT <left paren> <subtype operand> AS <target subtype> <right paren>

<subtype operand> ::=
<value expression>

<target subtype> ::=
<path-resolved user-defined type name>
| <reference type>

...

Conformance Rules

Without Feature S162, "Subtype treatment for references", conforming SQL language shall not
contain a <target subtype> that contains a <reference type>.
```

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

Transact-SQL does not support this feature.

2.1.2.149 S201, SQL routines on arrays

V0185:

The specification states the following:

```
Subclause 10.4, "<routine invocation>":

Conformance Rules
```

```
Without Feature S201, "SQL routines on arrays", conforming SQL language shall not contain an
<SQL argument> whose declared type is an array type.

Subclause 11.60, "<SQL-invoked routine>":

Conformance Rules

Without Feature S201, "SQL routines on arrays", conforming SQL language shall not contain a
<parameter type> that is based on an array type.

Without Feature S201, "SQL routines on arrays", conforming SQL language shall not contain a
<returns data type> that is based on an array type.
```

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

Transact-SQL does not support this feature.

2.1.2.150 S201-01, Array parameters

V0186:

The specification states the following:

```
Subclause 10.4, "<routine invocation>":

<SQL argument> ::=
<value expression>
| <generalized expression>
| <target specification>

<generalized expression> ::=
<value expression> AS <path-resolved user-defined type name>

Subclause 11.60, "<SQL-invoked routine>":

<parameter type> ::=
<data type> [ <locator indication> ]

<locator indication> ::=
AS LOCATOR
```

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

Transact-SQL does not support this feature.

2.1.2.151 S201-02, Array as result type of functions

V0187:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<locator indication> ::=
AS LOCATOR
```

```
...

<returns clause> ::=
RETURNS <returns type>

<returns type> ::=
<returns data type> [ <result cast> ]
| <returns table type>

...

<returns data type> ::=
<data type> [ <locator indication> ]
```

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

Transact-SQL does not support this feature.

2.1.2.152 S202, SQL-invoked routines on multisets

V0188:

The specification states the following:

```
Subclause 10.4, "<routine invocation>":

<SQL argument> ::=
<value expression>
| <generalized expression>
| <target specification>

<generalized expression> ::=
<value expression> AS <path-resolved user-defined type name>

...

Conformance Rules

Without Feature S202, "SQL-invoked routines on multisets", conforming SQL language shall not
contain an <SQL argument> whose declared type is a multiset type.

Subclause 11.60, "<SQL-invoked routine>":

<parameter type> ::=
<data type> [ <locator indication> ]

...

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature S202, "SQL-invoked routines on multisets", conforming SQL language shall not
contain a <parameter type> that is based on a multiset type.
```

```
Without Feature S202, "SQL-invoked routines on multisets", conforming SQL language shall not
contain a <returns data type> that is based on a multiset type.
```

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

Transact-SQL does not support this feature.

2.1.2.153 S211, User-defined cast functions

V0189:

The specification states the following:

```
Subclause 11.63, "<user-defined cast definition>":

<user-defined cast definition> ::=
CREATE CAST <left paren> <source data type> AS <target data type> <right paren>
WITH <cast function>
[ AS ASSIGNMENT ]

...

Conformance Rules

Without Feature S211, "User-defined cast functions", conforming SQL language shall not
contain a <user-defined cast definition>.

Subclause 11.64, "<drop user-defined cast statement>":

<drop user-defined cast statement> ::=
DROP CAST <left paren> <source data type> AS <target data type> <right paren>
<drop behavior>

...

Conformance Rules

Without Feature S211, "User-defined cast functions", conforming SQL language shall not
contain a <drop user-defined cast statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.154 S231, Structured type locators

V0190:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<user-defined type name> ::=
[ <schema name> <period> ] <qualified identifier>
```

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<path-resolved user-defined type name> ::=
<user-defined type name>

Subclause 11.60, "<SQL-invoked routine>":

<parameter type> ::=
<data type> [ <locator indication> ]

<locator indication> ::=
AS LOCATOR

...

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<parameter type> that contains a <locator indication> and that simply contains a <data type>
that identifies a structured type.

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<returns data type> that contains a <locator indication> and that simply contains a <data
type> that identifies a structured type.

Subclause 13.3, "<externally-invoked procedure>":

<host parameter data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<host parameter data type> that simply contains a <data type> that specifies a structured
type and that contains a <locator indication>.

Subclause 21.3, "<embedded SQL Ada program>":

<Ada user-defined type locator variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS LOCATOR

...
```

```
Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<path-resolved user-defined type name> simply contained in an <Ada user-defined type locator
variable> that identifies a structured type.

Subclause 21.4, "<embedded SQL C program>":

<C user-defined type locator variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS LOCATOR
<C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [
<C initial value> ] }... ]

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<path-resolved user-defined type name> simply contained in a <C user-defined type locator
variable> that identifies a structured type.

Subclause 21.5, "<embedded SQL COBOL program>":

<COBOL user-defined type locator variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS <path-resolved user-defined type name> AS LOCATOR

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<path-resolved user-defined type name> simply contained in a <COBOL user-defined type locator
variable> that identifies a structured type.

Subclause 21.6, "<embedded SQL Fortran program>":

<Fortran user-defined type locator variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS LOCATOR

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<path-resolved user-defined type name> simply contained in a <Fortran user-defined type
locator variable> that identifies a structured type.

Subclause 21.7, "<embedded SQL MUMPS program>":

<MUMPS user-defined type locator variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS LOCATOR

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<path-resolved user-defined type name> simply contained in a <MUMPS user-defined type locator
variable> that identifies a structured type.
```

```
Subclause 21.8, "<embedded SQL Pascal program>":

<Pascal user-defined type locator variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS LOCATOR

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<path-resolved user-defined type name> simply contained in a <Pascal user-defined type
locator variable> that identifies a structured type.

Subclause 21.9, "<embedded SQL PL/I program>":

<PL/I user-defined type locator variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS LOCATOR

...

Conformance Rules

Without Feature S231, "Structured type locators", conforming SQL language shall not contain a
<path-resolved user-defined type name> simply contained in a <PL/I user-defined type locator
variable> that identifies a structured type.
```

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

Transact-SQL does not support this feature.

2.1.2.155 S232, Array locators

V0191:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<array type> ::=
<data type> ARRAY
[ <left bracket or trigraph> <maximum cardinality> <right bracket or trigraph> ]

Subclause 11.60, "<SQL-invoked routine>":

<parameter type> ::=
<data type> [ <locator indication> ]

<locator indication> ::=
```

```
AS LOCATOR

...

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain a
<parameter type> that contains a <locator indication> and that simply contains a <data type>
that identifies an array type.

Without Feature S232, "Array locators", conforming SQL language shall not contain a <returns
data type> that contains a <locator indication> and that simply contains a <data type> that
identifies an array type.

Subclause 13.3, "<externally-invoked procedure>":

<host parameter data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain a <host
parameter data type> that simply contains an <array type> and that contains a <locator
indication>.

Subclause 21.3, "<embedded SQL Ada program>":

<Ada array locator variable> ::=
SQL TYPE IS <array type> AS LOCATOR

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain an <Ada
array locator variable>.

Subclause 21.4, "<embedded SQL C program>":

<C array locator variable> ::=
SQL TYPE IS <array type> AS LOCATOR
<C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [
<C initial value> ] } ... ]

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain an <C array
locator variable>.

Subclause 21.5, "<embedded SQL COBOL program>":
```

```
<COBOL array locator variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS <array type> AS LOCATOR

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain a <COBOL
array locator variable>.

Subclause 21.6, "<embedded SQL Fortran program>":

<Fortran array locator variable> ::=
SQL TYPE IS <array type> AS LOCATOR

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain a <Fortran
array locator variable>.

Subclause 21.7, "<embedded SQL MUMPS program>":

<MUMPS array locator variable> ::=
SQL TYPE IS <array type> AS LOCATOR

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain a <MUMPS
array locator variable>.

Subclause 21.8, "<embedded SQL Pascal program>":

<Pascal array locator variable> ::=
SQL TYPE IS <array type> AS LOCATOR

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain a <Pascal
array locator variable>.

Subclause 21.9, "<embedded SQL PL/I program>":

<PL/I array locator variable> ::=
SQL TYPE IS <array type> AS LOCATOR

...

Conformance Rules

Without Feature S232, "Array locators", conforming SQL language shall not contain a <PL/I
array locator variable>.
```

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

Transact-SQL does not support this feature.

2.1.2.156 S233, Multiset locators

V0192:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<multiset type> ::=
<data type> MULTISET

Subclause 11.60, "<SQL-invoked routine>":

<parameter type> ::=
<data type> [ <locator indication> ]

<locator indication> ::=
AS LOCATOR

...

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a
<parameter type> that contains a <locator indication> and that simply contains a <data type>
that identifies a multiset type.

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a
<returns data type> that contains a <locator indication> and that simply contains a <data
type> that identifies a multiset type.

Subclause 13.3, "<externally-invoked procedure>":

<host parameter data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules
```

```
Without Feature S233, "Multiset locators", conforming SQL language shall not contain a <host
parameter data type> that simply contains a <multiset type> and that contains a <locator
indication>.

Subclause 21.3, "<embedded SQL Ada program>":

<Ada multiset locator variable> ::=
SQL TYPE IS <multiset type> AS LOCATOR

...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain an <Ada
multiset locator variable>.

Subclause 21.4, "<embedded SQL C program>":

<C multiset locator variable> ::=
SQL TYPE IS <multiset type> AS LOCATOR
<C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [
<C initial value> ] } ... ]

...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a <C
multiset locator variable>.

Subclause 21.5, "<embedded SQL COBOL program>":

<COBOL multiset locator variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS <multiset type> AS LOCATOR

...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a <COBOL
multiset locator variable>.

Subclause 21.6, "<embedded SQL Fortran program>":

<Fortran multiset locator variable> ::=
SQL TYPE IS <multiset type> AS LOCATOR

...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a
<Fortran multiset locator variable>.

Subclause 21.7, "<embedded SQL MUMPS program>":

<MUMPS multiset locator variable> ::=
SQL TYPE IS <multiset type> AS LOCATOR
```

```
...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a <MUMPS
multiset locator variable>.

Subclause 21.8, "<embedded SQL Pascal program>":

<Pascal multiset locator variable> ::=
SQL TYPE IS <multiset type> AS LOCATOR

...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a
<Pascal multiset locator variable>.

Subclause 21.9, "<embedded SQL PL/I program>":

<PL/I multiset locator variable> ::=
SQL TYPE IS <multiset type> AS LOCATOR

...

Conformance Rules

Without Feature S233, "Multiset locators", conforming SQL language shall not contain a <PL/I
multiset locator variable>.
```

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

Transact-SQL does not support this feature.

2.1.2.157 S241, Transform functions

V0193:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
```

```
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain
CURRENT_DEFAULT_TRANSFORM_GROUP.

Without Feature S241, "Transform functions", conforming SQL language shall not contain
CURRENT_TRANSFORM_GROUP_FOR_TYPE.

Subclause 11.60, "<SQL-invoked routine>":

<transform group specification> ::=
TRANSFORM GROUP { <single group specification> | <multiple group specification> }

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<transform group specification>.

Subclause 11.67, "<transform definition>":

<transform definition> ::=
CREATE { TRANSFORM | TRANSFORMS } FOR
<schema-resolved user-defined type name> <transform group>...

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<transform definition>.

Subclause 11.71, "<drop transform statement>":

<drop transform statement> ::=
DROP { TRANSFORM | TRANSFORMS } <transforms to be dropped>
FOR <schema-resolved user-defined type name> <drop behavior>

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<drop transform statement>.

Subclause 13.1, "<SQL-client module definition>":

<module transform group specification> ::=
<transform group specification>

...
```

```
Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<module transform group specification>.

Subclause 19.9, "<set transform group statement>":

<set transform group statement> ::=
SET <transform group characteristic>

<transform group characteristic> ::=
DEFAULT TRANSFORM GROUP <value specification>
| TRANSFORM GROUP FOR TYPE <path-resolved user-defined type name> <value specification>

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a <set
transform group statement>.

Subclause 21.1, "<embedded SQL host program>":

<embedded transform group specification> ::=
<transform group specification>

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<embedded transform group specification>.

Subclause 21.3, "<embedded SQL Ada program>":

<Ada user-defined type variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS <predefined type>

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain an
<Ada user-defined type variable>.

Subclause 21.4, "<embedded SQL C program>":

<C user-defined type variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS <predefined type>
<C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [
<C initial value> ] } ... ]

...

Conformance Rules
```

```
Without Feature S241, "Transform functions", conforming SQL language shall not contain a <C
user-defined type variable>.

Subclause 21.5, "<embedded SQL COBOL program>":

<COBOL user-defined type variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS <path-resolved user-defined type name>
AS <predefined type>

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<COBOL user-defined type variable>.

Subclause 21.6, "<embedded SQL Fortran program>":

<Fortran user-defined type variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS <predefined type>

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<Fortran user-defined type variable>.

Subclause 21.7, "<embedded SQL MUMPS program>":

<MUMPS user-defined type variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS <predefined type

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<MUMPS user-defined type variable>.

Subclause 21.8, "<embedded SQL Pascal program>":

<Pascal user-defined type variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS <predefined type>

...

Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<Pascal user-defined type variable>.

Subclause 21.9, "<embedded SQL PL/I program>":

<PL/I user-defined type variable> ::=
SQL TYPE IS <path-resolved user-defined type name> AS <predefined type>

...
```

```
Conformance Rules

Without Feature S241, "Transform functions", conforming SQL language shall not contain a
<PL/I user-defined type variable>.
```

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

Transact-SQL does not support this feature.

2.1.2.158 S242, Alter transform statement

V0194:

The specification states the following:

```
Subclause 11.68, "<alter transform statement>":

<alter transform statement> ::=
ALTER { TRANSFORM | TRANSFORMS }
FOR <schema-resolved user-defined type name> <alter group>...

<alter group> ::=
<group name> <left paren> <alter transform action list> <right paren>

<alter transform action list> ::=
<alter transform action> [ { <comma> <alter transform action> }... ]

<alter transform action> ::=
<add transform element list>
| <drop transform element list>

...

Conformance Rules

Without Feature S242, "Alter transform statement", conforming SQL language shall not contain
an <alter transform statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.159 S251, User-defined orderings

V0195:

The specification states the following:

```
Subclause 11.65, "<user-defined ordering definition>":

<user-defined ordering definition> ::=
CREATE ORDERING FOR <schema-resolved user-defined type name> <ordering form>

...

Conformance Rules
```

```
Without Feature S251, "User-defined orderings", conforming SQL shall not contain a <user-
defined ordering definition>.

NOTE — If MAP is specified, then the Conformance Rules of Subclause 9.11, "Equality
operations", apply. If ORDER FULL BY MAP is specified, then the Conformance Rules of
Subclause 9.14, "Ordering operations", also apply.

Subclause 9.11, "Equality operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an equality operation shall not be ST-ordered.

Without Feature T042, "Extended LOB data type support", in conforming SQL language, the
declared type of an operand of an equality operation shall not be LOB-ordered.

Without Feature S275, "Advanced multiset support", in conforming SQL language, the declared
type of an operand of an equality operation shall not be multiset-ordered.

NOTE — If the declared type of an operand OP of an equality operation is a multiset type,
then OP is a multiset operand of a multiset element grouping operation. The Conformance Rules
of Subclause 9.13, "Multiset element grouping operations", apply.

Subclause 9.13, "Multiset element grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
element type of a multiset operand of a multiset element grouping operation shall not be ST-
ordered.

Subclause 9.14, "Ordering operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an ordering operation shall not be ST-ordered.

Subclause 11.66, "<drop user-defined ordering statement>":

<drop user-defined ordering statement> ::=
DROP ORDERING FOR <schema-resolved user-defined type name> <drop behavior>

...

Conformance Rules

Without Feature S251, "User-defined orderings", conforming SQL language shall not contain a
<drop user-defined ordering statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.160 S261, Specific type method

V0196:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<string value function> ::=
<character value function>
| <binary value function>

<character value function> ::=
<character substring function>
| <regular expression substring function>
| <regex substring function>
| <fold>
| <transcoding>
| <character transliteration>
| <regex transliteration>
| <trim function>
| <character overlay function>
| <normalize function>
| <specific type method>

...

<specific type method> ::=
<user-defined type value expression> <period> SPECIFICTYPE
[ <left paren> <right paren> ]

...

Conformance Rules

Without Feature S261, "Specific type method", conforming SQL language shall not contain a
<specific type method>.
```

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

Transact-SQL does not support this feature.

2.1.2.161 S271, Basic multiset support

V0197:

The specification states the following:

```
Subclause 6.1, "<data type>":

<multiset type> ::=
<data type> MULTISET

...
```

```
Conformance Rules

Without Feature S271, "Basic multiset support", conforming SQL language shall not contain a
<multiset type>.

Subclause 6.5, "<contextually typed value specification>":

<empty specification> ::=
ARRAY <left bracket or trigraph> <right bracket or trigraph>
| MULTISET <left bracket or trigraph> <right bracket or trigraph>

...

Conformance Rules

Without Feature S271, "Basic multiset support", conforming SQL language shall not contain an
<empty specification> that simply contains MULTISET.

Subclause 6.25, "<multiset element reference>":

<multiset element reference> ::=
ELEMENT <left paren> <multiset value expression> <right paren>

...

Conformance Rules

Without Feature S271, "Basic multiset support", conforming SQL language shall not contain a
<multiset element reference>.

Subclause 6.28, "<numeric value function>":

<cardinality expression> ::=
CARDINALITY <left paren> <collection value expression> <right paren>

...

Conformance Rules

Without Feature S091, "Basic array support", or Feature S271, "Basic multiset support",
conforming SQL language shall not contain a <cardinality expression>.

Subclause 6.40, "<multiset value function>":

<multiset value function> ::=
<multiset set function>

<multiset set function> ::=
SET <left paren> <multiset value expression> <right paren>

...

Conformance Rules

Without Feature S271, "Basic multiset support", conforming SQL language shall not contain a
<multiset value function>.

NOTE — The Conformance Rules of Subclause 9.13, "Multiset element grouping operations", also
apply.
```

```
Subclause 9.13, "Multiset element grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
element type of a multiset operand of a multiset element grouping operation shall not be ST-
ordered.

Subclause 6.41, "<multiset value constructor>":

<multiset value constructor> ::=
<multiset value constructor by enumeration>
| <multiset value constructor by query>
| <table value constructor by query>

<multiset value constructor by enumeration> ::=
MULTISET <left bracket or trigraph> <multiset element list> <right bracket or trigraph>

...

<multiset value constructor by query> ::=
MULTISET <table subquery>

<table value constructor by query> ::=
TABLE <table subquery>

...

Conformance Rules

Without Feature S271, "Basic multiset support", conforming SQL language shall not contain a
<multiset value constructor>.

Subclause 7.6, "<table reference>":

<collection derived table> ::=
UNNEST <left paren> <collection value expression>
[ { <comma> <collection value expression> }... ] <right paren>
[ WITH ORDINALITY ]

...

Conformance Rules

Without Feature S091, "Basic array support", or Feature S271, "Basic multiset support",
conforming SQL language shall not contain a <collection derived table>.

Subclause 8.16, "<member predicate>":

<member predicate> ::=
<row value predicand> <member predicate part 2>

<member predicate part 2> ::=
[ NOT ] MEMBER [ OF ] <multiset value expression>

...

Conformance Rules
```

```
Without Feature S271, "Basic multiset support", conforming SQL language shall not contain a
<member predicate>.

NOTE — The Conformance Rules of Subclause 9.11, "Equality operations", also apply.

Subclause 9.11, "Equality operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an equality operation shall not be ST-ordered.

Without Feature T042, "Extended LOB data type support", in conforming SQL language, the
declared type of an operand of an equality operation shall not be LOB-ordered.

Without Feature S275, "Advanced multiset support", in conforming SQL language, the declared
type of an operand of an equality operation shall not be multiset-ordered.

NOTE — If the declared type of an operand OP of an equality operation is a multiset type,
then OP is a multiset operand of a multiset element grouping operation. The Conformance Rules
of Subclause 9.13, "Multiset element grouping operations", apply.

Subclause 8.18, "<set predicate>":

<set predicate> ::=
<row value predicand> <set predicate part 2>

<set predicate part 2> ::=
IS [ NOT ] A SET

...

Conformance Rules

Without Feature S271, "Basic multiset support", conforming SQL language shall not contain a
<set predicate>.

NOTE — The Conformance Rules of Subclause 9.13, "Multiset element grouping operations", also
apply.

Subclause 10.9, "<aggregate function>":

<computational operation> ::=
AVG
| MAX
| MIN
| SUM
| EVERY
| ANY
| SOME
| COUNT
| STDDEV_POP
| STDDEV_SAMP
| VAR_SAMP
| VAR_POP
| COLLECT
| FUSION
| INTERSECTION
```

```
...

Conformance Rules

Without Feature S271, "Basic multiset support", conforming SQL language shall not contain a
<computational operation> that immediately contains COLLECT.
```

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

Transact-SQL does not support this feature.

2.1.2.162 S272, Multisets of user-defined types

V0198:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<path-resolved user-defined type name> ::=
<user-defined type name>

<collection type> ::=
<array type>
| <multiset type>

...

<multiset type> ::=
<data type> MULTISET


...

Conformance Rules

Without Feature S272, "Multisets of user-defined types", conforming SQL language shall not
contain a <multiset type> that is based on a <data type> that contains a <path-resolved user-
defined type name>.
```

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

Transact-SQL does not support this feature.

2.1.2.163 S274, Multisets of reference types

V0199:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<reference type> ::=
REF <left paren> <referenced type> <right paren> [ <scope clause> ]

...

<collection type> ::=
<array type>
| <multiset type>

...

<multiset type> ::=
<data type> MULTISET

...

Conformance Rules

Without Feature S274, "Multisets of reference types", conforming SQL language shall not
contain a <multiset type> that is based on a <data type> that contains a <reference type>.
```

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

Transact-SQL does not support this feature.

2.1.2.164 S275, Advanced multiset support

V0200:

The specification states the following:

```
Subclause 6.39, "<multiset value expression>":

<multiset value expression> ::=
<multiset term>
| <multiset value expression> MULTISET UNION [ ALL | DISTINCT ] <multiset term>
| <multiset value expression> MULTISET EXCEPT [ ALL | DISTINCT ] <multiset term>

<multiset term> ::=
```

```
<multiset primary>
| <multiset term> MULTISET INTERSECT [ ALL | DISTINCT ] <multiset primary>

...
```

Conformance Rules

Without Feature S275, "Advanced multiset support", conforming SQL language shall not contain MULTISET UNION, MULTISET INTERSECTION, or MULTISET EXCEPT.

NOTE — If MULTISET UNION DISTINCT, MULTISET INTERSECTION, or MULTISET EXCEPT is specified, then the Conformance Rules of Subclause 9.13, "Multiset element grouping operations", also apply.

Subclause 9.13, "Multiset element grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared element type of a multiset operand of a multiset element grouping operation shall not be ST-ordered.

Subclause 8.17, "<submultiset predicate>":

```
<submultiset predicate> ::=
<row value predicand> <submultiset predicate part 2>

<submultiset predicate part 2> ::=
[ NOT ] SUBMULTISET [ OF ] <multiset value expression>

...
```

Conformance Rules

Without Feature S275, "Advanced multiset support", conforming SQL language shall not contain a <submultiset predicate>.

NOTE — The Conformance Rules of Subclause 9.13, "Multiset element grouping operations", also apply.

Subclause 9.11, "Equality operations":

Conformance Rules

Without Feature S275, "Advanced multiset support", in conforming SQL language, the declared type of an operand of an equality operation shall not be multiset-ordered.

NOTE — If the declared type of an operand OP of an equality operation is a multiset type, then OP is a multiset operand of a multiset element grouping operation. The Conformance Rules of Subclause 9.13, "Multiset element grouping operations", apply.

Subclause 10.9, "<aggregate function>":

```
<computational operation> ::=
AVG
| MAX
| MIN
| SUM
| EVERY
| ANY
```

```
| SOME
| COUNT
| STDDEV_POP
| STDDEV_SAMP
| VAR_SAMP
| VAR_POP
| COLLECT
| FUSION
| INTERSECTION

...

Conformance Rules

Without Feature S275, "Advanced multiset support", conforming SQL language shall not contain
a <computational operation> that immediately contains FUSION or INTERSECTION.

NOTE — If INTERSECTION is specified, then the Conformance Rules of Subclause 9.13, "Multiset
element grouping operations", also apply.
```

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

Transact-SQL does not support this feature.

2.1.2.165 S281, Nested collection types

V0201:

The specification states the following:

```
Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<collection type> ::=
<array type>
| <multiset type>

...

Conformance Rules

Without Feature S281, "Nested collection types", conforming SQL language shall not contain a
collection type that is based on a <data type> that contains a <collection type>.
```

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

Transact-SQL does not support this feature. Transact-SQL does not support either multisets or arrays.

2.1.2.166 S291, Unique constraint on entire row

V0202:

The specification states the following:

```
Subclause 11.7, "<unique constraint definition>":

<unique constraint definition> ::=
<unique specification> <left paren> <unique column list> <right paren>
| UNIQUE ( VALUE )

<unique specification> ::=
UNIQUE
| PRIMARY KEY

<unique column list> ::=
<column name list>

...

Conformance Rules

Without Feature S291, "Unique constraint on entire row", conforming SQL language shall not
contain UNIQUE(VALUE).
```

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

Transact-SQL does not support this feature.

2.1.2.167 S301, Enhanced UNNEST

V0203:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<collection derived table> ::=
UNNEST <left paren> <collection value expression>
[ { <comma> <collection value expression> }... ] <right paren>
[ WITH ORDINALITY ]

...

Conformance Rules

Without Feature S301, "Enhanced UNNEST", in conforming SQL language, a <collection derived
table> shall not simply contain more than one <collection value expression>.
```

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

Transact-SQL does not support this feature.

2.1.2.168 S401, Distinct types based on array types

V0204:

The specification states the following:

```
Subclause 6.1, "<data type>":

<collection type> ::=
<array type>
| <multiset type>

<array type> ::=
<data type> ARRAY
[ <left bracket or trigraph> <maximum cardinality> <right bracket or trigraph> ]

Subclause 11.51, "<user-defined type definition>":

<representation> ::=
<predefined type>
| <collection type>
| <member list>

...

Conformance Rules:

Without Feature S401, "Distinct types based on array types", in conforming SQL language,
<representation> shall not contain <array type>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.169 S402, Distinct types based on multiset types

V0205:

The specification states the following:

```
Subclause 6.1, "<data type>":

<collection type> ::=
<array type>
| <multiset type>

...

<multiset type> ::=
<data type> MULTISET [Subclause 6.1]

Subclause 11.51, "<user-defined type definition>":
```

```
<representation> ::=
<predefined type>
| <collection type>
| <member list>

...

Conformance Rules:

Without Feature S402, "Distinct types based on multiset types", in conforming SQL language,
epresentation> shall not contain <multiset type>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.170 S403, ARRAY_MAX_CARDINALITY

V0206:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<max cardinality expression> ::=
ARRAY_MAX_CARDINALITY <left paren> <array value expression> <right paren>

...

Conformance Rules:

Without Feature S403, “ARRAY_MAX_CARDINALITY”, conforming SQL language shall not contain <max
cardinality expression>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.171 S404, TRIM_ARRAY

V0207:

The specification states the following:

```
Subclause 6.37, "<array value function>":

<trim array function> ::=
TRIM_ARRAY <left paren> <array value expression> <comma> <numeric value expression> <right
```

```
paren>

...

Conformance Rules:

Without Feature S404, "TRIM_ARRAY", conforming SQL language shall not contain a <trim array
function>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.172 T011, Timestamp in Information Schema

V0208:

The specification states the following:

```
Subclause 6.1, "<data type>": The TIMESTAMP <datetime type> without the <with or without time
zone> clause

<datetime type> ::=
DATE
...
| TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]
...
```

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

Transact-SQL does not support this feature. The Transact-SQL **timestamp** data type is different from the **timestamp** data type that is defined in [ISO/IEC9075-2:2011].

For more details, see feature F051-03 (section 2.1.1.27).

2.1.2.173 T022, Advanced support for BINARY and VARBINARY data types

V0209:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<binary value function> ::=
<binary substring function>
| <binary trim function>
| <binary overlay function>

...

Conformance Rules
```

```
Without Feature T042, "Extended LOB data type support", or Feature T022, "Advanced support
for BINARY and VARBINARY data types", conforming SQL language shall not contain a <binary
value function>.

Subclause 8.5, "<like predicate>":

<octet like predicate> ::=
<row value predicand> <octet like predicate part 2>

<octet like predicate part 2> ::=
[ NOT ] LIKE <octet pattern> [ ESCAPE <escape octet> ]

<octet pattern> ::=
<binary value expression>

<escape octet> ::=
<binary value expression>

...

Conformance Rules

Without Feature T042, "Extended LOB data type support", or Feature T022, "Advanced support
for BINARY and VARBINARY data types", conforming SQL language shall not contain an <octet
like predicate>.
```

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

Transact-SQL does not support this feature.

2.1.2.174 T023, Compound binary literals

V0210:

The specification states the following:

```
Subclause 5.3, "<literal>":

<binary string literal> ::=
X <quote> [ <space>... ] [ { <hexit> [ <space>... ] <hexit> [ <space>... ] }... ] <quote>
[ { <separator> <quote> [ <space>... ] [ { <hexit> [ <space>... ]
<hexit> [ <space>... ] }... ] <quote> }... ]

<hexit> ::=
<digit> | A | B | C | D | E | F | a | b | c | d | e | f

...

Conformance Rules

Without Feature T023, "Compound binary literals", in conforming SQL language, a <binary
string literal> shall contain exactly one repetition of "<quote> [ { <hexit> <hexit> }... ]
<quote>".
```

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

Transact-SQL does not support this feature.

2.1.2.175 T024, Spaces in binary literals

V0211:

The specification states the following:

```
Subclause 5.1, "<SQL terminal character>":

<space> ::=
!! See the Syntax Rules.

...

Syntax Rules
1) Every character set shall contain a <space> character that is equivalent to U+0020.

Subclause 5.3, "<literal>":

<binary string literal> ::=
X <quote> [ <space>... ] [ { <hexit> [ <space>... ] <hexit> [ <space>... ] }... ] <quote>
[ { <separator> <quote> [ <space>... ] [ { <hexit> [ <space>... ]
<hexit> [ <space>... ] }... ] <quote> }... ]

<hexit> ::=
<digit> | A | B | C | D | E | F | a | b | c | d | e | f

...

Conformance Rules

Without Feature T024, "Spaces in binary literals", in conforming SQL language, a <binary
string literal> shall not contain a <space>.
```

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

Transact-SQL does not support this feature.

2.1.2.176 T031, BOOLEAN data type

V0212:

The specification states the following:

```
Subclause 5.3, "<literal>":

<boolean literal> ::=
TRUE
| FALSE
| UNKNOWN

...

Conformance Rules
```

```
Without Feature T031, "BOOLEAN data type", conforming SQL language shall not contain a
<boolean literal>.

Subclause 6.1, "<data type>":

<boolean type> ::=
BOOLEAN

...

Conformance Rules

Without Feature T031, "BOOLEAN data type", conforming SQL language shall not contain a
<boolean type>.

Subclause 6.26, "<value expression>":

<value expression> ::=
<common value expression>
| <boolean value expression>
| <row value expression>

...

Conformance Rules

Without Feature T031, "BOOLEAN data type", conforming SQL language shall not contain a <value
expression> that is a <boolean value expression>.

Subclause 6.35, "<boolean value expression>":

<boolean value expression> ::=
<boolean term>
| <boolean value expression> OR <boolean term>

<boolean term> ::=
<boolean factor>
| <boolean term> AND <boolean factor>

<boolean factor> ::=
[ NOT ] <boolean test>

<boolean test> ::=
<boolean primary> [ IS [ NOT ] <truth value> ]


...

<boolean primary> ::=
<predicate>
| <boolean predicand>

<boolean predicand> ::=
<parenthesized boolean value expression>
| <nonparenthesized value expression primary>

...

Conformance Rules
```

```
Without Feature T031, "BOOLEAN data type", conforming SQL language shall not contain a
<boolean primary> that simply contains a <nonparenthesized value expression primary>.

Subclause 6.3, "<value expression primary>":

<nonparenthesized value expression primary> ::=
<unsigned value specification>
| <column reference>
| <set function specification>
| <window function>
| <scalar subquery>
| <case expression>
| <cast specification>
| <field reference>
| <subtype treatment>
| <method invocation>
| <static method invocation>
| <new specification>
| <attribute or method reference>
| <reference resolution>
| <collection value constructor>
| <array element reference>
| <multiset element reference>
| <next value expression>
| <routine invocation>

Subclause 7.1, "<row value constructor>":

<row value constructor predicand> ::=
<common value expression>
| <boolean predicand>
| <explicit row value constructor>

...

Conformance Rules

Without Feature T031, "BOOLEAN data type", conforming SQL language shall not contain a <row
value constructor predicand> that immediately contains a <Boolean predicand>.

Subclause 10.9, "<aggregate function>":

<computational operation> ::=
AVG
| MAX
| MIN
| SUM
| EVERY
| ANY
| SOME
| COUNT
| STDDEV_POP
| STDDEV_SAMP
| VAR_SAMP
| VAR_POP
| COLLECT
| FUSION
| INTERSECTION
```

```
...

Conformance Rules

Without Feature T031, "BOOLEAN data type", conforming SQL language shall not contain a
<computational operation> that immediately contains EVERY, ANY, or SOME.
```

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

Transact-SQL does not support this feature.

2.1.2.177 T041, Basic LOB data type support

V0213:

The specification states the following:

```
Subclause 5.3, "<literal>":

<binary string literal> ::=
X <quote> [ <space>... ] [ { <hexit> [ <space>... ] <hexit> [ <space>... ] }... ] <quote>
[ { <separator> <quote> [ <space>... ] [ { <hexit> [ <space>... ]
<hexit> [ <space>... ] }... ] <quote> }... ]

<hexit> ::=
<digit> | A | B | C | D | E | F | a | b | c | d | e | f

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", or Feature T021, "BINARY and VARBINARY
data types", conforming SQL language shall not contain a <binary string literal>.

Subclause 6.1, "<data type>":

<data type> ::=
<predefined type>
| <row type>
| <path-resolved user-defined type name>
| <reference type>
| <collection type>

...

<character large object type> ::=
CHARACTER LARGE OBJECT [ <left paren> <character large object length> <right paren> ]
| CHAR LARGE OBJECT [ <left paren> <character large object length> <right paren> ]
| CLOB [ <left paren> <character large object length> <right paren> ]

...

<national character large object type> ::=
NATIONAL CHARACTER LARGE OBJECT [ <left paren> <character large object length> <right
paren> ]
| NCHAR LARGE OBJECT [ <left paren> <character large object length> <right paren> ]
```

```
| NCLOB [ <left paren> <character large object length> <right paren> ]

...

<binary large object string type> ::=
BINARY LARGE OBJECT [ <left paren> <large object length> <right paren> ]
| BLOB [ <left paren> <large object length> <right paren> ]

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <binary large object string type>, a <character large object type>, or a <national
character large object type>.

Subclause 11.60, "<SQL-invoked routine>":

<parameter type> ::=
<data type> [ <locator indication> ]

<locator indication> ::=
AS LOCATOR

...

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <parameter type> that contains a <locator indication> and that simply contains a
<data type> that identifies a large object type.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <returns data type> that contains a <locator indication> and that simply contains a
<data type> that identifies a large object type.

Subclause 21.3, "<embedded SQL Ada program>":

<Ada CLOB variable> ::=
SQL TYPE IS CLOB <left paren> <character large object length> <right paren>
[ CHARACTER SET [ IS ] <character set specification> ]

<Ada CLOB locator variable> ::=
SQL TYPE IS CLOB AS LOCATOR

...

<Ada BLOB variable> ::=
SQL TYPE IS BLOB <left paren> <large object length> <right paren>

<Ada BLOB locator variable> ::=
SQL TYPE IS BLOB AS LOCATOR

...
```

```
Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain an <Ada BLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain an <Ada CLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain an <Ada BLOB locator variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain an <Ada CLOB locator variable>.

Subclause 21.4, "<embedded SQL C program>":

<C CLOB variable> ::=
SQL TYPE IS CLOB <left paren> <character large object length> <right paren>
[ CHARACTER SET [ IS ] <character set specification> ]
<C host identifier> [ <C initial value> ] [ { <comma> <C host identifier> [
<C initial value> ] }... ]

...

<C BLOB variable> ::=
SQL TYPE IS BLOB <left paren> <large object length> <right paren>
<C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [
<C initial value> ] } ... ]

<C CLOB locator variable> ::=
SQL TYPE IS CLOB AS LOCATOR
<C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [
<C initial value> ] } ... ]

<C BLOB locator variable> ::=
SQL TYPE IS BLOB AS LOCATOR
<C host identifier> [ <C initial value> ]
[ { <comma> <C host identifier> [
<C initial value> ] } ... ]

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <C BLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <C CLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <C BLOB locator variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <C CLOB locator variable>.

Subclause 21.5, "<embedded SQL COBOL program>":
```

```
<COBOL CLOB variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS CLOB <left paren> <character large object length> <right
paren>
[ CHARACTER SET [ IS ] <character set specification> ]

...

<COBOL BLOB variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS BLOB <left paren> <large object length> <right paren>

...

<COBOL CLOB locator variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS CLOB AS LOCATOR

<COBOL BLOB locator variable> ::=
[ USAGE [ IS ] ] SQL TYPE IS BLOB AS LOCATOR

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <COBOL BLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <COBOL CLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <COBOL BLOB locator variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <COBOL CLOB locator variable>.

Subclause 21.6, "<embedded SQL Fortran program>":

<Fortran CLOB variable> ::=
SQL TYPE IS CLOB <left paren> <character large object length> <right paren>
[ CHARACTER SET [ IS ] <character set specification> ]

...

<Fortran BLOB variable> ::=
SQL TYPE IS BLOB <left paren> <large object length> <right paren>

...

<Fortran CLOB locator variable> ::=
SQL TYPE IS CLOB AS LOCATOR

<Fortran BLOB locator variable> ::=
SQL TYPE IS BLOB AS LOCATOR

...

Conformance Rules
```

```
Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Fortran BLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Fortran CLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Fortran BLOB locator variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Fortran CLOB locator variable>.

Subclause 21.7, "<embedded SQL MUMPS program>":

<MUMPS CLOB variable> ::=
SQL TYPE IS CLOB <left paren> <character large object length> <right paren>
[ CHARACTER SET [ IS ] <character set specification> ]

...

<MUMPS BLOB variable> ::=
SQL TYPE IS BLOB <left paren> <large object length> <right paren>

...

<MUMPS CLOB locator variable> ::=
SQL TYPE IS CLOB AS LOCATOR

<MUMPS BLOB locator variable> ::=
SQL TYPE IS BLOB AS LOCATOR

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <MUMPS BLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <MUMPS CLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <MUMPS BLOB locator variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <MUMPS CLOB locator variable>.

Subclause 21.8, "<embedded SQL Pascal program>":

<Pascal CLOB variable> ::=
SQL TYPE IS CLOB <left paren> <character large object length> <right paren>
[ CHARACTER SET [ IS ] <character set specification> ]

...

<Pascal BLOB variable> ::=
SQL TYPE IS BLOB <left paren> <large object length> <right paren>

<Pascal CLOB locator variable> ::=
SQL TYPE IS CLOB AS LOCATOR
```

```
...

<Pascal BLOB locator variable> ::=
SQL TYPE IS BLOB AS LOCATOR

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Pascal BLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Pascal CLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Pascal BLOB locator variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <Pascal BLOB variable>, <Pascal CLOB variable>, <Pascal CLOB locator variable>.

Subclause 21.9, "<embedded SQL PL/I program>":

<PL/I CLOB variable> ::=
SQL TYPE IS CLOB <left paren> <character large object length> <right paren>
[ CHARACTER SET [ IS ] <character set specification> ]

...

<PL/I BLOB variable> ::=
SQL TYPE IS BLOB <left paren> <large object length> <right paren>

...

<PL/I CLOB locator variable> ::=
SQL TYPE IS CLOB AS LOCATOR

<PL/I BLOB locator variable> ::=
SQL TYPE IS BLOB AS LOCATOR

...

Conformance Rules

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <PL/I BLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <PL/I CLOB variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <PL/I BLOB locator variable>.

Without Feature T041, "Basic LOB data type support", conforming SQL language shall not
contain a <PL/I CLOB locator variable>.
```

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

Transact-SQL does not support the large object (LOB) data type. See the **VARBINARY** [MSDN-binaryvarbinary] and **VARCHAR** [MSDN-charvarchar] or deprecated **IMAGE** and **TEXT** [MSDN-ntexttextimage] data types for equivalent functionality.

2.1.2.178 T041-01, BLOB data type

V0214:

The specification states the following:

```
BLOB data type
--Subclause 5.2, "<token> and <separator>": The <reserved word>s BINARY, BLOB, LARGE, and
OBJECT
--Subclause 5.3, "<literal>": <binary string literal>
--Subclause 6.1, "<data type>": The BINARY LARGE OBJECT data type
--Subclause 6.29, "<string value expression>": For values of type BINARY LARGE OBJECT
--Subclause 13.5, "Data type correspondences": Type correspondences for BINARY LARGE OBJECT
for all supported languages
```

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

Transact-SQL does not support the binary large object (BLOB) data type. See the **VARBINARY** [MSDN-binaryvarbinary] and deprecated **IMAGE** [MSDN-ntexttextimage] data types for equivalent functionality.

2.1.2.179 T041-02, CLOB data type

V0215:

The specification states the following:

```
CLOB data type
--Subclause 5.2, "<token> and <separator>": The <reserved word>s CHARACTER, CLOB, LARGE, and
OBJECT
--Subclause 6.1, "<data type>": The CHARACTER LARGE OBJECT data type
--Subclause 6.29, "<string value expression>": For values of type CHARACTER LARGE OBJECT
--Subclause 13.5, "Data type correspondences": Type correspondences for CHARACTER LARGE
OBJECT for all supported languages
--The implicit casting among the fixed-length and variable-length character string types
supported by subfeature E021-10 is extended to support the character large object type.
```

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

Transact-SQL does not support the character large object (CLOB) data type. See the **VARCHAR** [MSDN-charvarchar] and deprecated **TEXT** [MSDN-ntexttextimage] data types for equivalent functionality.

2.1.2.180 T041-03, POSITION, LENGTH, LOWER, TRIM, UPPER, and SUBSTRING functions for LOB data types

V0216:

The specification states the following:

```
POSITION, LENGTH, LOWER, TRIM, UPPER, and SUBSTRING functions for
```

```
LOB data types
--Subclause 6.28, "<numeric value function>": The <position expression> for expressions of
type BINARY LARGE OBJECT and CHARACTER LARGE OBJECT
--Subclause 6.28, "<numeric value function>": The <char length expression> for expressions of
type CHARACTER LARGE OBJECT
--Subclause 6.28, "<numeric value function>": The <octet length expression> for expressions
of type BINARY LARGE OBJECT and CHARACTER LARGE OBJECT
--Subclause 6.30, "<string value function>": The <fold> function for expressions of type
CHARACTER LARGE OBJECT
--Subclause 6.30, "<string value function>": The <trim function> for expressions of type
CHARACTER LARGE OBJECT
--Subclause 6.30, "<string value function>": The <binary trim function> for expressions of
type BINARY LARGE OBJECT
--Subclause 6.30, "<string value function>": The <character substring function> for
expressions of type CHARACTER LARGE OBJECT
--Subclause 6.30, "<string value function>": The <binary substring function> for expressions
of type BINARY LARGE OBJECT
```

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

Transact-SQL does not support this feature. Transact-SQL does not contain either the binary large object (**BLOB**) or character large object (**CLOB**) data types. However, some equivalent functions exist for the **VARBINARY** [MSDN-binaryvarbinary] and **VARCHAR** [MSDN-charvarchar] and deprecated **IMAGE** and **TEXT** [MSDN-ntexttextimage] data types.

2.1.2.181 T041-04, Concatenation of LOB data types

V0217:

The specification states the following:

```
Concatenation of LOB data types
--Subclause 6.29, "<string value expression>": The <concatenation> expression for expressions
of type CHARACTER LARGE OBJECT
--Subclause 6.29, "<string value expression>": The <binary concatenation> expression for
expressions of type BINARY LARGE OBJECT
```

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

Transact-SQL does not support this feature.

2.1.2.182 T041-05, LOB locator: non-holdable

V0218:

The specification states the following:

```
LOB locator: non-holdable
--Subclause 13.3, "<externally-invoked procedure>": <locator indication>
--Subclause 14.17, "<free locator statement>"
```

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

Transact-SQL does not support this feature.

2.1.2.183 T042, Extended LOB data type support

V0219:

The specification states the following:

```
Subclause 6.13, "<cast specification>":

<cast operand> ::=
<value expression>
| <implicitly typed value specification>

...

Conformance Rules

Without Feature T042, "Extended LOB data type support", conforming SQL language shall not
contain a <cast operand> whose declared type is BINARY LARGE OBJECT or CHARACTER LARGE
OBJECT.

...

Without Feature T042, "Extended LOB data type support", conforming SQL language shall not
contain a <cast operand> whose declared type is NATIONAL CHARACTER LARGE OBJECT.

Subclause 6.29, "<string value expression>":

<character value expression> ::=
<concatenation>
| <character factor>

Subclause 6.30, "<string value function>":

<binary value function> ::=
<binary substring function>
| <binary trim function>
| <binary overlay function>

...

Conformance Rules

Without Feature T042, "Extended LOB data type support", or Feature T022, "Advanced support
for BINARY and VARBINARY data types", conforming SQL language shall not contain a <binary
value function>.

Subclause 8.5, "<like predicate>":

<like predicate> ::=
<character like predicate>
| <octet like predicate>

...

<octet like predicate> ::=
<row value predicand> <octet like predicate part 2>

<octet like predicate part 2> ::=
[ NOT ] LIKE <octet pattern> [ ESCAPE <escape octet> ]
```

```
<octet pattern> ::=
<binary value expression>

<escape octet> ::=
<binary value expression>

...

Conformance Rules

Without Feature T042, "Extended LOB data type support", or Feature T022, "Advanced support
for BINARY and VARBINARY data types", conforming SQL language shall not contain an <octet
like predicate>.

...

Without Feature T042, "Extended LOB data type support", in conforming SQL language, a
<character value expression> simply contained in a <like predicate> shall not be of declared
type CHARACTER LARGE OBJECT

Without Feature F421, "National character", and Feature T042, "Extended LOB data type
support", in conforming SQL language, a <character value expression> simply contained in a
<like predicate> shall not be of declared type NATIONAL CHARACTER LARGE OBJECT.

Subclause 8.6, "<similar predicate>":

<similar predicate> ::=
<row value predicand> <similar predicate part 2>

<similar predicate part 2> ::=
[ NOT ] SIMILAR TO <similar pattern> [ ESCAPE <escape character> ]

<similar pattern> ::=
<character value expression>

...

Conformance Rules

Without Feature T042, "Extended LOB data type support", in conforming SQL language, a
<character value expression> simply contained in a <similar predicate> shall not be of
declared type CHARACTER LARGE OBJECT.

Subclause 9.11, "Equality operations":

Conformance Rules

Without Feature T042, "Extended LOB data type support", in conforming SQL language, the
declared type of an operand of an equality operation shall not be LOB-ordered.
```

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

Transact-SQL does not support this feature.

2.1.2.184 T043, Multiplier T

V0220:

The specification states the following:

```
Subclause 5.2, "<token> and <separator>":

<large object length token> ::=
<digit>... <multiplier>

<multiplier> ::=
K
| M
| G
| T
| P

...

Conformance Rules

Without Feature T043, "Multiplier T", in conforming SQL language, a <multiplier> shall not be
T.
```

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

Transact-SQL does not support this feature.

2.1.2.185 T044, Multiplier P

V0221:

The specification states the following:

```
Subclause 5.2, "<token> and <separator>":

<large object length token> ::=
<digit>... <multiplier>

<multiplier> ::=
K
| M
| G
| T
| P

...

Conformance Rules

Without Feature T044, "Multiplier P", in conforming SQL language, a <multiplier> shall not be
P.
```

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

Transact-SQL does not support this feature.

2.1.2.186 T051, Row types

V0222:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<field name> ::=
<identifier>

...

Conformance Rules

Without Feature T051, "Row types", conforming SQL language shall not contain a <field name>.

Subclause 6.1, "<data type>":

<row type> ::=
ROW <row type body>

<row type body> ::=
<left paren> <field definition> [ { <comma> <field definition> }... ] <right paren>

...

Conformance Rules

Without Feature T051, "Row types", conforming SQL language shall not contain a <row type>.

Subclause 6.2, "<field definition>":

<field definition> ::=
<field name> <data type>

...

Conformance Rules

Without Feature T051, "Row types", conforming SQL language shall not contain a <field
definition>.

Subclause 6.15, "<field reference>":

<field reference> ::=
<value expression primary> <period> <field name>

...

Conformance Rules

Without Feature T051, "Row types", conforming SQL language shall not contain a <field
reference>.

Subclause 7.1, "<row value constructor>":

<explicit row value constructor> ::=
<left paren> <row value constructor element> <comma>
```

```
<row value constructor element list> <right paren>
| ROW <left paren> <row value constructor element list> <right paren>
| <row subquery>

...

<contextually typed row value constructor> ::=
<common value expression>
| <boolean value expression>
| <contextually typed value specification>
| <left paren> <contextually typed value specification> <right paren>
| <left paren> <contextually typed row value constructor element> <comma>
<contextually typed row value constructor element list> <right paren>
| ROW <left paren> <contextually typed row value constructor element list> <right paren>

...

Conformance Rules

Without Feature T051, "Row types", conforming SQL language shall not contain an <explicit row
value constructor> that immediately contains ROW.

Without Feature T051, "Row types", conforming SQL language shall not contain a <contextually
typed row value constructor> that immediately contains ROW.

Subclause 7.2, "<row value expression>":

<row value special case> ::=
<nonparenthesized value expression primary>

...

Conformance Rules

Without Feature T051, "Row types", conforming SQL language shall not contain a <row value
special case>.

Subclause 7.12, "<query specification>":

<all fields reference> ::=
<value expression primary> <period> <asterisk>
[ AS <left paren> <all fields column name list> <right paren> ]

...

Conformance Rules

Without Feature T051, "Row types", conforming SQL language shall not contain an <all fields
reference>.
```

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

Transact-SQL does not support this feature.

2.1.2.187 T052, MAX and MIN for row types

V0223:

The [ISO/IEC9075-2:2008] specification states the following:

```
Subclause 6.26, "<value expression>":

<value expression> ::=
<common value expression>
| <boolean value expression>
| <row value expression>

Subclause 10.9, "<aggregate function>":

<computational operation> ::=
AVG
| MAX
| MIN
| SUM
| EVERY
| ANY
| SOME
| COUNT
| STDDEV_POP
| STDDEV_SAMP
| VAR_SAMP
| VAR_POP
| COLLECT
| FUSION
| INTERSECTION

...

Conformance Rules

Without Feature T052, "MAX and MIN for row types", conforming SQL language shall not contain
a <computational operation> that immediately contains MAX or MIN in which the declared type
of the <value expression> is a row type.

NOTE — If DISTINCT is specified, then the Conformance Rules of Subclause 9.10, "Grouping
operations", also apply. If MAX or MIN is specified, then the Conformance Rules of Subclause
9.12, "Ordering operations", also apply.

Subclause 9.10, "Grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of a grouping operation shall not be ST-ordered.

Subclause 9.12, "Ordering operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an ordering operation shall not be ST-ordered.
```

Microsoft SQL Server 2008 R2 varies as follows:

Transact-SQL does not support this feature.

Microsoft SQL Server 2012 varies as follows:

This feature is absent in the [ISO/IEC9075-02:2011] standard.

2.1.2.188 T053, Explicit aliases for all-fields reference

V0224:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<column name list> ::=
<column name> [ { <comma> <column name> }... ]

Subclause 7.12, "<query specification>":

<all fields column name list> ::=
<column name list>

...

Conformance Rules

Without Feature T053, "Explicit aliases for all-fields reference", conforming SQL language
shall not contain an <all fields column name list>.

NOTE — If a <set quantifier> DISTINCT is specified, then the Conformance Rules of Subclause
9.12, "Grouping operations", also apply.

Subclause 9.12, "Grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of a grouping operation shall not be ST-ordered.

Subclause 10.9, "<aggregate function>":

<set quantifier> ::=
DISTINCT
| ALL
```

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

Transact-SQL does not support this feature.

2.1.2.189 T061, UCS support

V0225:

The specification states the following:

```
Subclause 6.1, "<data type>":

<character length> ::=
<unsigned integer> [ <char length units> ]

...
```

```
<character large object length> ::=
<large object length> [ <char length units> ]

<char length units> ::=
CHARACTERS
| OCTETS

...

Conformance Rules

Without Feature T061, "UCS support", conforming SQL language shall not contain a <char length
units>.

Subclause 6.30, "<string value function>":

<normalize function> ::=
NORMALIZE <left paren> <character value expression>
[ <comma> <normal form> [ <comma> <normalize function result length> ] ] <right paren>

<normal form> ::=
NFC
| NFD
| NFKC
| NFKD

<normalize function result length> ::=
<character length>
| <character large object length>

...

Conformance Rules

Without Feature T061, "UCS support", conforming SQL language shall not contain a <normalize
function>.

Subclause 8.12, "<normalized predicate>":

<normalized predicate> ::=
<row value predicand> <normalized predicate part 2>

<normalized predicate part 2> ::=
IS [ NOT ] [ <normal form> ] NORMALIZED

...

Conformance Rules

Without Feature T061, "UCS support", conforming SQL language shall not contain a <normalized
predicate>.
```

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

Transact-SQL does not support this feature.

2.1.2.190 T122, WITH (excluding RECURSIVE) in subquery

V0226:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

<with clause> ::=
WITH [ RECURSIVE ] <with list>

...

Conformance Rules

Without Feature T122, "WITH (excluding RECURSIVE) in subquery", in conforming SQL language, a
<query expression> contained in a <query expression> shall not contain a <with clause>.
```

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

Transact-SQL does not support this feature. Transact-SQL supports a **WITH** keyword in queries, but the keyword takes a common table expression as a parameter.

See [ISO/IEC9075-1:2011] for the definition of "table".

2.1.2.191 T131, Recursive query

V0227:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

<with clause> ::=
WITH [ RECURSIVE ] <with list>

...

Conformance Rules

Without Feature T131, "Recursive query", conforming SQL language shall not contain a <query
expression> that contains RECURSIVE.

Subclause 11.32, "<view definition>":

<view definition> ::=
CREATE [ RECURSIVE ] VIEW <table name> <view specification>
AS <query expression> [ WITH [ <levels clause> ] CHECK OPTION ]
```

```
...

Conformance Rules

Without Feature T131, "Recursive query", conforming SQL language shall not contain a <view
definition> that immediately contains RECURSIVE.
```

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

Transact-SQL does not support this feature. Transact-SQL does not support the **RECURSIVE** keyword, although Transact-SQL does support some recursive functionality.

2.1.2.192 T132, Recursive query in subquery

V0228:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

<with clause> ::=
WITH [ RECURSIVE ] <with list>

...

Conformance Rules

Without Feature T132, "Recursive query in subquery", in conforming SQL language, a <query
expression> contained in a <query expression> shall not contain RECURSIVE.
```

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

Transact-SQL does not support this feature. Transact-SQL does not support the **RECURSIVE** keyword, although Transact-SQL does support some recursive functionality.

2.1.2.193 T141, SIMILAR predicate

V0229:

The specification states the following:

```
Subclause 8.6, "<similar predicate>":

<similar predicate> ::=
<row value predicand> <similar predicate part 2>

<similar predicate part 2> ::=
[ NOT ] SIMILAR TO <similar pattern> [ ESCAPE <escape character> ]

<similar pattern> ::=
```

```
<character value expression>

...

Conformance Rules

Without Feature T141, "SIMILAR predicate", conforming SQL language shall not contain a
<similar predicate>.
```

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

Transact-SQL does not support this feature.

2.1.2.194 T151, DISTINCT predicate

V0230:

The specification states the following:

```
Subclause 8.15, "<distinct predicate>":

<distinct predicate> ::=
<row value predicand 3> <distinct predicate part 2>

<distinct predicate part 2> ::=
IS [ NOT ] DISTINCT FROM <row value predicand 4>

<row value predicand 3> ::=
<row value predicand>

<row value predicand 4> ::=
<row value predicand>

...

Conformance Rules

Without Feature T151, "DISTINCT predicate", conforming SQL language shall not contain a
<distinct predicate>.

NOTE — The Conformance Rules of Subclause 9.11, "Equality operations", also apply.

Subclause 9.11, "Equality operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an equality operation shall not be ST-ordered.

Without Feature T042, "Extended LOB data type support", in conforming SQL language, the
declared type of an operand of an equality operation shall not be LOB-ordered.

Without Feature S275, "Advanced multiset support", in conforming SQL language, the declared
type of an operand of an equality operation shall not be multiset-ordered.
```

```
NOTE — If the declared type of an operand OP of an equality operation is a multiset type,
then OP is a multiset operand of a multiset element grouping operation. The Conformance Rules
of Subclause 9.13, "Multiset element grouping operations", apply.

Subclause 9.13, "Multiset element grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
element type of a multiset operand of a multiset element grouping operation shall not be ST-
ordered.
```

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

Transact-SQL does not support this feature.

2.1.2.195 T152, DISTINCT predicate with negation

V0231:

The specification states the following:

```
Subclause 8.15, "<distinct predicate>":

<distinct predicate> ::=
<row value predicand 3> <distinct predicate part 2>

<distinct predicate part 2> ::=
IS [ NOT ] DISTINCT FROM <row value predicand 4>

...

Conformance Rules

Without Feature T152, "DISTINCT predicate with negation", conforming SQL language shall not
contain a <distinct predicate part 2> that immediately contains NOT.
```

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

Transact-SQL does not support this feature.

2.1.2.196 T171, LIKE clause in table definition

V0232:

The specification states the following:

```
Subclause 11.3, "<table definition>":

<like clause> ::=
LIKE <table name> [ <like options> ]

...

Conformance Rules
```

```
Without Feature T171, "LIKE clause in table definition", conforming SQL language shall not
contain a <like clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.197 T172, AS subquery clause in table definition

V0233:

The specification states the following:

```
Subclause 11.3, "<table definition>":

<as subquery clause> ::=
[ <left paren> <column name list> <right paren> ] AS <table subquery>
<with or without data>

<with or without data> ::=
WITH NO DATA
| WITH DATA

...

Conformance Rules

Without Feature T172, "AS subquery clause in table definition", conforming SQL language shall
not contain an <as subquery clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.198 T173, Extended LIKE clause in table definition

V0234:

The specification states the following:

```
Subclause 11.3, "<table definition>":

<like clause> ::=
LIKE <table name> [ <like options> ]

<like options> ::=
<like option>...

<like option> ::=
<identity option>
| <column default option>
| <generation option>
```

```
<identity option> ::=
INCLUDING IDENTITY
| EXCLUDING IDENTITY

<column default option> ::=
INCLUDING DEFAULTS
| EXCLUDING DEFAULTS

<generation option> ::=
INCLUDING GENERATED
| EXCLUDING GENERATED

...

Conformance Rules

Without Feature T173, "Extended LIKE clause in table definition", a <like clause> shall not
contain <like options>.
```

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

Transact-SQL does not support this feature.

2.1.2.199 T174, Identity columns

V0235:

The specification states the following:

```
Subclause 11.4, "<column definition>":

<column definition> ::=
<column name> [ <data type or domain name> ]
[ <default clause> | <identity column specification> | <generation clause> ]
[ <column constraint definition>... ]
[ <collate clause> ]

...

<identity column specification> ::=
GENERATED { ALWAYS | BY DEFAULT } AS IDENTITY
[ <left paren> <common sequence generator options> <right paren> ]

...

Conformance Rules

Without Feature T174, "Identity columns", conforming SQL language shall not contain an
<identity column specification>.

Subclause 11.12, "<alter column definition>":

<alter column definition> ::=

ALTER [ COLUMN ] <column name> <alter column action>
```

```
<alter column action> ::=
<set column default clause>
| <drop column default clause>
| <add column scope clause>
| <drop column scope clause>
| <alter column data type clause>
| <alter identity column specification>

Subclause 11.20, "<alter identity column specification>":

<alter identity column specification> ::=
<alter identity column option>...

<alter identity column option> ::=
<alter sequence generator restart option>
| SET <basic sequence generator option>

...

Conformance Rules

Without Feature T174, "Identity columns", an <alter column definition> shall not contain an
<alter identity column specification>.
```

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

Transact-SQL does not support this feature. See the **IDENTITY** property [MSDN-IDENTITY] for equivalent functionality.

2.1.2.200 T175, Generated columns

V0236:

The specification states the following:

```
Subclause 11.4, "<column definition>":

<generation clause> ::=
<generation rule> AS <generation expression>

<generation rule> ::=
GENERATED ALWAYS

<generation expression> ::=
<left paren> <value expression> <right paren>

...

Conformance Rules

Without Feature T175, "Generated columns", conforming SQL language shall not contain a
<generation clause>.
```

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

Transact-SQL does not support this feature. Computed **columns** provide equivalent functionality.

See [ISO/IEC9075-2:2011] for the definition of "column".

2.1.2.201 T176, Sequence generator support

V0237:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<sequence generator name> ::=
<schema qualified name>

...

Conformance Rules

Without Feature T176, "Sequence generator support", conforming SQL language shall not contain
a <sequence generator name>.

Subclause 6.14, "<next value expression>":

<next value expression> ::=
NEXT VALUE FOR <sequence generator name>

...

Conformance Rules

Without Feature T176, "Sequence generator support", conforming SQL language shall not contain
a <next value expression>.

Subclause 11.72, "<sequence generator definition>":

<sequence generator definition> ::=
CREATE SEQUENCE <sequence generator name> [ <sequence generator options> ]

...

Conformance Rules

Without Feature T176, "Sequence generator support", conforming SQL language shall not contain
a <sequence generator definition>.

Subclause 11.73, "<alter sequence generator statement>":

<alter sequence generator statement> ::=
ALTER SEQUENCE <sequence generator name> <alter sequence generator options>

...

Conformance Rules

Without Feature T176, "Sequence generator support", conforming SQL language shall not contain
an <alter sequence generator statement>.

Subclause 11.74, "<drop sequence generator statement>":

<drop sequence generator statement> ::=
```

```
DROP SEQUENCE <sequence generator name> <drop behavior>

...

Conformance Rules

Without Feature T176, "Sequence generator support", conforming SQL language shall not contain
a <drop sequence generator statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.202 T178, Identity columns: simple restart option

V0238:

The specification states the following:

```
Subclause 11.20, "<alter identity column specification>":

<alter identity column specification> ::=
<set identity column generation clause> [ <alter identity column option>... ]
| <alter identity column option>...

<set identity column generation clause> ::=
SET GENERATED { ALWAYS | BY DEFAULT }

<alter identity column option> ::=
<alter sequence generator restart option>
| SET <basic sequence generator option>

...

Conformance Rules:

Without Feature T178, "Identity columns: simple restart option", in conforming SQL language,
an <alter sequence generator restart option> contained in an <alter identity column
specification> shall contain a <sequence generator restart value>.

Subclause 11.73, "<alter sequence generator statement>":

<alter sequence generator restart option> ::=
RESTART [ WITH <sequence generator restart value> ]

<sequence generator restart value> ::=
<signed numeric literal>
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.203 T180, System-versioned tables

V0239:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<query system time period specification> ::=
FOR SYSTEM_TIME AS OF <point in time 1>
| FOR SYSTEM_TIME BETWEEN [ ASYMMETRIC | SYMMETRIC ]
<point in time 1> AND <point in time 2>
| FOR SYSTEM_TIME FROM <point in time 1> TO <point in time 2>

...

Conformance Rules:

Without Feature T180, "System-versioned tables", conforming SQL language shall not contain
<query system time period specification>.

Subclause 11.3, "<table definition>":

<system versioning clause> ::=
SYSTEM VERSIONING

...

<table period definition> ::=
<system or application time period specification>
<left paren> <period begin column name> <comma> <period end column name> <right paren>

...

<system time period specification> ::=
PERIOD FOR SYSTEM_TIME

...

Conformance Rules:

Without Feature T180, "System-versioned tables", conforming SQL language shall not contain
"WITH <system versioning clause>" or a <table period definition> that specifies SYSTEM_TIME.

Subclause 11.4, "<column definition>":

<system time period start column specification> ::=
<timestamp generation rule> AS ROW START

<system time period end column specification> ::=
<timestamp generation rule> AS ROW END

...

Conformance Rules:

Without Feature T180, "System-versioned tables", conforming SQL language shall not contain
<system time period start column specification> or <system time period end column
specification>.
```

```
Subclause 11.27, "<add table period definition>":


<add table period definition> ::=
ADD <table period definition> [ <add system time period column list> ]

<add system time period column list> ::=
ADD [ COLUMN ] <column definition 1> ADD [ COLUMN ] <column definition 2>

...

Conformance Rules:

Without Feature T180, "System-versioned tables", conforming SQL language shall not contain
ADD <system time period specification>.

Subclause 11.28, "<drop table period definition>":

Conformance Rules:

Without Feature T180, "System-versioned tables", conforming SQL language shall not contain
DROP PERIOD SYSTEM_TIME.

Subclause 11.29, "<add system versioning clause>":

<add system versioning clause> ::=
ADD <system versioning clause>

...

Conformance Rules:

Without Feature T180, "System-versioned tables", conforming SQL language shall not contain
<add system versioning clause>.

Subclause 11.30, "<drop system versioning clause>":

<drop system versioning clause> ::=
DROP SYSTEM VERSIONING <drop behavior>

...

Conformance Rules:

Without Feature T180, "System-versioned tables", conforming SQL language shall not contain
<drop system versioning clause>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.204 T181, Application-time period tables

V0240:

The specification states the following:

```
Subclause 11.3, "<table definition>":

<table period definition> ::=
<system or application time period specification>
<left paren> <period begin column name> <comma> <period end column name> <right paren>

<system or application time period specification> ::=
<system time period specification>
| <application time period specification>

<system time period specification> ::=
PERIOD FOR SYSTEM_TIME

<application time period specification> ::=
PERIOD FOR <application time period name>

...

<application time period specification> ::=
PERIOD FOR <application time period name>

...

Conformance Rules:

Without Feature T181, "Application-time period tables", conforming SQL language shall not
contain a <table period definition> that contains an <application time period specification>.

Subclause 11.7, "<unique constraint definition>":

<without overlap specification> ::=
<application time period name> WITHOUT OVERLAPS

...

Conformance Rules:

Without Feature T181, "Application-time period tables", conforming SQL language shall not
contain <without overlap specification>.

Subclause 11.8, "<referential constraint definition>":

<referencing period specification> ::=
PERIOD <application time period name>

...

<referenced period specification> ::=
PERIOD <application time period name>

...

Conformance Rules:
```

```
Without Feature T181, "Application-time period tables", conforming SQL language shall not
contain a <referencing period specification>.

Without Feature T181, "Application-time period tables", conforming SQL language shall not
contain a <referenced period specification>.

Subclause 11.27, "<add table period definition>":

<add table period definition> ::=
ADD <table period definition> [ <add system time period column list> ]

<add system time period column list> ::=
ADD [ COLUMN ] <column definition 1> ADD [ COLUMN ] <column definition 2>

...

Conformance Rules:

Without Feature T181, "Application-time period tables", conforming SQL language shall not
contain ADD <application time period specification>.

Subclause 11.28, "<drop table period definition>":

<drop table period definition> ::=
DROP <system or application time period specification> <drop behavior>

...

Conformance Rules:

Without Feature T181, "Application-time period tables", conforming SQL language shall not
contain a <drop table period definition> that contains an <application time period name>.

Subclause 14.9, "<delete statement: searched>":

<delete statement: searched> ::=
DELETE FROM <target table>
[ FOR PORTION OF <application time period name>
FROM <point in time 1> TO <point in time 2> ]
[ [ AS ] <correlation name> ]
[ WHERE <search condition> ]

...

Conformance Rules:

Without Feature T181, "Application-time period tables", in conforming SQL language, a <delete
statement: searched> shall not contain FOR PORTION OF.

Subclause 14.14, "<update statement: searched>":

<update statement: searched> ::=
UPDATE <target table>
[ FOR PORTION OF <application time period name>
FROM <point in time 1> TO <point in time 2> ]
[ [ AS ] <correlation name> ]
SET <set clause list>
[ WHERE <search condition> ]
```

```
...

Conformance Rules:

Without Feature T181, "Application-time period tables", in conforming SQL language, an
<update statement: searched> shall not contain FOR PORTION OF.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.205 T191, Referential action RESTRICT

V0241:

The specification states the following:

```
Subclause 11.8, "<referential constraint definition>":

<update rule> ::=
ON UPDATE <referential action>

<delete rule> ::=
ON DELETE <referential action>

<referential action> ::=
CASCADE
| SET NULL
| SET DEFAULT
| RESTRICT
| NO ACTION

...

Conformance Rules

Without Feature T191, "Referential action RESTRICT", conforming SQL language shall not
contain a <referential action> that contains RESTRICT.
```

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

Transact-SQL does not support this feature.

2.1.2.206 T201, Comparable data types for referential constraints

V0242:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<column name list> ::=
<column name> [ { <comma> <column name> }... ]

Subclause 11.8, "<referential constraint definition>":

<referencing columns> ::=
<reference column list>

...

<reference column list> ::=
<column name list>

...

Conformance Rules

Without Feature T201, "Comparable data types for referential constraints", conforming SQL
language shall not contain a <referencing columns> in which the data type of each referencing
column is not the same as the data type of the corresponding referenced column.

NOTE — The Conformance Rules of Subclause 9.12, "Grouping operations", also apply.

Subclause 9.12, , "Grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of a grouping operation shall not be ST-ordered.
```

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

Transact-SQL does not support this feature.

2.1.2.207 T211, Basic trigger capability

V0243:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<table or query name> ::=
<table name>
| <transition table name>
| <query name>

...

Conformance Rules

Without Feature T211, "Basic trigger capability", conforming SQL language shall not contain a
<transition table name>.

Subclause 11.49, "<trigger definition>":
```

```
<trigger definition> ::=
CREATE TRIGGER <trigger name> <trigger action time> <trigger event>
ON <table name> [ REFERENCING <transition table or variable list> ]
<triggered action>

...

<transition table name> ::=
<identifier>

...

Conformance Rules

Without Feature T211, "Basic trigger capability", conforming SQL language shall not contain a
<trigger definition>.

Subclause 11.50, "<drop trigger statement>":

<drop trigger statement> ::=
DROP TRIGGER <trigger name>

...

Conformance Rules

Without Feature T211, "Basic trigger capability", conforming SQL language shall not contain a
<drop trigger statement>.

Subclause 12.3, "<privileges>":

Conformance Rules

Without Feature T211, "Basic trigger capability", conforming SQL language shall not contain
an <action> that contains TRIGGER.

Subclause 4.35.2, "Privileges":

A privilege descriptor with an <action> of INSERT, UPDATE, DELETE, SELECT, TRIGGER, or
REFERENCES is called a table privilege descriptor and identifies the existence of a privilege
on the table or view component identified by the privilege descriptor. If a table privilege
descriptor identifies a view component, the privilege descriptor is called a view component
table privilege descriptor.
```

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

Transact-SQL partially supports this feature.

2.1.2.208 T211-02, BEFORE triggers

V0244:

The specification states the following:

```
Subclause 4.39.1, "General description of triggers":
```

```
The triggered action is specified to take place either immediately before the triggering
event, instead of it, or immediately after it, according to its specified trigger action
time, BEFORE, INSTEAD OF, or AFTER. The trigger is a BEFORE trigger, an INSTEAD OF trigger,
or an AFTER trigger, according to its trigger action time.
```

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

Transact-SQL does not support this feature.

2.1.2.209 T211-04, FOR EACH ROW triggers

V0245:

The specification states the following:

```
Subclause 11.49, "<trigger definition>":

<trigger definition> ::=
CREATE TRIGGER <trigger name> <trigger action time> <trigger event>
ON <table name> [ REFERENCING <transition table or variable list> ]
<triggered action>

...

<triggered action> ::=
[ FOR EACH { ROW | STATEMENT } ]
[ <triggered when clause> ]
<triggered SQL statement>

...

Syntax Rules

Let OR, OT, NR, and NT be the <old transition variable name>, <old transition table name>,
<new transition variable name>, and <new transition table name>, respectively.

...

If neither FOR EACH ROW nor FOR EACH STATEMENT is specified, then FOR EACH STATEMENT is
implicit.

If OR or NR is specified, then FOR EACH ROW shall be specified.
```

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

Transact-SQL does not support this feature.

2.1.2.210 T211-05, Ability to specify a search condition that shall be True before the trigger is invoked

V0246:

The specification states the following:

```
Subclause 4.28.2, "Characteristics of SQL-invoked routines":
```

```
If a <routine invocation> is contained in the original <query expression> of a view, in the
<search condition> of a check constraint or an assertion, the <triggered action> of a
trigger, or in an <SQL-invoked routine>, then the subject routine for that invocation is
determined at the time the view is created, the check constraint is defined, the assertion is
created, the trigger is created, or the SQL-invoked routine is created. If a <routine
invocation> is contained in the hierarchical <query expression> of a view V, then the subject
routine for that invocation is determined at the time the subview SV of V whose original
<query expression> contains the <routine invocation> is created. If the subject routine is an
SQL-invoked procedure, an SQL-invoked regular function, or a static SQL-invoked method, then
the same SQL-invoked routine is executed whenever the view is used, the check constraint or
assertion is evaluated, the trigger is executed, or the SQL-invoked routine is invoked. If
the subject routine is an instance SQL-invoked method, then the SQL-invoked routine that is
executed is determined whenever the view is used, the check constraint or assertion is
evaluated, the trigger is executed, or the SQL-invoked routine is invoked, based on the most
specific type of the value resulting from the evaluation of the SQL argument that correspond
to the subject parameter.
```

The [ISO/IEC9075-2:2008] specification differs as follows:

```
Subclause 4.27.2, "Characteristics of SQL-invoked routines":

If a <routine invocation> is contained in a <query expression> of a view, in the <search
condition> of a check constraint or an assertion, the <triggered action> of a trigger, or in
an <SQL-invoked routine>, then the subject routine for that invocation is determined at the
time the view is created, the check constraint is defined, the assertion is created, the
trigger is created, or the SQL-invoked routine is created. If the subject routine is an SQL-
invoked procedure, an SQL-invoked regular function, or a static SQL-invoked method, then the
same SQL-invoked routine is executed whenever the view is used, the check constraint or
assertion is evaluated, the trigger is executed, or the SQL-invoked routine is invoked. If
the subject routine is an instance SQL-invoked method, then the SQL-invoked routine that is
executed is determined whenever the view is used, the check constraint or assertion is
evaluated, the trigger is executed, or the SQL-invoked routine is invoked, based on the most
specific type of the value resulting from the evaluation of the SQL argument that correspond
to the subject parameter.
```

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

Transact-SQL does not support this feature.

2.1.2.211 T211-06, Support for run-time rules for the interaction of triggers and constraints

V0247:

The specification states the following:

```
Subclause 4.28.2, "Characteristics of SQL-invoked routines":

If a <routine invocation> is contained in the original <query expression> of a view, in the
<search condition> of a check constraint or an assertion, the <triggered action> of a
trigger, or in an <SQL-invoked routine>, then the subject routine for that invocation is
determined at the time the view is created, the check constraint is defined, the assertion is
created, the trigger is created, or the SQL-invoked routine is created. If a <routine
invocation> is contained in the hierarchical <query expression> of a view V, then the subject
routine for that invocation is determined at the time the subview SV of V whose original
<query expression> contains the <routine invocation> is created. If the subject routine is an
SQL-invoked procedure, an SQL-invoked regular function, or a static SQL-invoked method, then
the same SQL-invoked routine is executed whenever the view is used, the check constraint or
assertion is evaluated, the trigger is executed, or the SQL-invoked routine is invoked. If
```

```
the subject routine is an instance SQL-invoked method, then the SQL-invoked routine that is
executed is determined whenever the view is used, the check constraint or assertion is
evaluated, the trigger is executed, or the SQL-invoked routine is invoked, based on the most
specific type of the value resulting from the evaluation of the SQL argument that correspond
to the subject parameter.
```

The [ISO/IEC9075-2:2008] specification differs as follows:

```
Subclause 4.27.2, "Characteristics of SQL-invoked routines":

If a <routine invocation> is contained in a <query expression> of a view, in the <search
condition> of a check constraint or an assertion, the <triggered action> of a trigger, or in
an <SQL-invoked routine>, then the subject routine for that invocation is determined at the
time the view is created, the check constraint is defined, the assertion is created, the
trigger is created, or the SQL-invoked routine is created. If the subject routine is an SQL-
invoked procedure, an SQL-invoked regular function, or a static SQL-invoked method, then the
same SQL-invoked routine is executed whenever the view is used, the check constraint or
assertion is evaluated, the trigger is executed, or the SQL-invoked routine is invoked. If
the subject routine is an instance SQL-invoked method, then the SQL-invoked routine that is
executed is determined whenever the view is used, the check constraint or assertion is
evaluated, the trigger is executed, or the SQL-invoked routine is invoked, based on the most
specific type of the value resulting from the evaluation of the SQL argument that correspond
to the subject parameter.
```

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

Transact-SQL does not support this feature.

2.1.2.212 T211-07, TRIGGER privilege

V0248:

The specification states the following:

```
Subclause 4.35.2, "Privileges":

A privilege descriptor with an <action> of INSERT, UPDATE, DELETE, SELECT, TRIGGER, or
REFERENCES is called a table privilege descriptor and identifies the existence of a privilege
on the table or view component identified by the privilege descriptor. If a table privilege
descriptor identifies a view component, the privilege descriptor is called a view component
table privilege descriptor.
```

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

Transact-SQL does not support this feature.

2.1.2.213 T211-08, Multiple triggers for the same event are executed in the order in which they were created in the catalog

V0249:

The specification states the following:

```
Subclause 4.39.1, "General description of triggers":
```

```
The order of execution of a set of triggers is ascending by value of their timestamp of
creation in their descriptors, such that the oldest trigger executes first. If one or more
triggers have the same timestamp value, then their relative order of execution is
implementation-defined.
```

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

Transact-SQL does not support this feature.

2.1.2.214 T212, Enhanced trigger capability

V0250:

The specification states the following:

```
Subclause 11.49, "<trigger definition>":

<trigger definition> ::=
CREATE TRIGGER <trigger name> <trigger action time> <trigger event>
ON <table name> [ REFERENCING <transition table or variable list> ]
<triggered action>

...

<triggered action> ::=
[ FOR EACH { ROW | STATEMENT } ]
[ <triggered when clause> ]
<triggered SQL statement>

...

Conformance Rules

Without Feature T212, "Enhanced trigger capability", in conforming SQL language, a <triggered
action> shall contain FOR EACH ROW.
```

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

Transact-SQL does not support this feature. Transact-SQL does not support **FOR EACH ROW** triggers in general (see feature T211-04, section 2.1.2.209).

See [ISO/IEC9075-1:2011] for the definition of "trigger".

2.1.2.215 T231, Sensitive cursors

V0251:

The specification states the following:

```
Subclause 14.2, "<cursor properties>":

<cursor properties> ::=
[ <cursor sensitivity> ] [ <cursor scrollability> ] CURSOR
[ <cursor holdability> ]
[ <cursor returnability> ]
```

```
<cursor sensitivity> ::=
SENSITIVE
| INSENSITIVE
| ASENSITIVE

...

Conformance Rules

Without Feature T231, "Sensitive cursors", conforming SQL language shall not contain a
<cursor sensitivity> that immediately contains SENSITIVE.

...

Without Feature F791, "Insensitive cursors", or Feature T231, "Sensitive cursors", conforming
SQL language shall not contain a <cursor sensitivity> that immediately contains ASENSITIVE.
```

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

Transact-SQL does not support this feature.

2.1.2.216 T241, START TRANSACTION statement

V0252:

The specification states the following:

```
Subclause 17.1, "<start transaction statement>":

<start transaction statement> ::=
START TRANSACTION [ <transaction characteristics> ]

...

Conformance Rules

Without Feature T241, "START TRANSACTION statement", conforming SQL language shall not
contain a <start transaction statement>.
```

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

Transact-SQL does not support this feature. See the **BEGIN TRANSACTION** [MSDN-BEGIN] statement for equivalent functionality.

2.1.2.217 T251, SET TRANSACTION statement: LOCAL option

V0253:

The specification states the following:

```
Subclause 17.2, "<set transaction statement>":

<set transaction statement> ::=
SET [ LOCAL ] TRANSACTION <transaction characteristics>
```

```
...

Conformance Rules

Without Feature T251, "SET TRANSACTION statement: LOCAL option", conforming SQL language
shall not contain a <set transaction statement> that immediately contains LOCAL.
```

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

Transact-SQL does not support this feature.

2.1.2.218 T261, Chained transactions

V0254:

The specification states the following:

```
Subclause 17.7, "<commit statement>":

<commit statement> ::=
COMMIT [ WORK ] [ AND [ NO ] CHAIN ]

...

Conformance Rules

Without Feature T261, "Chained transactions", conforming SQL language shall not contain a
<commit statement> that immediately contains CHAIN.

Subclause 17.8, "<rollback statement>":

<rollback statement> ::=
ROLLBACK [ WORK ] [ AND [ NO ] CHAIN ] [ <savepoint clause> ]

<savepoint clause> ::=
TO SAVEPOINT <savepoint specifier>

...

Conformance Rules

Without Feature T261, "Chained transactions", conforming SQL language shall not contain a
<rollback statement> that immediately contains CHAIN.
```

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

Transact-SQL does not support this feature.

2.1.2.219 T271, Savepoints

V0255:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<savepoint name> ::=
<identifier>

...

Conformance Rules

Without Feature T271, "Savepoints", conforming SQL language shall not contain a <savepoint
name>.

Subclause 17.5, "<savepoint statement>":

<savepoint statement> ::=
SAVEPOINT <savepoint specifier>

<savepoint specifier> ::=
<savepoint name>

...

Conformance Rules

Without Feature T271, "Savepoints", conforming SQL language shall not contain a <savepoint
statement>.

Subclause 17.6, "<release savepoint statement>":

<release savepoint statement> ::=
RELEASE SAVEPOINT <savepoint specifier>

...

Conformance Rules

Without Feature T271, "Savepoints", conforming SQL language shall not contain a <release
savepoint statement>.

Subclause 17.8, "<rollback statement>":

<rollback statement> ::=
ROLLBACK [ WORK ] [ AND [ NO ] CHAIN ] [ <savepoint clause> ]

<savepoint clause> ::=
TO SAVEPOINT <savepoint specifier>

...

Conformance Rules

Without Feature T271, "Savepoints", conforming SQL language shall not contain a <savepoint
clause>.
```

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

Transact-SQL does not support this feature. See the **SAVE TRANSACTION** function [MSDN-SAVETRAN] for equivalent functionality.

2.1.2.220 T272, Enhanced savepoint management

V0256:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<routine characteristics> ::=
[ <routine characteristic>... ]

<routine characteristic> ::=
<language clause>
| <parameter style clause>
| SPECIFIC <specific name>
| <deterministic characteristic>
| <SQL-data access indication>
| <null-call clause>
| <returned result sets characteristic>
| <savepoint level indication>

<savepoint level indication> ::=
NEW SAVEPOINT LEVEL
| OLD SAVEPOINT LEVEL

...

Conformance Rules

Without Feature T272, "Enhanced savepoint management", conforming SQL language shall not
contain a <routine characteristics> that contains a <savepoint level indication>.
```

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

Transact-SQL does not support this feature.

2.1.2.221 T301, Functional dependencies

V0257:

The specification states the following:

```
Subclause 8.21, "<search condition>":

<search condition> ::=
<boolean value expression>

Subclause 7.10, "<having clause>":

<having clause> ::=
HAVING <search condition>

...
```

```
Conformance Rules

Without Feature T301, "Functional dependencies", in conforming SQL language, each column
reference directly contained in the <search condition> shall be one of the following:
a) An unambiguous reference to a grouping column of T.
b) An outer reference.

Without Feature T301, "Functional dependencies", in conforming SQL language, each column
reference contained in a <query expression> in the <search condition> that references a
column of T shall be one of the following:
a) An unambiguous reference to a grouping column of T.
b) Contained in an aggregated argument of a <set function specification>.

Subclause 7.13, "<query expression>":

<query expression> ::=
[ <with clause> ] <query expression body>
[ <order by clause> ] [ <fetch first clause> ]

Subclause 7.11, "<window clause>":

<window clause> ::=
WINDOW <window definition list>

...

Conformance Rules

Without Feature T301, "Functional dependencies", in conforming SQL language, if T is a
grouped table, then each column reference contained in <window clause> that references a
column of T shall be a reference to a grouping column of T or be contained in an aggregated
argument of a <set function specification>.

Subclause 6.9, "<set function specification>":

<set function specification> ::=
<aggregate function>
| <grouping operation>

<grouping operation> ::=
GROUPING <left paren> <column reference>
[ { <comma> <column reference> }... ] <right paren>

Subclause 6.26, "<value expression>":

<value expression> ::=
<common value expression>
| <boolean value expression>
| <row value expression>

Subclause 6.7, "<column reference>":

<column reference> ::=
<basic identifier chain>
| MODULE <period> <qualified identifier> <period> <column name>

Subclause 7.12, "<query specification>":

<select list> ::=
```

```
<asterisk>
| <select sublist> [ { <comma> <select sublist> }... ]

<select sublist> ::=
<derived column>
| <qualified asterisk>

...

Conformance Rules

Without Feature T301, "Functional dependencies", in conforming SQL language, if T is a
grouped table, then in each <value expression> contained in the <select list>, each <column
reference> that references a column of T shall reference a grouping column or be specified in
an aggregated argument of a <set function specification>.

Subclause 20.4, "<get descriptor statement>":

<descriptor item name> ::=
CARDINALITY
| CHARACTER_SET_CATALOG
| CHARACTER_SET_NAME
| CHARACTER_SET_SCHEMA
| COLLATION_CATALOG
| COLLATION_NAME
| COLLATION_SCHEMA
| DATA
| DATETIME_INTERVAL_CODE
| DATETIME_INTERVAL_PRECISION
| DEGREE
| INDICATOR
| KEY_MEMBER
| LENGTH
| LEVEL
| NAME
| NULLABLE
| OCTET_LENGTH
| PARAMETER_MODE
| PARAMETER_ORDINAL_POSITION
| PARAMETER_SPECIFIC_CATALOG
| PARAMETER_SPECIFIC_NAME
| PARAMETER_SPECIFIC_SCHEMA
| PRECISION
| RETURNED_CARDINALITY
| RETURNED_LENGTH
| RETURNED_OCTET_LENGTH
| SCALE
| SCOPE_CATALOG
| SCOPE_NAME
| SCOPE_SCHEMA
| TYPE
| UNNAMED
| USER_DEFINED_TYPE_CATALOG
| USER_DEFINED_TYPE_NAME
| USER_DEFINED_TYPE_SCHEMA
| USER_DEFINED_TYPE_CODE

...
```

```
Conformance Rules

Without Feature T301, "Functional dependencies", conforming SQL language shall not contain a
<descriptor item name> that contains KEY_MEMBER.
```

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

Transact-SQL partially supports this feature.

2.1.2.222 T312, OVERLAY function

V0258:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<character overlay function> ::=
OVERLAY <left paren> <character value expression> PLACING <character value expression>
FROM <start position> [ FOR <string length> ]
[ USING <char length units> ] <right paren>

...

<binary overlay function> ::=
OVERLAY <left paren> <binary value expression> PLACING <binary value expression>
FROM <start position> [ FOR <string length> ] <right paren>

...

Conformance Rules

Without Feature T312, "OVERLAY function", conforming SQL language shall not contain a
<character overlay function>.

Without Feature T312, "OVERLAY function", conforming SQL language shall not contain a <binary
overlay function>.
```

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

Transact-SQL does not support this feature. See the **SUBSTRING** function [MSDN-SUBSTRING] and the concatenation operator **+** [MSDN-Add] for equivalent functionality.

2.1.2.223 T323, Explicit security for external routines

V0259:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<external body reference> ::=
EXTERNAL [ NAME <external routine name> ]
[ <parameter style clause> ]
[ <transform group specification> ]
[ <external security clause> ]
```

```
<external security clause> ::=
EXTERNAL SECURITY DEFINER
| EXTERNAL SECURITY INVOKER
| EXTERNAL SECURITY IMPLEMENTATION DEFINED

...

Conformance Rules

Without Feature T323, "Explicit security for external routines", conforming SQL language
shall not contain an <external security clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.224 T324, Explicit security for SQL routines

V0260:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<SQL routine spec> ::=
[ <rights clause> ] <SQL routine body>

<rights clause> ::=
SQL SECURITY INVOKER
| SQL SECURITY DEFINER

...

Conformance Rules

Without Feature T324, "Explicit security for SQL routines", conforming SQL language shall not
contain a <rights clause>.
```

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

Transact-SQL does not support this feature.

2.1.2.225 T325, Qualified SQL parameter references

V0261:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<qualified identifier> ::=
<identifier>

Subclause 10.4, "<routine invocation>":
```

```
<routine name> ::=
[ <schema name> <period> ] <qualified identifier>

Subclause 6.6, "<identifier chain>":

<identifier chain> ::=
<identifier> [ { <period> <identifier> }... ]

<basic identifier chain> ::=
<identifier chain>

...

Conformance Rules

Without Feature T325, "Qualified SQL parameter references", conforming SQL language shall not
contain an SQL parameter reference whose first <identifier> is the <qualified identifier> of
a <routine name>.

Subclause 7.12, "<query specification>":

<asterisked identifier chain> ::=
<asterisked identifier> [ { <period> <asterisked identifier> }... ]

<asterisked identifier> ::=
<identifier>

...

Conformance Rules

Without Feature T325, "Qualified SQL parameter references", conforming SQL language shall not
contain an <asterisked identifier chain> whose referent is an SQL parameter and whose first
<identifier> is the <qualified identifier> of a <routine name>.
```

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

Transact-SQL does not support this feature.

2.1.2.226 T326, Table functions

V0262:

The specification states the following:

```
Subclause 6.41, "<multiset value constructor>":

<multiset value constructor> ::=
<multiset value constructor by enumeration>
| <multiset value constructor by query>
| <table value constructor by query>

...

<table value constructor by query> ::=
TABLE <table subquery>
```

```
...

Conformance Rules

Without Feature T326, "Table functions", a <multiset value constructor> shall not contain a
<table value constructor by query>.

Subclause 7.6, "<table reference>":

<table function derived table> ::=
TABLE <left paren> <collection value expression> <right paren>

...

Conformance Rules

Without Feature T326, "Table functions", conforming SQL language shall not contain a <table
function derived table>.

Subclause 11.60, "<SQL-invoked routine>":

<returns table type> ::=
TABLE <table function column list>

...

Conformance Rules

Without Feature T326, "Table functions", conforming SQL language shall not contain a <returns
table type>.
```

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

Transact-SQL does not support this feature.

2.1.2.227 T331, Basic roles

V0263:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<role name> ::=
<identifier>

...

Conformance Rules

Without Feature T331, "Basic roles", conforming SQL language shall not contain a <role name>.

Subclause 12.4, "<role definition>":

<role definition> ::=
CREATE ROLE <role name> [ WITH ADMIN <grantor> ]
```

```
...

Conformance Rules

Without Feature T331, "Basic roles", conforming SQL language shall not contain a <role
definition>.

Subclause 12.5, "<grant role statement>":

<grant role statement> ::=
GRANT <role granted> [ { <comma> <role granted> }... ]
TO <grantee> [ { <comma> <grantee> }... ]
[ WITH ADMIN OPTION ]
[ GRANTED BY <grantor> ]

...

Conformance Rules

Without Feature T331, "Basic roles", conforming SQL language shall not contain a <grant role
statement>.

Subclause 12.6, "<drop role statement>":

<drop role statement> ::=
DROP ROLE <role name>

...

Conformance Rules

Without Feature T331, "Basic roles", conforming SQL language shall not contain a <drop role
statement>.

Subclause 12.7, "<revoke statement>":

<revoke role statement> ::=
REVOKE [ ADMIN OPTION FOR ] <role revoked> [ { <comma> <role revoked> }... ]
FROM <grantee> [ { <comma> <grantee> }... ]
[ GRANTED BY <grantor> ]
<drop behavior>

...

Conformance Rules

Without Feature T331, "Basic roles", conforming SQL language shall not contain a <revoke role
statement>.

Subclause 19.3, "<set role statement>":

<set role statement> ::=
SET ROLE <role specification>

<role specification> ::=
<value specification>
| NONE

...
```

```
Conformance Rules

Without Feature T331, "Basic roles", conforming SQL language shall not contain a <set role
statement>.
```

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

Transact-SQL partially supports this feature. The **CREATE ROLE** statement is supported, with syntax that varies from the standard. Dropping roles is supported with the standard syntax. The **SET ROLE** statement is not supported. Granting and revoking roles are not supported. See the system stored procedure **sp_addrolemember** [MSDN-spaddrolemember] for some equivalent functionality.

2.1.2.228 T332, Extended roles

V0264:

The specification states the following:

```
Subclause 6.4, "<value specification> and <target specification>":

<general value specification> ::=
<host parameter specification>
| <SQL parameter reference>
| <dynamic parameter specification>
| <embedded variable specification>
| <current collation specification>
| CURRENT_CATALOG
| CURRENT_DEFAULT_TRANSFORM_GROUP
| CURRENT_PATH
| CURRENT_ROLE
| CURRENT_SCHEMA
| CURRENT_TRANSFORM_GROUP_FOR_TYPE <path-resolved user-defined type name>
| CURRENT_USER
| SESSION_USER
| SYSTEM_USER
| USER
| VALUE

...

Conformance Rules

Without Feature T332, "Extended roles", conforming SQL language shall not contain
CURRENT_ROLE.

Subclause 5.4, "Names and identifiers":

<authorization identifier> ::=
<role name>
| <user identifier>

...

<role name> ::=
```

```
<identifier>

Subclause 11.1, "<schema definition>":

<schema authorization identifier> ::=
<authorization identifier>

...

Conformance Rules

Without Feature T332, "Extended roles", in conforming SQL language a <schema authorization
identifier> shall not be a <role name>.

Subclause 11.5, "<default clause>":

<default option> ::=
<literal>
| <datetime value function>
| USER
| CURRENT_USER
| CURRENT_ROLE
| SESSION_USER
| SYSTEM_USER
| CURRENT_CATALOG
| CURRENT_SCHEMA
| CURRENT_PATH
| <implicitly typed value specification>

...

Conformance Rules

Without Feature T332, "Extended roles", conforming SQL language shall not contain a <default
option> that contains CURRENT_ROLE.

Subclause 12.3, "<privileges>":

<grantor> ::=
CURRENT_USER
| CURRENT_ROLE

...

Conformance Rules

Without Feature T332, "Extended roles", conforming SQL language shall not contain a
<grantor>.

Subclause 12.4, "<role definition>":

<role definition> ::=
CREATE ROLE <role name> [ WITH ADMIN <grantor> ]

...

Conformance Rules
```

```
Without Feature T332, "Extended roles", conforming SQL language shall not contain a <role
definition> that immediately contains WITH ADMIN.

Subclause 12.8, "Grantor determination":

Conformance Rules

Without Feature T332, "Extended roles", conforming SQL language shall contain no <grantor>.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **CURRENT_ROLE** role.

2.1.2.229 T341, Overloading of SQL-invoked functions and SQL-invoked procedures

V0265:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<SQL-invoked routine> ::=
<schema routine>

<schema routine> ::=
<schema procedure>
| <schema function>

<schema procedure> ::=
CREATE <SQL-invoked procedure>

<schema function> ::=
CREATE <SQL-invoked function>

<SQL-invoked procedure> ::=
PROCEDURE <schema qualified routine name> <SQL parameter declaration list>
<routine characteristics>
<routine body>

<SQL-invoked function> ::=
{ <function specification> | <method specification designator> } <routine body>

...

<function specification> ::=
FUNCTION <schema qualified routine name> <SQL parameter declaration list>
<returns clause>
<routine characteristics>
[ <dispatch clause> ]

...

Conformance Rules
```

```
Without Feature T341, "Overloading of SQL-invoked functions and procedures", conforming SQL
language shall not contain a <schema routine> in which the schema identified by the explicit
or implicit schema name of the <schema qualified routine name> includes a routine descriptor
whose routine name is <schema qualified routine name>.
```

The [ISO/IEC9075-2:2008] specification differs as follows:

This feature was labeled as feature T322.

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

Transact-SQL does not support this feature.

2.1.2.230 T433, Multiargument GROUPING function

V0266:

The specification states the following:

```
Subclause 6.9, "<set function specification>":

<set function specification> ::=
<aggregate function>
| <grouping operation>

<grouping operation> ::=
GROUPING <left paren> <column reference>
[ { <comma> <column reference> }... ] <right paren>

...

Conformance Rules

Without Feature T433, "Multiargument GROUPING function", conforming SQL language shall not
contain a <grouping operation> that contains more than one <column reference>.
```

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

Transact-SQL does not support this feature.

2.1.2.231 T434, GROUP BY DISTINCT

V0267:

The specification states the following:

```
Subclause 7.4, "<table expression>":

<table expression> ::=
<from clause>
[ <where clause> ]
[ <group by clause> ]
[ <having clause> ]
[ <window clause> ]

Subclause 10.9, "<aggregate function>":
```

```
<set quantifier> ::=
DISTINCT
| ALL

Subclause 7.9, "<group by clause>":

<group by clause> ::=
GROUP BY [ <set quantifier> ] <grouping element list>

<grouping element list> ::=
<grouping element> [ { <comma> <grouping element> }... ]

...

Conformance Rules

Without Feature T434, "GROUP BY DISTINCT", conforming SQL language shall not contain a <group
by clause> that simply contains a <set quantifier>.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **DISTINCT** keyword, but SQL Server provides the **DISTINCT** functionality implicitly by default.

2.1.2.232 T441, ABS and MOD functions

V0268:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<absolute value expression> ::=
ABS <left paren> <numeric value expression> <right paren>

<modulus expression> ::=
MOD <left paren> <numeric value expression dividend> <comma>
<numeric value expression divisor> <right paren>

...

Conformance Rules

Without Feature T441, "ABS and MOD functions", conforming language shall not contain an
<absolute value expression>.

Without Feature T441, "ABS and MOD functions", conforming language shall not contain a
<modulus expression>.
```

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

Transact-SQL partially supports this feature. The **ABS** function is supported. For equivalent functionality to the **MOD** function, see the **%** operator [MSDN-Modulo].

2.1.2.233 T461, Symmetric BETWEEN predicate

V0269:

The specification states the following:

```
Subclause 8.3, "<between predicate>":

<between predicate> ::=
<row value predicand> <between predicate part 2>

<between predicate part 2> ::=
[ NOT ] BETWEEN [ ASYMMETRIC | SYMMETRIC ]
<row value predicand> AND <row value predicand>

...

Conformance Rules

Without Feature T461, "Symmetric BETWEEN predicate", conforming SQL language shall not
contain SYMMETRIC or ASYMMETRIC.

NOTE — Since <between predicate> is an ordering operation, the Conformance Rules of Subclause
9.14, "Ordering operations", also apply.

Subclause 9.14, "Ordering operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an ordering operation shall not be ST-ordered.
```

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

Transact-SQL does not support this feature.

2.1.2.234 T471, Result sets return value

V0270:

The specification states the following:

```
Subclause 11.60, "<SQL-invoked routine>":

<returned result sets characteristic> ::=
DYNAMIC RESULT SETS <maximum returned result sets>

...

Conformance Rules

Without Feature T471, "Result sets return value", conforming SQL language shall not contain a
<returned result sets characteristic>.

Subclause 14.2, "<cursor properties>":

<cursor properties> ::=
[ <cursor sensitivity> ] [ <cursor scrollability> ] CURSOR
```

```
[ <cursor holdability> ]
[ <cursor returnability> ]

...

<cursor returnability> ::=
WITH RETURN
| WITHOUT RETURN

...

Conformance Rules

Without Feature T471, "Result sets return value", conforming SQL language shall not contain a
<cursor returnability>.

Subclause 20.16, "<allocate cursor statement>":

<allocate received cursor statement> ::=
ALLOCATE <cursor name>
[ CURSOR ] FOR PROCEDURE <specific routine designator>

Conformance Rules

Without Feature T471, "Result sets return value", conforming SQL language shall not contain
an <allocate received cursor statement>.
```

The [ISO/IEC9075-2:2008] specification differs as follows:

```
Subclause 20.15, "<allocate cursor statement>":

<cursor intent> ::=
<statement cursor>
| <result set cursor>

...

<result set cursor> ::=
[ CURSOR ] FOR PROCEDURE <specific routine designator>

...

Conformance Rules

Without Feature T471, "Result sets return value", conforming SQL language shall not contain a
<result set cursor>.
```

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

Transact-SQL does not support this feature.

2.1.2.235 T472, DESCRIBE CURSOR

V0271:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<cursor name> ::=
<local qualified name>

Subclause 20.9, "<describe statement>":

<describe output statement> ::=
DESCRIBE [ OUTPUT ] <described object> <using descriptor> [ <nesting option> ]

<nesting option> ::=
WITH NESTING
| WITHOUT NESTING

<using descriptor> ::=
USING [ SQL ] DESCRIPTOR <descriptor name>

<described object> ::=
<SQL statement name>
| CURSOR <cursor name> STRUCTURE

...

Conformance Rules:

Without Feature T472, "DESCRIBE CURSOR", conforming SQL language shall not contain a
<describe output statement> that contains a <described object> that contains a <cursor name>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.236 T491, LATERAL derived table

V0272:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<table primary> ::=
<table or query name> [ [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ] ]
| <derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <lateral derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <collection derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <table function derived table> [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ]
| <only spec> [ [ AS ] <correlation name>
[ <left paren> <derived column list> <right paren> ] ]
```

```
| <parenthesized joined table>

...

<lateral derived table> ::=
LATERAL <table subquery>

...

Conformance Rules

Without Feature T491, "LATERAL derived table", conforming SQL language shall not contain a
<lateral derived table>.
```

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

Transact-SQL does not support this feature. See the **APPLY** operator [MSDN-APPLY] for equivalent functionality.

2.1.2.237 T495, Combined data change and retrieval

V0273:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<data change delta table> ::=
<result option> TABLE <left paren> <data change statement> <right paren>

...

Conformance Rules

Without Feature T495, "Combined data change and retrieval", conforming SQL language shall not
contain <data change delta table>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.238 T502, Period predicates

V0274:

The specification states the following:

```
Subclause 8.20, "<period predicate>":

<period predicate> ::=
<period overlaps predicate>
```

```
| <period equals predicate>
| <period contains predicate>
| <period precedes predicate>
| <period succeeds predicate>
| <period immediately precedes predicate>
| <period immediately succeeds predicate>

...

Conformance Rules:

Without Feature T502, "Period predicates", conforming SQL language shall not contain a
<period predicate>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.239 T511, Transaction counts

V0275:

The specification states the following:

```
Subclause 23.1, "<get diagnostics statement>":

<statement information> ::=
<statement information item> [ { <comma> <statement information item> }... ]

<statement information item> ::=
<simple target specification> <equals operator> <statement information item name>

<statement information item name> ::=
NUMBER
| MORE
| COMMAND_FUNCTION
| COMMAND_FUNCTION_CODE
| DYNAMIC_FUNCTION
| DYNAMIC_FUNCTION_CODE
| ROW_COUNT
| TRANSACTIONS_COMMITTED
| TRANSACTIONS_ROLLED_BACK
| TRANSACTION_ACTIVE

...

Conformance Rules

Without Feature T511, "Transaction counts", conforming SQL language shall not contain a
<statement information item name> that contains TRANSACTIONS_COMMITTED,
TRANSACTIONS_ROLLED_BACK, or TRANSACTION_ACTIVE.
```

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

Transact-SQL does not support this feature.

2.1.2.240 T521, Named arguments in CALL statement

V0276:

The specification states the following:

```
Subclause 10.4, "<routine invocation>":

<named argument specification> ::=
<SQL parameter name> <named argument assignment token>
<named argument SQL argument>

...

Conformance Rules:

Without Feature T521, "Named arguments in CALL statement", conforming SQL language shall not
contain a <named argument specification>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.241 T522, Default values for IN parameters of SQL-invoked procedures

V0277:

The specification states the following:

```
Subclause 6.5, "<contextually typed value specification>":

<contextually typed value specification> ::=
<implicitly typed value specification>
| <default specification>

Subclause 10.4, "<routine invocation>":

<SQL argument> ::=
<value expression>
| <generalized expression>
| <target specification>
| <contextually typed value specification>
| <named argument specification>

...

Conformance Rules:

Without Feature T522, "Default values for IN parameters of SQL-invoked procedures", in
conforming SQL language an <SQL argument> shall not be a <contextually typed value
specification>.
```

```
Subclause 11.60, "<SQL-invoked routine>":

<parameter default> ::=
<value expression>
| <contextually typed value specification>

...

Conformance Rules:

Without Feature T522, "Default values for IN parameters of SQL-invoked procedures",
conforming SQL language shall not contain a <parameter default>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL partially supports this feature. The syntax differs from the standard.

2.1.2.242 T551, Optional key words for default syntax

V0278:

The specification states the following:

```
Subclause 7.13, "<query expression>":

<query expression body> ::=
<query term>
| <query expression body> UNION [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>
| <query expression body> EXCEPT [ ALL | DISTINCT ]
[ <corresponding spec> ] <query term>

<query term> ::=
<query primary>
| <query term> INTERSECT [ ALL | DISTINCT ]
[ <corresponding spec> ] <query primary>

...

Conformance Rules

Without Feature T551, "Optional key words for default syntax", conforming SQL language shall
not contain UNION DISTINCT, EXCEPT DISTINCT, or INTERSECT DISTINCT.

Subclause 14.2, "<cursor properties>":

<cursor properties> ::=
[ <cursor sensitivity> ] [ <cursor scrollability> ] CURSOR
[ <cursor holdability> ]
[ <cursor returnability> ]

...
```

```
<cursor holdability> ::=
WITH HOLD
| WITHOUT HOLD

...

Conformance Rules

Without Feature T551, "Optional key words for default syntax", conforming SQL language shall
not contain a <cursor holdability> that immediately contains WITHOUT HOLD.
```

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

Transact-SQL does not support this feature. See the **EXCEPT**, **INTERSECT** [MSDN-EXCEPTINTERSECT], and **UNION** [MSDN-UNION] operators without the **DISTINCT** keyword for equivalent functionality.

2.1.2.243 T561, Holdable locators

V0279:

The specification states the following:

```
Subclause 14.17, "<free locator statement>":

<free locator statement> ::=
FREE LOCATOR <locator reference> [ { <comma> <locator reference> }... ]

<locator reference> ::=
<host parameter name>
| <embedded variable name>
| <dynamic parameter specification>

...

Conformance Rules

Without Feature T561, "Holdable locators", conforming SQL language shall not contain a <free
locator statement>.

Subclause 14.18, "<hold locator statement>":

<hold locator statement> ::=
HOLD LOCATOR <locator reference> [ { <comma> <locator reference> }... ]

...

Conformance Rules

Without Feature T561, "Holdable locators", conforming SQL language shall not contain a <hold
locator statement>.
```

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

Transact-SQL does not support this feature.

2.1.2.244 T571, Array-returning external SQL-invoked functions

V0280:

The specification states the following:

```
Subclause 6.1, "<data type>":

<array type> ::=
<data type> ARRAY
[ <left bracket or trigraph> <maximum cardinality> <right bracket or trigraph> ]

Subclause 11.51, "<user-defined type definition>":

<method specification> ::=
<original method specification>
| <overriding method specification>

<original method specification> ::=
<partial method specification> [ SELF AS RESULT ] [ SELF AS LOCATOR ]
[ <method characteristics> ]

<overriding method specification> ::=
OVERRIDING <partial method specification>

<partial method specification> ::=
[ INSTANCE | STATIC | CONSTRUCTOR ]
METHOD <method name> <SQL parameter declaration list>
<returns clause>
[ SPECIFIC <specific method name> ]

...

Conformance Rules

Without Feature T571, "Array-returning external SQL-invoked functions", conforming SQL
language shall not contain a <method specification> that contains a <returns clause> that
satisfies either of the following conditions:
a) A <result cast from type> is specified that simply contains an <array type> and does not
contain a <locator indication>.
b) A <result cast from type> is not specified and <returns data type> simply contains an
<array type> and does not contain a <locator indication>.

Subclause 11.60, "<SQL-invoked routine>":

<SQL-invoked routine> ::=
<schema routine>

<schema routine> ::=
<schema procedure>
| <schema function>

<schema procedure> ::=
CREATE <SQL-invoked procedure>

<schema function> ::=
CREATE <SQL-invoked function>

...
```

```
<locator indication> ::=
AS LOCATOR

...

<returns clause> ::=
RETURNS <returns type>

<returns type> ::=
<returns data type> [ <result cast> ]
| <returns table type>

...

<result cast from type> ::=
<data type> [ <locator indication> ]

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules

Without Feature T571, "Array-returning external SQL-invoked functions", conforming SQL
language shall not contain an <SQL-invoked routine> that defines an array-returning external
function.
```

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

Transact-SQL does not support this feature.

2.1.2.245 T572, Multiset-returning external SQL-invoked functions

V0281:

The specification states the following:

```
Subclause 6.1, "<data type>":

<multiset type> ::=
<data type> MULTISET

Subclause 11.51, "<user-defined type definition>":

<method specification> ::=
<original method specification>
| <overriding method specification>

<original method specification> ::=
<partial method specification> [ SELF AS RESULT ] [ SELF AS LOCATOR ]
[ <method characteristics> ]

<overriding method specification> ::=
OVERRIDING <partial method specification>
```

```
<partial method specification> ::=
[ INSTANCE | STATIC | CONSTRUCTOR ]
METHOD <method name> <SQL parameter declaration list>
<returns clause>
[ SPECIFIC <specific method name> ]

...

Conformance Rules

Without Feature T572, "Multiset-returning external SQL-invoked functions", conforming SQL
language shall not contain a <method specification> that contains a <returns clause> that
satisfies either of the following conditions:
a) A <result cast from type> is specified that simply contains a <multiset type> and does not
contain a <locator indication>.
b) A <result cast from type> is not specified and <returns data type> simply contains a
<multiset type> and does not contain a <locator indication>.

Subclause 11.60, "<SQL-invoked routine>":

<SQL-invoked routine> ::=
<schema routine>

<schema routine> ::=
<schema procedure>
| <schema function>

<schema procedure> ::=
CREATE <SQL-invoked procedure>

<schema function> ::=
CREATE <SQL-invoked function>

...

<locator indication> ::=
AS LOCATOR

...

<returns clause> ::=
RETURNS <returns type>

<returns type> ::=
<returns data type> [ <result cast> ]
| <returns table type>

...

<result cast from type> ::=
<data type> [ <locator indication> ]

<returns data type> ::=
<data type> [ <locator indication> ]

...

Conformance Rules
```

```
Without Feature T572, "Multiset-returning external SQL-invoked functions", conforming SQL
language shall not contain an <SQL-invoked routine> that defines a multiset-returning
external function.
```

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

Transact-SQL does not support this feature.

2.1.2.246 T581, Regular expression substring function

V0282:

The specification states the following:

```
Subclause 6.30, "<string value function>":

<regular expression substring function> ::=
SUBSTRING <left paren> <character value expression> SIMILAR <character value expression>
ESCAPE <escape character> <right paren>

...

Conformance Rules

Without Feature T581, "Regular expression substring function", conforming SQL language shall
not contain a <regular expression substring function>.
```

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

Transact-SQL does not support this feature.

2.1.2.247 T601, Local cursor references

V0283:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<cursor name> ::=
<local qualified name>

<local qualified name> ::=
[ <local qualifier> <period> ] <qualified identifier>

<local qualifier> ::=
MODULE

...

Conformance Rules

Without Feature T601, "Local cursor references", a <cursor name> shall not contain a <local
qualifier>.
```

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

Transact-SQL does not support this feature.

2.1.2.248 T611, Elementary OLAP operations

V0284:

The specification states the following:

```
Subclause 6.10, "<window function>":

<window function> ::=
<window function type> OVER <window name or specification>

<window function type> ::=
<rank function type> <left paren> <right paren>
| ROW_NUMBER <left paren> <right paren>
| <aggregate function>

<rank function type> ::=
RANK
| DENSE_RANK
| PERCENT_RANK
| CUME_DIST

<window name or specification> ::=
<window name>
| <in-line window specification>

<in-line window specification> ::=
<window specification>

Conformance Rules

1) Without Feature T611, "Elementary OLAP operations", conforming SQL language shall not
contain a <window function>.

Subclause 7.11, "<window clause>":

<window specification> ::=
<left paren> <window specification details> <right paren>

<window specification details> ::=
[ <existing window name> ]
[ <window partition clause> ]
[ <window order clause> ]
[ <window frame clause> ]

...

Conformance Rules

1) Without Feature T611, "Elementary OLAP operations", conforming SQL language shall not
contain a <window specification>.

Subclause 10.10, "<sort specification list>":

<sort specification> ::=
```

```
<sort key> [ <ordering specification> ] [ <null ordering> ]

...

<null ordering> ::=
NULLS FIRST
| NULLS LAST

...

Conformance Rules

1) Without Feature T611, "Elementary OLAP operations", conforming SQL language shall not
contain a <null ordering>.

NOTE — The Conformance Rules of Subclause 9.14, "Ordering operations", also apply.

Subclause 9.14, "Ordering operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of an ordering operation shall not be ST-ordered.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support null ordering or all aggregate functions.

2.1.2.249 T612, Advanced OLAP operations

V0285:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<window name> ::=
<identifier>

...

Conformance Rules

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<window name>.

Subclause 6.10, "<window function>":

<window function> ::=
<window function type> OVER <window name or specification>

<window function type> ::=
<rank function type> <left paren> <right paren>
| ROW_NUMBER <left paren> <right paren>
| <aggregate function>
```

```
<rank function type> ::=
RANK
| DENSE_RANK
| PERCENT_RANK
| CUME_DIST

<window name or specification> ::=
<window name>
| <in-line window specification>

<in-line window specification> ::=
<window specification>

...

Conformance Rules

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<window name>.

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain
PERCENT_RANK or CUME_DIST.

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<window function> that simply contains ROW_NUMBER and immediately contains a <window name or
specification> whose window structure descriptor does not contain a window ordering clause.

Subclause 6.28, "<numeric value function>":

<width bucket function> ::=
WIDTH_BUCKET <left paren> <width bucket operand> <comma> <width bucket bound 1> <comma>
<width bucket bound 2> <comma> <width bucket count> <right paren>

...

Conformance Rules

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<width bucket function>.

Subclause 7.11, "<window clause>":

<window clause> ::=
WINDOW <window definition list>

...

<existing window name> ::=
<window name>

...

<window frame exclusion> ::=
EXCLUDE CURRENT ROW
| EXCLUDE GROUP
| EXCLUDE TIES
| EXCLUDE NO OTHERS

...
```

```
Conformance Rules

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<window clause>.

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain
an <existing window name>.

...

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<window frame exclusion>.

NOTE — The Conformance Rules of Subclause 9.12, "Grouping operations", also apply.

Subclause 9.12, "Grouping operations":

Conformance Rules

Without Feature S024, "Enhanced structured types", in conforming SQL language, the declared
type of an operand of a grouping operation shall not be ST-ordered.

Subclause 10.9, "<aggregate function>":

<filter clause> ::=
FILTER <left paren> WHERE <search condition> <right paren>

...

<hypothetical set function> ::=
<rank function type> <left paren>
<hypothetical set function value expression list> <right paren>
<within group specification>

...

<inverse distribution function> ::=
<inverse distribution function type> <left paren>
<inverse distribution function argument> <right paren>
<within group specification>

...

Conformance Rules

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<hypothetical set function> or an <inverse distribution function>.

Without Feature T612, "Advanced OLAP operations", conforming SQL language shall not contain a
<filter clause>.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **WIDTH_BUCKET**, **PERCENT_RANK**, and **CUME_DIST** functions or the **WINDOW** and **FILTER** clauses.

2.1.2.250 T613, Sampling

V0286:

The specification states the following:

```
Subclause 7.6, "<table reference>":

<table reference> ::=
<table factor>
| <joined table>

<table factor> ::=
<table primary> [ <sample clause> ]

<sample clause> ::=
TABLESAMPLE <sample method> <left paren> <sample percentage> <right paren>
[ <repeatable clause> ]

...

Conformance Rules

Without Feature T613, "Sampling", conforming SQL language shall not contain a <sample
clause>.
```

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

Transact-SQL partially supports this feature. The **TABLESAMPLE** clause does not support the **BERNOULLI** keyword, and the syntax of the clause differs slightly from the standard.

2.1.2.251 T614, NTILE function

V0287:

The specification states the following:

```
Subclause 6.10, "<window function>":

<ntile function> ::=
NTILE <left paren> <number of tiles> <right paren>

...

Conformance Rules

Without Feature T614, "NTILE function", conforming SQL language shall not contain <ntile
function>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL partially supports this feature. The parameters and exact functionality of the Transact-SQL **NTILE** function differ.

2.1.2.252 T616, Null treatment option for LEAD and LAG functions

V0288:

The specification states the following:

```
Subclause 6.10, "<window function>":

<lead or lag function> ::=
<lead or lag> <left paren> <lead or lag extent>
[ <comma> <offset> [ <comma> <default expression> ] ] <right paren>
[ <null treatment> ]

<lead or lag> ::=
LEAD | LAG

<lead or lag extent> ::=
<value expression>

...

<null treatment> ::=
RESPECT NULLS | IGNORE NULLS

...

Conformance Rules:

Without Feature T616, "Null treatment option for LEAD and LAG functions", in conforming SQL
language, <lead or lag function> shall not contain <null treatment>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.253 T618, NTH_VALUE function

V0289:

The specification states the following:

```
Subclause 6.10, "<window function>":

<nth value function> ::=
NTH_VALUE <left paren> <value expression> <comma> <nth row> <right paren>
[ <from first or last> ] [ <null treatment> ]

...

Conformance Rules
```

```
Without Feature T618, "NTH_VALUE function", conforming SQL language shall not contain <nth
value function>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.254 T619, Nested window functions

V0290:

The specification states the following:

```
Subclause 6.11, "<nested window function>":

<nested window function> ::=
<nested row number function>
| <value_of expression at row>

<nested row number function> ::=
ROW_NUMBER <left paren> <row marker> <right paren>

<value_of expression at row> ::=
VALUE_OF <left paren> <value expression> AT <row marker expression>
[ <comma> <value_of default value> ] <right paren>

...

Conformance Rules:

Without Feature T619, "Nested window functions", conforming SQL language shall not contain
<nested window function>.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.255 T620, WINDOW clause: GROUPS option

V0291:

The specification states the following:

```
Subclause 7.11, "<window clause>":

<window frame units> ::=
ROWS
```

```
| RANGE
| GROUPS

...

Conformance Rules:

Without Feature T620, "WINDOW clause: GROUPS option", conforming SQL language shall not
contain <window frame units> that specifies GROUPS.
```

Microsoft SQL Server 2008 R2 varies as follows:

This feature is absent from the [ISO/IEC9075-2:2008] standard.

Microsoft SQL Server 2012 varies as follows:

Transact-SQL does not support this feature.

2.1.2.256 T621, Enhanced numeric functions

V0292:

The specification states the following:

```
Subclause 6.28, "<numeric value function>":

<natural logarithm> ::=
LN <left paren> <numeric value expression> <right paren>

<exponential function> ::=
EXP <left paren> <numeric value expression> <right paren>

<power function> ::=
POWER <left paren> <numeric value expression base> <comma>
<numeric value expression exponent> <right paren>

...

<square root> ::=
SQRT <left paren> <numeric value expression> <right paren>

<floor function> ::=
FLOOR <left paren> <numeric value expression> <right paren>

<ceiling function> ::=
{ CEIL | CEILING } <left paren> <numeric value expression> <right paren>

...

Conformance Rules

Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
a <natural logarithm>.

Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
an <exponential function>.
```

```
Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
a <power function>.

Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
a <square root>.

Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
a <floor function>.

Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
a <ceiling function>.

Subclause 10.9, "<aggregate function>":

<computational operation> ::=
AVG
| MAX
| MIN
| SUM
| EVERY
| ANY
| SOME
| COUNT
| STDDEV_POP
| STDDEV_SAMP
| VAR_SAMP
| VAR_POP
| COLLECT
| FUSION
| INTERSECTION

...

<binary set function type> ::=
COVAR_POP
| COVAR_SAMP
| CORR
| REGR_SLOPE
| REGR_INTERCEPT
| REGR_COUNT
| REGR_R2
| REGR_AVGX
| REGR_AVGY
| REGR_SXX
| REGR_SYY
| REGR_SXY

...

Conformance Rules

Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
a <computational operation> that immediately contains STDDEV_POP, STDDEV_SAMP, VAR_POP, or
VAR_SAMP.

Without Feature T621, "Enhanced numeric functions", conforming SQL language shall not contain
a <binary set function type>.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support the **LN**, **CEIL**, **STDDEV_POP**, **STDDEV_SAMP**, **VAR_POP**, or **VAR_SAMP** keywords. See the **LOG** [MSDN-LOG], **CEILING** [MSDN-CEILING], **STDEV** [MSDN-STDEV], **STDEVP** [MSDN-STDEVP], **VARP** [MSDN-VARP], and **VAR** [MSDN-VAR] functions for equivalent functionality. Transact-SQL does not support binary set functions, such as **COVAR_POP**.

2.1.2.257 T641, Multiple column assignment

V0293:

The specification states the following:

```
Subclause 14.15, "<set clause list>":

<set clause list> ::=
<set clause> [ { <comma> <set clause> }... ]

<set clause> ::=
<multiple column assignment>
| <set target> <equals operator> <update source>

<set target> ::=
<update target>
| <mutated set clause>

<multiple column assignment> ::=
<set target list> <equals operator> <assigned row>

...

Conformance Rules

Without Feature T641, "Multiple column assignment", conforming SQL language shall not contain
a <multiple column assignment>.
```

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

Transact-SQL partially supports this feature. Transact-SQL does not support multiple columns on both sides of an assignment.

See [ISO/IEC9075-2:2011] for the definition of "column".

2.1.2.258 T652, SQL-dynamic statements in SQL routines

V0294:

The specification states the following:

```
Subclause 4.34.2.9, "SQL-dynamic statements":

The following are the SQL-dynamic statements:
— <execute immediate statement>.
— <allocate descriptor statement>.
— <deallocate descriptor statement>.
— <get descriptor statement>.
— <set descriptor statement>.
— <prepare statement>.
```

```
— <deallocate prepared statement>.
— <describe input statement>.
— <describe output statement>.
— <execute statement>.

Subclause 11.60, "<SQL-invoked routine>":

<SQL routine body> ::=
<SQL procedure statement>

...

Conformance Rules

Without Feature T652, "SQL-dynamic statements in SQL routines", conforming SQL language shall
not contain an <SQL routine body> that contains an SQL-dynamic statement.

Subclause 13.4, "<SQL procedure statement>":

<SQL procedure statement> ::=
<SQL executable statement>

<SQL executable statement> ::=
<SQL schema statement>
| <SQL data statement>
| <SQL control statement>
| <SQL transaction statement>
| <SQL connection statement>
| <SQL session statement>
| <SQL diagnostics statement>
| <SQL dynamic statement>
```

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

Transact-SQL partially supports this feature. Transact-SQL supports the **EXECUTE** and **EXECUTE IMMEDIATE** statements.

2.1.2.259 T654, SQL-dynamic statements in external routines

V0295:

The specification states the following:

```
Subclause 5.4, "Names and identifiers":

<external routine name> ::=
<identifier>
| <character string literal>

Subclause 11.60, "<SQL-invoked routine>":

<external body reference> ::=
EXTERNAL [ NAME <external routine name> ]
[ <parameter style clause> ]
[ <transform group specification> ]
[ <external security clause> ]
```

```
Conformance Rules

Without Feature T654, "SQL-dynamic statements in external routines", conforming SQL language
shall not contain an <external routine name> that identifies a program in which an SQL-
dynamic statement appears.
```

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

Transact-SQL does not support this feature.

2.2 Clarifications

Unless otherwise stated, the specified products conform to all SHOULD and RECOMMENDED behavior in [ISO/IEC9075-2:2008] and [ISO/IEC9075-2:2011].

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