

Information technology — Database languages — SQL —

Part 4: Persistent Stored Modules (SQL/PSM)

ICS 35.060

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI
Documents



Document Sharing Hub

National foreword

This British Standard reproduces verbatim ISO/IEC 9075-4:1999 and implements it as the UK national standard. It supersedes BS ISO/IEC 9075-4:1996 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/40, Data management and interchange, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the ISO/IEC title page, pages i-2 to i-3, pages ii to x, pages 1 to 152, an inside back cover and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

This British Standard, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Committee and comes into effect on 15 May 2000

© BSI 05-2000

ISBN 0 580 34631 5

Amendments issued since publication

Amd. No.	Date	Comments

**Information technology — Database
languages — SQL —**

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

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

Reference number
ISO/IEC 9075-4:1999(E)



Contents

Page

Foreword	vii
Introduction	ix
1 Scope	1
2 Normative references	3
3 Definitions, notations, and conventions	5
3.1 Definitions	5
3.1.1 Definitions provided in Part 4	5
3.2 Notations	5
3.3 Conventions	5
3.3.1 Use of terms	5
3.3.1.1 Exceptions	5
3.3.1.2 Other terms	6
3.3.2 Relationships to other parts of ISO/IEC 9075	6
3.3.2.1 Clause, Subclause, and Table relationships	6
3.4 Object identifier for Database Language SQL	12
4 Concepts	13
4.1 SQL-server modules	13
4.2 SQL-invoked routines	14
4.3 SQL-paths	14
4.4 Tables	14
4.5 SQL-schemas	14
4.6 Host parameters	15
4.6.1 Status parameters	15
4.7 Diagnostics area	15
4.8 Cursors	15
4.9 Condition handling	15
4.10 SQL-statements	17
4.10.1 SQL-statements classified by function	17
4.10.2 Embeddable SQL-statements	18
4.10.3 Directly executable SQL-statements	18
4.10.4 Iterated SQL-statements	18
4.10.5 SQL-statements and transaction states	18

4.10.6	Compound statements	19
4.10.7	SQL-statement atomicity	19
4.11	Basic security model	19
4.11.1	Privileges	19
5	Lexical elements	21
5.1	<token> and <separator>	21
5.2	Names and identifiers	23
6	Scalar expressions	27
6.1	<value specification> and <target specification>	27
6.2	<identifier chain>	28
6.3	<SQL variable reference>	30
7	Query expressions	31
7.1	<query specification>	31
8	Additional common elements	33
8.1	<routine invocation>	33
8.2	<privileges>	35
8.3	<sqlstate value>	36
9	Schema definition and manipulation	37
9.1	<schema definition>	37
9.2	<drop schema statement>	38
9.3	<default clause>	39
9.4	<drop column scope clause>	40
9.5	<drop column definition>	41
9.6	<drop table constraint definition>	43
9.7	<drop table statement>	44
9.8	<view definition>	45
9.9	<drop view statement>	46
9.10	<drop domain statement>	47
9.11	<drop character set statement>	48
9.12	<drop collation statement>	49
9.13	<drop translation statement>	50
9.14	<assertion definition>	51
9.15	<drop assertion statement>	52
9.16	<trigger definition>	53
9.17	<drop user-defined ordering statement>	54
9.18	<SQL-server module definition>	55
9.19	<drop module statement>	58
9.20	<drop data type statement>	59
9.21	<SQL-invoked routine>	60
9.22	<drop routine statement>	62
9.23	<drop user-defined cast statement>	63

ISO/IEC 9075-4:1999 (E)

10 Access control	65
10.1 <grant statement>	65
10.2 <revoke statement>	66
11 SQL-client modules	69
11.1 Calls to an <externally-invoked procedure>	69
11.2 <SQL procedure statement>	70
12 Data manipulation	73
12.1 <open statement>	73
12.2 <fetch statement>	74
12.3 <close statement>	75
12.4 <select statement: single row>	76
12.5 <delete statement: positioned>	77
12.6 <update statement: positioned>	78
12.7 <temporary table declaration>	79
13 Control statements	81
13.1 <compound statement>	81
13.2 <handler declaration>	85
13.3 <condition declaration>	88
13.4 <SQL variable declaration>	89
13.5 <assignment statement>	90
13.6 <case statement>	93
13.7 <if statement>	95
13.8 <iterate statement>	97
13.9 <leave statement>	98
13.10 <loop statement>	99
13.11 <while statement>	100
13.12 <repeat statement>	101
13.13 <for statement>	102
14 Dynamic SQL	105
14.1 <prepare statement>	105
15 Embedded SQL	107
15.1 <embedded SQL host program>	107
16 Diagnostics management	109
16.1 <get diagnostics statement>	109
16.2 <signal statement>	111
16.3 <resignal statement>	113
17 Information Schema	115
17.1 MODULE_COLUMN_USAGE view	115
17.2 MODULE_PRIVILEGES view	116
17.3 MODULE_TABLE_USAGE view	117

17.4	MODULES view	118
17.5	ROLE_MODULE_GRANTS view	119
17.6	Short name views	120
18	Definition Schema	121
18.1	MODULE_COLUMN_USAGE base table	121
18.2	MODULE_PRIVILEGES base table	123
18.3	MODULE_TABLE_USAGE base table	125
18.4	MODULES base table	126
19	Status codes	129
19.1	SQLSTATE	129
20	Conformance	131
20.1	Claims of conformance	131
Annex A	SQL Conformance Summary	133
Annex B	Implementation-defined elements	135
Annex C	Implementation-dependent elements	137
Annex D	Deprecated features	139
Annex E	Incompatibilities with ISO/IEC 9075:1992	141
Annex F	SQL Feature Taxonomy	143
Index		145

TABLES

Tables	Page
1 Clause, Subclause, and Table relationships	6
2 <identifier>s for use with <get diagnostics statement>	109
3 SQL-statement codes for use in the diagnostics area	110
4 SQLSTATE class and subclass values	129
5 SQL/PSM feature taxonomy for features outside Core SQL	143

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 9075-4, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

This second edition of this part of ISO/IEC 9075 cancels and replaces the first edition, ISO/IEC 9075-4:1996.

ISO/IEC 9075 consists of the following parts, under the general title *Information technology — Database languages — SQL*:

- *Part 1: Framework (SQL/Framework)*
- *Part 2: Foundation (SQL/Foundation)*
- *Part 3: Call-Level Interface (SQL/CLI)*
- *Part 4: Persistent Stored Modules (SQL/PSM)*
- *Part 5: Host Language Bindings (SQL/Bindings)*

Annexes A, B, C, D, E, and F of this part of ISO/IEC 9075 are for information only.

Introduction

The organization of this part of ISO/IEC 9075 is as follows:

- 1) Clause 1, “Scope”, specifies the scope of this part of ISO/IEC 9075.
- 2) Clause 2, “Normative references”, identifies additional standards that, through reference in this part of ISO/IEC 9075, constitute provisions of this part of ISO/IEC 9075.
- 3) Clause 3, “Definitions, notations, and conventions”, defines the notations and conventions used in this part of ISO/IEC 9075.
- 4) Clause 4, “Concepts”, presents concepts used in the definition of persistent stored modules.
- 5) Clause 5, “Lexical elements”, defines a number of lexical elements used in the definition of persistent stored modules.
- 6) Clause 6, “Scalar expressions”, defines a number of scalar expressions used in the definition of persistent stored modules.
- 7) Clause 7, “Query expressions”, defines the elements of the language that produce rows and tables of data as used in persistent stored modules.
- 8) Clause 8, “Additional common elements”, defines additional common elements used in the definition of persistent stored modules.
- 9) Clause 9, “Schema definition and manipulation”, defines the schema definition and manipulation statements associated with the definition of persistent stored modules.
- 10) Clause 10, “Access control”, defines facilities for controlling access to SQL-data.
- 11) Clause 11, “SQL-client modules”, defines the facilities for using persistent stored modules.
- 12) Clause 12, “Data manipulation”, defines data manipulation operations associated with persistent stored modules.
- 13) Clause 13, “Control statements”, defines the control statements used with persistent stored modules.
- 14) Clause 14, “Dynamic SQL”, defines the facilities for executing SQL-statements dynamically in the context of persistent stored modules.
- 15) Clause 15, “Embedded SQL”, defines the host language embeddings.
- 16) Clause 16, “Diagnostics management”, defines enhancements to the facilities used with persistent stored modules.
- 17) Clause 17, “Information Schema”, defines the Information and Definition Schema objects associated with persistent stored modules.
- 18) Clause 18, “Definition Schema”, defines base tables on which the viewed tables containing schema information depend.

ISO/IEC 9075-4:1999 (E)

- 19) Clause 19, “Status codes”, defines SQLSTATE values related to persistent stored modules.
- 20) Clause 20, “Conformance”, defines the criteria for conformance to this part of ISO/IEC 9075.
- 21) Annex A, “SQL Conformance Summary”, is an informative Annex. It summarizes the conformance requirements of the SQL language.
- 22) Annex B, “Implementation-defined elements”, is an informative Annex. It lists those features for which the body of this part of ISO/IEC 9075 states that the syntax, the meaning, the returned results, the effect on SQL-data and/or schemas, or any other behavior is partly or wholly implementation-defined.
- 23) Annex C, “Implementation-dependent elements”, is an informative Annex. It lists those features for which the body of this part of ISO/IEC 9075 states that the syntax, the meaning, the returned results, the effect on SQL-data and/or schemas, or any other behavior is partly or wholly implementation-dependent.
- 24) Annex D, “Deprecated features”, is an informative Annex. It lists features that the responsible Technical Committee intends will not appear in a future revised version of ISO/IEC 9075.
- 25) Annex E, “Incompatibilities with ISO/IEC 9075:1992”, is an informative Annex. It lists the incompatibilities between this edition of this part of ISO/IEC 9075 and ISO/IEC 9075-4:1996.
- 26) Annex F, “SQL Feature Taxonomy”, is an informative Annex. It identifies features of the SQL language specified in this part of ISO/IEC 9075 by a numeric identifier and a short descriptive name. This taxonomy is used to specify conformance to Core SQL and may be used to develop other profiles involving the SQL language.

In the text of this part of ISO/IEC 9075, Clauses begin a new odd-numbered page, and in Clause 5, “Lexical elements”, through Clause 20, “Conformance”, Subclauses begin a new page. Any resulting blank space is not significant.

Information technology — Database languages — SQL —

Part 4:

Persistent Stored Modules (SQL/PSM)

1 Scope

This part of International Standard ISO/IEC 9075 specifies the syntax and semantics of a database language for declaring and maintaining persistent database language routines in SQL-server modules.

The database language for <externally-invoked procedure>s and <SQL-invoked routine>s includes:

- The specification of statements to direct the flow of control.
- The assignment of the result of expressions to variables and parameters.
- The specification of condition handlers that allow SQL-invoked routines to deal with various conditions that arise during their execution.
- The specification of statements to signal and resignal conditions.
- The declaration of local cursors.
- The declaration of local variables.

It also includes the definition of the Information Schema tables that contain schema information pertaining to SQL-server modules and SQL-invoked routines.

NOTE 1 – The context for this part of ISO/IEC 9075 is described by the Reference Model of Data Management (ISO/IEC 10032:1993).

ISO/IEC 9075-4:1999 (E)

2 Normative references

The following standards contain provisions that, through reference in this text, constitute provisions of this part of ISO/IEC 9075. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9075 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8652:1995, *Information technology — Programming languages — Ada*.

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

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

ISO/IEC 9075-3:1996, *Information technology — Database languages — SQL — Part 3: Call-Level Interface (SQL/CLI)*.

ISO/IEC 9075-5, *Information technology — Database languages — SQL — Part 5: Host Language Bindings (SQL/Bindings)*.

ISO/IEC 9075-4:1999 (E)

3 Definitions, notations, and conventions

3.1 Definitions

3.1.1 Definitions provided in Part 4

Insert this paragraph For the purposes of this part of ISO/IEC 9075, the definitions given in ISO/IEC 9075-1 and ISO/IEC 9075-2 apply.

3.2 Notations

Insert this paragraph The syntax notation used in this part of ISO/IEC 9075 is an extended version of BNF ("Backus Normal Form" or "Backus Naur Form"). This version of BNF is fully described in Subclause 6.1, "Notation", of ISO/IEC 9075-1.

3.3 Conventions

Insert this paragraph Except as otherwise specified in this part of ISO/IEC 9075, the conventions used in this part of ISO/IEC 9075 are identical to those described in ISO/IEC 9075-1 and ISO/IEC 9075-2.

3.3.1 Use of terms

3.3.1.1 Exceptions

Modified paragraph The phrase "an exception condition is raised:", followed by the name of a condition, is used in General Rules and elsewhere to indicate that:

- The execution of a statement is unsuccessful.
- The application of General Rules, other than those of Subclause 10.4, "<routine invocation>", in ISO/IEC 9075-2, Subclause 11.2, "<SQL procedure statement>", Subclause 17.1, "<direct SQL statement>", in ISO/IEC 9075-5, Subclause 13.1, "<compound statement>", and Subclause 13.2, "<handler declaration>", may be terminated.
- Diagnostic information is to be made available.
- Execution of the statement is to have no effect on SQL-data or schemas.

Insert this paragraph The phrase "C is re-raised by S" is used in General Rules and elsewhere to indicate that C, a condition raised by an SQL-statement executed during execution of S, is raised again by S.

ISO/IEC 9075-4:1999 (E)
3.3 Conventions

3.3.1.2 Other terms

Insert this paragraph An SQL-statement *S1* may be said to be executed as a *direct result of executing an SQL-statement* if *S1* is the SQL-statement contained in an <externally-invoked procedure> or <SQL-invoked routine> that has been executed.

Insert this paragraph An SQL-statement *S1* may be said to be executed as a *direct result of executing an <SQL control statement>* *S2* if *S2* contains *S1*.

Insert this paragraph The phrase “The scope of a <handler declaration> contained in a/an *Y* is that *Y*, excluding every <SQL schema statement> contained in that *Y*” means that the scope of the <handler declaration> does not extend to SQL-statements contained in such an <SQL schema statement>; it does, however, extend to the <SQL schema statement> itself.

3.3.2 Relationships to other parts of ISO/IEC 9075

3.3.2.1 Clause, Subclause, and Table relationships

Table 1—Clause, Subclause, and Table relationships

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Clause 1, “Scope”	Clause 1, "Scope"	ISO/IEC 9075-2
Clause 2, “Normative references”	Clause 2, "Normative references"	ISO/IEC 9075-2
Clause 3, “Definitions, notations, and conventions”	Clause 3, "Definitions, notations, and conventions"	ISO/IEC 9075-2
Subclause 3.1, “Definitions”	Subclause 3.1, "Definitions"	ISO/IEC 9075-2
Subclause 3.1.1, “Definitions provided in Part 4”	(none)	(none)
Subclause 3.3, “Conventions”	Subclause 3.3, "Conventions"	ISO/IEC 9075-2
Subclause 3.3.1, “Use of terms”	Subclause 3.3.1, "Use of terms"	ISO/IEC 9075-2
Subclause 3.3.1.1, “Exceptions”	Subclause 6.2.3.1, "Exceptions"	ISO/IEC 9075-1
Subclause 3.3.1.2, “Other terms”	Subclause 6.2.3.7, "Other terms"	ISO/IEC 9075-1
Subclause 3.3.2, “Relationships to other parts of ISO/IEC 9075”	(none)	(none)
Subclause 3.3.2.1, “Clause, Subclause, and Table relationships”	(none)	(none)
Subclause 3.4, “Object identifier for Database Language SQL”	Subclause 6.3, "Object identifier for Database Language SQL"	ISO/IEC 9075-1
Clause 4, “Concepts”	Clause 4, "Concepts"	ISO/IEC 9075-2
Subclause 4.1, “SQL-server modules”	(none)	(none)

Table 1—Clause, Subclause, and Table relationships (Cont.)

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 4.2, "SQL-invoked routines"	Subclause 4.23, "SQL-invoked routines"	ISO/IEC 9075-2
Subclause 4.3, "SQL-paths"	Subclause 4.25, "SQL-paths"	ISO/IEC 9075-2
Subclause 4.4, "Tables"	Subclause 4.16, "Tables"	ISO/IEC 9075-2
Subclause 4.5, "SQL-schemas"	Subclause 4.20, "SQL-schemas"	ISO/IEC 9075-2
Subclause 4.6, "Host parameters"	Subclause 4.26, "Host parameters"	ISO/IEC 9075-2
Subclause 4.6.1, "Status parameters"	Subclause 4.26.1, "Status parameters"	ISO/IEC 9075-2
Subclause 4.7, "Diagnostics area"	Subclause 4.27, "Diagnostics area"	ISO/IEC 9075-2
Subclause 4.8, "Cursors"	Subclause 4.29, "Cursors"	ISO/IEC 9075-2
Subclause 4.9, "Condition handling"	(none)	(none)
Subclause 4.10, "SQL-statements"	Subclause 4.30, "SQL-statements"	ISO/IEC 9075-2
Subclause 4.10.1, "SQL-statements classified by function"	Subclause 4.30.2, "SQL-statements classified by function"	ISO/IEC 9075-2
Subclause 4.10.2, "Embeddable SQL-statements"	Subclause 4.6.4, "Embeddable SQL-statements"	ISO/IEC 9075-5
Subclause 4.10.3, "Directly executable SQL-statements"	Subclause 4.6.6, "Directly executable SQL-statements"	ISO/IEC 9075-5
Subclause 4.10.4, "Iterated SQL-statements"	(none)	(none)
Subclause 4.10.5, "SQL-statements and transaction states"	Subclause 4.30.3, "SQL-statements and transaction states"	ISO/IEC 9075-2
Subclause 4.10.6, "Compound statements"	(none)	(none)
Subclause 4.10.7, "SQL-statement atomicity"	(none)	(none)
Subclause 4.11, "Basic security model"	Subclause 4.31, "Basic security model"	ISO/IEC 9075-2
Subclause 4.11.1, "Privileges"	Subclause 4.31, "Basic security model"	ISO/IEC 9075-2
Clause 5, "Lexical elements"	Clause 5, "Lexical elements"	ISO/IEC 9075-2
Subclause 5.1, "<token> and <separator>"	Subclause 5.2, "<token> and <separator>"	ISO/IEC 9075-2
Subclause 5.2, "Names and identifiers"	Subclause 5.4, "Names and identifiers"	ISO/IEC 9075-2
Clause 6, "Scalar expressions"	Clause 6, "Scalar expressions"	ISO/IEC 9075-2

ISO/IEC 9075-4:1999 (E)
3.3 Conventions

Table 1—Clause, Subclause, and Table relationships (Cont.)

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 6.1, "<value specification> and <target specification>"	Subclause 6.3, "<value specification> and <target specification>"	ISO/IEC 9075-2
Subclause 6.2, "<identifier chain>"	Subclause 6.5, "<identifier chain>"	ISO/IEC 9075-2
Subclause 6.3, "<SQL variable reference>"	(none)	(none)
Clause 7, "Query expressions"	Clause 7, "Query expressions"	ISO/IEC 9075-2
Subclause 7.1, "<query specification>"	Subclause 7.11, "<query specification>"	ISO/IEC 9075-2
Clause 8, "Additional common elements"	Clause 10, "Additional common elements"	ISO/IEC 9075-2
Subclause 8.1, "<routine invocation>"	Subclause 10.4, "<routine invocation>"	ISO/IEC 9075-2
Subclause 8.2, "<privileges>"	Subclause 10.5, "<privileges>"	ISO/IEC 9075-2
Subclause 8.3, "<sqlstate value>"	(none)	(none)
Clause 9, "Schema definition and manipulation"	Clause 11, "Schema definition and manipulation"	ISO/IEC 9075-2
Subclause 9.1, "<schema definition>"	Subclause 11.1, "<schema definition>"	ISO/IEC 9075-2
Subclause 9.2, "<drop schema statement>"	Subclause 11.2, "<drop schema statement>"	ISO/IEC 9075-2
Subclause 9.3, "<default clause>"	Subclause 11.5, "<default clause>"	ISO/IEC 9075-2
Subclause 9.4, "<drop column scope clause>"	Subclause 11.16, "<drop column scope clause>"	ISO/IEC 9075-2
Subclause 9.5, "<drop column definition>"	Subclause 11.17, "<drop column definition>"	ISO/IEC 9075-2
Subclause 9.6, "<drop table constraint definition>"	Subclause 11.19, "<drop table constraint definition>"	ISO/IEC 9075-2
Subclause 9.7, "<drop table statement>"	Subclause 11.20, "<drop table statement>"	ISO/IEC 9075-2
Subclause 9.8, "<view definition>"	Subclause 11.21, "<view definition>"	ISO/IEC 9075-2
Subclause 9.9, "<drop view statement>"	Subclause 11.22, "<drop view statement>"	ISO/IEC 9075-2
Subclause 9.10, "<drop domain statement>"	Subclause 11.29, "<drop domain statement>"	ISO/IEC 9075-2
Subclause 9.11, "<drop character set statement>"	Subclause 11.31, "<drop character set statement>"	ISO/IEC 9075-2
Subclause 9.12, "<drop collation statement>"	Subclause 11.33, "<drop collation statement>"	ISO/IEC 9075-2

Table 1—Clause, Subclause, and Table relationships (Cont.)

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 9.13, "<drop translation statement>"	Subclause 11.35, "<drop translation statement>"	ISO/IEC 9075-2
Subclause 9.14, "<assertion definition>"	Subclause 11.36, "<assertion definition>"	ISO/IEC 9075-2
Subclause 9.15, "<drop assertion statement>"	Subclause 11.37, "<drop assertion statement>"	ISO/IEC 9075-2
Subclause 9.16, "<trigger definition>"	Subclause 11.38, "<trigger definition>"	ISO/IEC 9075-2
Subclause 9.17, "<drop user-defined ordering statement>"	Subclause 11.55, "<drop user-defined ordering statement>"	ISO/IEC 9075-2
Subclause 9.18, "<SQL-server module definition>"	(none)	(none)
Subclause 9.19, "<drop module statement>"	(none)	(none)
Subclause 9.21, "<SQL-invoked routine>"	Subclause 11.49, "<SQL-invoked routine>"	ISO/IEC 9075-2
Subclause 9.22, "<drop routine statement>"	Subclause 11.51, "<drop routine statement>"	ISO/IEC 9075-2
Clause 10, "Access control"	Clause 12, "Access control"	ISO/IEC 9075-2
Subclause 10.1, "<grant statement>"	Subclause 12.1, "<grant statement>"	ISO/IEC 9075-2
Subclause 10.2, "<revoke statement>"	Subclause 12.6, "<revoke statement>"	ISO/IEC 9075-2
Clause 11, "SQL-client modules"	Subclause 13.1, "<SQL-client module definition>"	ISO/IEC 9075-2
Subclause 11.1, "Calls to an <externally-invoked procedure>"	Subclause 13.4, "Calls to an <externally-invoked procedure>"	ISO/IEC 9075-2
Subclause 11.2, "<SQL procedure statement>"	Subclause 13.5, "<SQL procedure statement>"	ISO/IEC 9075-2
Clause 12, "Data manipulation"	Clause 14, "Data manipulation"	ISO/IEC 9075-2
Subclause 12.1, "<open statement>"	Subclause 14.2, "<open statement>"	ISO/IEC 9075-2
Subclause 12.2, "<fetch statement>"	Subclause 14.3, "<fetch statement>"	ISO/IEC 9075-2
Subclause 12.3, "<close statement>"	Subclause 14.4, "<close statement>"	ISO/IEC 9075-2
Subclause 12.4, "<select statement: single row>"	Subclause 14.5, "<select statement: single row>"	ISO/IEC 9075-2
Subclause 12.5, "<delete statement: positioned>"	Subclause 14.6, "<delete statement: positioned>"	ISO/IEC 9075-2
Subclause 12.6, "<update statement: positioned>"	Subclause 14.9, "<update statement: positioned>"	ISO/IEC 9075-2

ISO/IEC 9075-4:1999 (E)
3.3 Conventions

Table 1—Clause, Subclause, and Table relationships (Cont.)

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Subclause 12.7, "<temporary table declaration>"	Subclause 14.11, "<temporary table declaration>"	ISO/IEC 9075-2
Clause 13, "Control statements"	(none)	(none)
Subclause 13.1, "<compound statement>"	(none)	(none)
Subclause 13.2, "<handler declaration>"	(none)	(none)
Subclause 13.3, "<condition declaration>"	(none)	(none)
Subclause 13.4, "<SQL variable declaration>"	(none)	(none)
Subclause 13.5, "<assignment statement>"	(none)	(none)
Subclause 13.6, "<case statement>"	(none)	(none)
Subclause 13.7, "<if statement>"	(none)	(none)
Subclause 13.8, "<iterate statement>"	(none)	(none)
Subclause 13.9, "<leave statement>"	(none)	(none)
Subclause 13.10, "<loop statement>"	(none)	(none)
Subclause 13.11, "<while statement>"	(none)	(none)
Subclause 13.12, "<repeat statement>"	(none)	(none)
Subclause 13.13, "<for statement>"	(none)	(none)
Clause 14, "Dynamic SQL"	Clause 15, "Dynamic SQL"	ISO/IEC 9075-5
Subclause 14.1, "<prepare statement>"	Subclause 15.6, "<prepare statement>"	ISO/IEC 9075-5
Clause 15, "Embedded SQL"	Clause 16, "Embedded SQL"	ISO/IEC 9075-5
Subclause 15.1, "<embedded SQL host program>"	Subclause 16.1, "<embedded SQL host program>"	ISO/IEC 9075-5
Clause 16, "Diagnostics management"	Clause 19, "Diagnostics management"	ISO/IEC 9075-2
Subclause 16.1, "<get diagnostics statement>"	Subclause 19.1, "<get diagnostics statement>"	ISO/IEC 9075-2
Subclause 16.2, "<signal statement>"	(none)	(none)
Subclause 16.3, "<resignal statement>"	(none)	(none)

Table 1—Clause, Subclause, and Table relationships (Cont.)

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Clause 17, "Information Schema"	Clause 20, "Information Schema"	ISO/IEC 9075-2
Subclause 17.1, "MODULE_COLUMN_USAGE view"	(none)	(none)
Subclause 17.2, "MODULE_PRIVILEGES view"	(none)	(none)
Subclause 17.3, "MODULE_TABLE_USAGE view"	(none)	(none)
Subclause 17.4, "MODULES view"	(none)	(none)
Clause 18, "Definition Schema"	Clause 21, "Definition Schema"	ISO/IEC 9075-2
Subclause 18.1, "MODULE_COLUMN_USAGE base table"	(none)	(none)
Subclause 18.2, "MODULE_PRIVILEGES base table"	(none)	(none)
Subclause 18.3, "MODULE_TABLE_USAGE base table"	(none)	(none)
Subclause 18.4, "MODULES base table"	(none)	(none)
Clause 19, "Status codes"	Clause 22, "Status codes"	ISO/IEC 9075-2
Subclause 19.1, "SQLSTATE"	Subclause 22.1, "SQLSTATE"	ISO/IEC 9075-2
Clause 20, "Conformance"	Clause 8, "Conformance"	ISO/IEC 9075-1
Subclause 20.1, "Claims of conformance"	Subclause 8.1.5, "Claims of conformance"	ISO/IEC 9075-1
Annex A, "SQL Conformance Summary"	Appendix A, "SQL Conformance Summary"	ISO/IEC 9075-2
Annex B, "Implementation-defined elements"	Appendix B, "Implementation-defined elements"	ISO/IEC 9075-2
Annex C, "Implementation-dependent elements"	Appendix C, "Implementation-dependent elements"	ISO/IEC 9075-2
Annex D, "Deprecated features"	Appendix D, "Deprecated features"	ISO/IEC 9075-2
Annex E, "Incompatibilities with ISO/IEC 9075:1992"	Appendix E, "Incompatibilities with ISO/IEC 9075:1992 and ISO/IEC 9075-4:1996"	ISO/IEC 9075-2
Annex F, "SQL Feature Taxonomy"	Appendix F, "SQL feature and package taxonomy"	ISO/IEC 9075-2
Table 1, "Clause, Subclause, and Table relationships"	(none)	(none)
Table 2, "<identifier>s for use with <get diagnostics statement>"	Table 25, "<identifier>s for use with <get diagnostics statement>"	ISO/IEC 9075-2

ISO/IEC 9075-4:1999 (E)
3.3 Conventions

Table 1—Clause, Subclause, and Table relationships (Cont.)

Clause, Subclause, or Table in this part of ISO/IEC 9075	Corresponding Clause, Subclause, or Table from another part	Part containing correspondence
Table 3, “SQL-statement codes for use in the diagnostics area”	Table 26, “SQL-statement codes”	ISO/IEC 9075-2
Table 4, “SQLSTATE class and subclass values”	Table 27, “SQLSTATE class and subclass values”	ISO/IEC 9075-2
Table 5, “SQL/PSM feature taxonomy for features outside Core SQL”	Table 13, “SQL/Bindings feature taxonomy for features outside Core SQL”	ISO/IEC 9075-5

3.4 Object identifier for Database Language SQL

The object identifier for Database Language SQL is defined in ISO/IEC 9075-1 in Subclause 6.3, “Object identifier for Database Language SQL”, with the following additions:

Format

```
<Part 4 yes> ::= <Part 4 conformance> <Part 4 module>  
<Part 4 conformance> ::= 4 | sqlpsm1999 <left paren> 4 <right paren>  
<Part 4 module> ::= <Part 4 module yes> | <Part 4 module no>  
<Part 4 module yes> ::= 1 | moduleyes <left paren> 1 <right paren>  
<Part 4 module no> ::= 0 | moduleno <left paren> 0 <right paren>
```

Syntax Rules

- 1) Specification of <Part 4 yes> implies that conformance to ISO/IEC 9075-4 is claimed.
- 2) Specification of <Part 4 module yes> implies that conformance to Feature P01, “Stored modules”, is claimed.
- 3) Specification of <Part 4 module no> implies that conformance to Feature P01, “Stored modules”, is not claimed.

4 Concepts

4.1 SQL-server modules

An *SQL-server module* is a persistent object defined in a schema and identified by an <SQL-server module name>. SQL-server modules are created with <SQL-server module definition>s and destroyed with <drop module statement>s and by <drop schema statement>s that destroy the schemas that contain them.

An <SQL-server module definition> contains an <SQL-server module name>, an optional <SQL-server module character set specification>, an optional <SQL-server module schema clause>, an optional <SQL-server module path specification>, zero or more declared local temporary tables specified by <temporary table declaration>s, and one or more <SQL-invoked routine>s.

The <SQL-server module name> of an SQL-server module is a <schema qualified name>. The character set specified by the <SQL-server module character set specification> identifies the character repertoire used for expressing the names of schema objects used in the <SQL-server module definition>. The <default schema name> specified by the <SQL-server module schema clause> identifies the schema name used for implicit qualification of unqualified names appearing in the <SQL-server module definition>. The SQL-invoked routines of an SQL-server module are invoked only from SQL-statements.

An SQL-server module has an *SQL-server module authorization identifier*, which is set to the authorization identifier of the owner of the schema that contains the SQL-server module at the time the SQL-server module is created. The SQL-server module authorization identifier acts as the current authorization identifier for privilege determination for the SQL objects, if any, contained in the SQL-server module.

An SQL-server module is described by an SQL-server module descriptor. An SQL-server module descriptor includes:

- The SQL-server module name of the SQL-server module.
- The descriptor of the character set in which the SQL-server module is represented.
- The default schema name used for implicit qualification of unqualified names in the SQL-server module.
- The SQL-server module authorization identifier of the SQL-server module.
- The list of schema names contained in the <SQL-server module path specification>.
- The table descriptor of every local temporary table declared in the SQL-server module.
- The descriptor of every SQL-invoked routine contained in the SQL-server module.
- The text of the <SQL-server module definition>.

4.2 SQL-invoked routines

Replace 2nd paragraph An SQL-invoked routine is either a component of an <SQL-server module definition> or an element of an SQL-schema. An SQL-invoked routine that is an element of an SQL-schema is called a schema-level routine.

Replace 22nd paragraph An SQL-invoked routine has a *routine SQL-path*, which is inherited from its containing SQL-server module or schema, the current SQL-session, or the containing SQL-client module.

Insert in the 23rd paragraph — If the SQL-invoked routine is not a schema-level routine, then the <SQL-server module name> of the SQL-server module that includes the SQL-invoked routine and the <schema name> of the schema that includes the SQL-server module.

4.3 SQL-paths

Replace 3rd paragraph The value specified by CURRENT_PATH is the value of the SQL-path of the current SQL-session. This SQL-path is used to search for the subject routine of a <routine invocation> whose <routine name> does not contain a <schema name> when the <routine invocation> is contained in <preparable statement>s that are prepared in the current SQL-session by either an <execute immediate statement> or a <prepare statement>, or contained in <direct SQL statement>s that are invoked directly. The definition of SQL-schemas and SQL-server modules specify an SQL-path that is used to search for the subject routine of a <routine invocation> whose <routine name>s do not contain a <schema name> when the <routine invocation> is contained respectively in the <schema definition> or the <SQL-server module definition>.

4.4 Tables

Replace 16th paragraph A declared local temporary table may be declared in an SQL-client module or in an SQL-server module.

Insert after 16th paragraph A declared local temporary table that is declared in an SQL-server module is a named table defined by a <temporary table declaration> that is effectively materialized the first time any <module routine> in the <SQL-server module definition> that contains the <temporary table declaration> is executed.

Insert after 16th paragraph A declared local temporary table is accessible only by <module routine>s in the <SQL-server module definition> that contains the <temporary table declaration>. The effective <schema name> of the <schema qualified name> of the declared local temporary table may be thought of as the implementation-dependent SQL-session identifier associated with the SQL-session and the name of the <SQL-server module definition> that contains the <temporary table declaration>.

4.5 SQL-schemas

Replace 2nd paragraph In this part of ISO/IEC 9075, the term “schema” is used only in the sense of SQL-schema. Each component descriptor is either a domain descriptor, a base table descriptor, a view descriptor, a constraint descriptor, a privilege descriptor, a character set descriptor, a collation descriptor, a translation descriptor, a trigger descriptor, a user-defined type descriptor, an SQL-server module descriptor, or an SQL-invoked routine descriptor. The persistent objects described by

the descriptors are said to be *owned by* or to have been *created by* the <authorization identifier> of the schema.

4.6 Host parameters

4.6.1 Status parameters

Insert this paragraph Exception conditions or completion conditions may be raised during the execution of an <SQL procedure statement>. One of the conditions becomes the active condition when the <SQL procedure statement> terminates; the *active condition* is the condition returned in SQLSTATE. If the active condition is an exception condition, then it is called the *active exception condition*. If the active condition is a completion condition, then it is called the *active completion condition*.

Insert this paragraph If the <SQL procedure statement> is a <compound statement>, then the active condition may result from the action of some exception handler specified in the <compound statement>.

4.7 Diagnostics area

Insert this paragraph An implementation shall place information about a completion or exception condition that causes a handler to be activated into the diagnostics area prior to activating the handler. If other conditions are raised, then it is implementation-defined whether the implementation places information about them into the diagnostics area.

Insert this paragraph The diagnostics area is emptied during the execution of a <signal statement>. Information is added to the diagnostics area during the execution of a <resignal statement>.

4.8 Cursors

Insert this paragraph For every <declare cursor> in a <compound statement>, a cursor is effectively created each time the <compound statement> is executed, and destroyed when that execution completes.

4.9 Condition handling

Condition handling is the method of handling exception and completion conditions in SQL/PSM. Condition handling provides a <handler declaration> to define a handler, specifying its type, the exception and completion conditions it can resolve, and the action it takes to do so. Condition handling also provides the ability to explicitly signal exception and completion conditions.

<handler declaration>s specify the handling of exception and completion conditions. <handler declaration>s can be specified in <compound statement>s. The scope of a <handler declaration> specified in a <compound statement> is that <compound statement> excluding every <SQL schema statement> contained in that <compound statement>.

A <handler declaration> associates one or more conditions with a handler action. The handler action is an <SQL procedure statement>.

A *general <handler declaration>* is one that is associated with the <condition value>s SQL EXCEPTION, SQLWARNING, or NOT FOUND. All other <handler declaration>s are *specific <handler declaration>s*.

ISO/IEC 9075-4:1999 (E)

4.9 Condition handling

A condition represents an error or informational state caused by execution of an <SQL procedure statement>. Conditions are raised to provide information in the diagnostics area about the execution of an <SQL procedure statement>.

A <condition declaration> is used to declare a <condition name>, and to optionally associate it with an SQLSTATE value. If a <condition declaration> does not specify an SQLSTATE value, it declares a *user-defined exception condition*. <condition name>s can be used in <handler declaration>s, <signal statement>s, and <resignal statement>s.

When the <compound statement> containing a <handler declaration> is executed, a handler is created for the conditions associated with that <handler declaration>. A created handler is *activated* when it is the most appropriate handler for an exception or completion condition that has been raised by an SQL-statement. Such a handler is an *active* handler.

The *most appropriate* handler is determined during execution of an implicit or explicit <resignal statement>. An implicit <resignal statement> is executed when a <compound statement> or <handler action> completes with a condition other than *successful completion*.

If there is no most appropriate handler and the condition is an exception condition, then the SQL-statement raising the exception condition is terminated with that exception condition. This type of exception condition is called an *unhandled exception condition*. Unhandled exception conditions are examined at the next visible scope for handling. If an exception condition remains unhandled at the outermost <externally-invoked procedure> or <direct SQL statement>, it is seen by the SQL-client. Even if the SQL-client resolves the exception condition, execution is not resumed in the SQL-server where the exception condition was raised.

If there is no most appropriate handler and the condition is a completion condition, then execution is resumed as specified in Subclause 6.2.3.1, "Exceptions", in ISO/IEC 9075-1. This type of completion condition is called an *unhandled completion condition*.

A handler type specifies CONTINUE, EXIT, or UNDO.

If a handler type specifies CONTINUE, then, when the handler is activated, it will:

- Execute the handler action.
- Return control to the SQL-statement following the executing statement that raised the condition. If the SQL-statement that raised the condition is an <SQL control statement>, then "the following SQL-statements" do not include the <SQL procedure statement>s contained within the <SQL control statement>.

NOTE 2 – For example, if the executing statement that raised the condition is an <if statement>, control is returned to the SQL-statement *following* the <if statement>.

If a handler type specifies EXIT, then, when the handler is activated, it will:

- Execute the handler action.
- Implicitly LEAVE the <compound statement> for which the handler was created, with no active exception condition.

If a handler type specifies UNDO, then, when the handler is activated, it will:

- Roll back all of the changes to SQL-data or to schemas by the execution of every SQL-statement contained in the SQL-statement list of the <compound statement> at the scope of the handler and cancel any <SQL procedure statement>s triggered by the execution of such statements.
- Execute the handler action.

- Return control to the end of the <compound statement> for which the handler was created.

If a <handler action> completes with a completion condition: *successful completion*, then it was able to resolve the condition, and execution resumes as specified in Subclause 13.2, “<handler declaration>”.

If a <handler action> completes with an exception or completion condition other than *successful completion*, then an implicit <resignal statement> is executed. The <resignal statement> determines whether there is another <handler declaration> that can resolve the condition.

4.10 SQL-statements

4.10.1 SQL-statements classified by function

Insert this paragraph The following are additional main classes of SQL-statements:

- SQL-control declarations

Insert this paragraph The following are additional SQL-schema statements:

- <SQL-server module definition>
- <alter module statement>
- <drop module statement>

Insert this paragraph The following are additional SQL-control statements:

- <compound statement>
- <case statement>
- <if statement>
- <iterate statement>
- <leave statement>
- <loop statement>
- <while statement>
- <repeat statement>
- <for statement>
- <assignment statement>

Insert this paragraph The following are the SQL-control declarations:

- <condition declaration>
- <handler declaration>
- <SQL variable declaration>

4.10 SQL-statements

Insert this paragraph The following are additional SQL-diagnostics statements:

- <signal statement>
- <resignal statement>

4.10.2 Embeddable SQL-statements

Insert this paragraph The following are additional SQL-statements that are embeddable in an <embedded SQL host program> and that may be the <SQL procedure statement> in an <externally-invoked procedure> in an SQL-client module:

- All SQL-control statements

NOTE 3 – SQL-control declarations contained in (for example) <compound statement>s are permitted, even when the containing SQL-statement is embedded in an <embedded SQL host program>.

4.10.3 Directly executable SQL-statements

Insert this paragraph The following are additional SQL-statements that may be executed directly:

- All SQL-control statements

4.10.4 Iterated SQL-statements

The following are the iterated SQL-statements:

- <loop statement>
- <while statement>
- <repeat statement>
- <for statement>

4.10.5 SQL-statements and transaction states

Insert this paragraph The following additional SQL-statement is a transaction-initiating SQL-statement:

- <for statement>

Insert this paragraph The following additional SQL-statement is not a transaction-initiating SQL-statement:

- <iterate statement>
- <leave statement>

Insert this paragraph The following additional SQL-statements are possibly transaction-initiating SQL-statements:

- SQL-control statements other than:
 - <for statement>
 - <iterate statement>
 - <leave statement>

Insert this paragraph If the initiation of an SQL-transaction occurs in an atomic execution context, and an SQL-transaction has already been completed in this atomic execution context, then an exception condition is raised: *invalid transaction initiation*.

4.10.6 Compound statements

A compound statement allows a sequence of SQL-statements to be considered as a single SQL-statement. A compound statement also defines a local scope in which SQL-variables, condition handlers, and cursors can be declared. See Subclause 13.1, "<compound statement>".

4.10.7 SQL-statement atomicity

Augment 1st paragraph The execution of <compound statement>s that specify ATOMIC is atomic.

4.11 Basic security model

4.11.1 Privileges

Insert this paragraph A privilege further authorizes a given category of <action> to be performed on a specified SQL-server module by a specified <authorization identifier>.

Insert this paragraph An *execute privilege descriptor* may also identify the existence of a privilege on the SQL-server module identified by the privilege descriptor.

Insert this paragraph The identification included in an EXECUTE privilege descriptor may also identify the SQL-server module described by the descriptor.

Insert this paragraph Individual SQL-invoked routines contained in an SQL-server module cannot be associated with EXECUTE privilege descriptors. Only schema-level routines and SQL-server modules are associated with EXECUTE privilege descriptors.

NOTE 4 – "schema-level routine" is defined in Subclause 11.49, "<SQL-invoked routine>", in ISO/IEC 9075-2.

ISO/IEC 9075-4:1999 (E)

5 Lexical elements

5.1 <token> and <separator>

Function

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

Format

```
<non-reserved word> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !! All alternatives from ISO/IEC 9075-5
    | CONDITION_IDENTIFIER
```

```
<reserved word> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !! All alternatives from ISO/IEC 9075-5

    | CONDITION

    | DO

    | ELSEIF | EXIT

    | HANDLER

    | IF | ITERATE

    | LEAVE | LOOP

    | REDO | REPEAT | RESIGNAL

    | SIGNAL

    | UNDO | UNTIL

    | WHILE
```

Syntax Rules

No additional Syntax Rules.

Access Rules

No additional Access Rules.

ISO/IEC 9075-4:1999 (E)
5.1 <token> and <separator>

General Rules

No additional General Rules.

5.2 Names and identifiers

Function

Specify names.

Format

<SQL-server module name> ::=
 <schema qualified name>

<SQL variable name> ::=
 <identifier>

<condition name> ::=
 <identifier>

Syntax Rules

- 1) Replace ISO/IEC 9075-5 SR1) If the <local or schema qualified name> is contained, without an intervening <schema definition> or <SQL-server module definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the default <unqualified schema name> for the SQL-session is implicit.
- 2) Insert before SR4)a) If the <local or schema qualified name> is contained in an <SQL-server module definition> without an intervening <schema definition>, then the <default schema name> that is specified or implicit in the <SQL-server module definition> is implicit.
- 3) Replace ISO/IEC 9075-5 SR2) If the <schema qualified name> is contained, without an intervening <schema definition> or <SQL-server module definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the default <unqualified schema name> for the SQL-session is implicit.
- 4) Insert before SR11)a) If the <schema qualified name> is contained in an <SQL-server module definition> without an intervening <schema definition>, then the <default schema name> that is specified or implicit in the <SQL-server module definition> is implicit.
- 5) Replace SR8) Case:
 - a) If <user-defined type name> *UDTN* with a <qualified identifier> *QI* is simply contained in <user-defined type>, then
 Case:
 - i) If *UDTN* contains a <schema name> *SN*, then the schema identified by *SN* shall contain the descriptor of a user-defined type *UDT* such that the <qualified identifier> of *UDT* is equivalent to *QI*. *UDT* is the user-defined type identified by *UDTN*.

ISO/IEC 9075-4:1999 (E)
5.2 Names and identifiers

- ii) Otherwise:
 - 1) Case:
 - A) If *UDTN* is contained, without an intervening <schema definition> or <SQL-server module definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or by a <prepare statement> or in a <direct SQL statement> that is invoked directly, then let *DP* be the SQL-path of the current SQL-session.
 - B) If *UDTN* is contained in an <SQL-server module definition> without in intervening <schema definition>, then let *DP* be the SQL-path of that <SQL-server module definition>.
 - C) If *UDTN* is contained in a <schema definition> that is not contained in an <SQL-client module definition>, then let *DP* be the SQL-path of that <schema definition>.
 - D) Otherwise, *UDTN* is contained in an <SQL-client module definition>; let *DP* be the SQL-path of that <SQL-client module definition>.
 - 2) Let *N* be the number of <schema name>s in *DP*. Let *S_i*, 1 (one) ≤ *i* ≤ *N*, be the *i*-th <schema name> in *DP*.
 - 3) Let the set of subject types be the set containing every user-defined type *T* in the schema identified by some *S_i*, 1 (one) ≤ *i* ≤ *N*, such that the <qualified identifier> of *T* is equivalent to *QI*. There shall be at least one type in the set of subject types.
 - 4) Case:
 - a) If the set of subject types contains exactly one user-defined type *UDT*, then *UDT* is the user-defined type identified by *UDTN*.
 - b) Otherwise, let *UDT* be the user-defined type contained in the set of subject types such that there is no other type *UDT2* for which the <schema name> of the schema that includes the user-defined type descriptor of *UDT2* precedes in *DP* the <schema name> identifying the schema that includes the user-defined type descriptor of *UDT*. *UDTN* identifies *UDT*.
 - 5) The implicit <schema name> of *UDTN* is the <schema name> of the schema that includes the user-defined type descriptor of *UDT*.
- b) Otherwise,
 - Case:
 - i) If *UDTN* is contained, without an intervening <schema definition> or <SQL-server module definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or by a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the implicit <schema name> of *UDTN* is the default <unqualified schema name> of the current SQL-session.
 - ii) If *UDTN* is contained in an <SQL-server module definition> without in intervening <schema definition>, then the implicit <schema name> of *UDTN* is the <schema name> that is specified or implicit in <SQL-server module definition>.

- iii) If *UDTN* is contained in a <schema definition> that is not contained in an <SQL-client module definition>, then the implicit <schema name> of *UDTN* is the <schema name> that is specified or implicit in <schema definition>.
- iv) Otherwise, *UDTN* is contained in an <SQL-client module definition>; the implicit <schema name> of *UDTN* is the <schema name> that is specified or implicit in <SQL-client module definition>.

Access Rules

No additional Access Rules.

General Rules

- 1) Insert this GR An <SQL-server module name> identifies an SQL-server module.
- 2) Insert this GR An <SQL variable name> identifies an SQL variable.
- 3) Insert this GR A <condition name> identifies an exception condition or a completion condition and optionally a corresponding SQLSTATE value.

6 Scalar expressions

6.1 <value specification> and <target specification>

Function

Specify one or more values, host parameters, SQL parameters, dynamic parameters, host variables, or SQL variables.

Format

```
<general value specification> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !! All alternatives from ISO/IEC 9075-5
    | <SQL variable reference>

<simple value specification> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !! All alternatives from ISO/IEC 9075-5
    | <SQL variable reference>

<target specification> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !! All alternatives from ISO/IEC 9075-5
    | <SQL variable reference>

<simple target specification> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !! All alternatives from ISO/IEC 9075-5
    | <SQL variable reference>
```

Syntax Rules

No additional Syntax Rules.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

6.2 <identifier chain>

Function

Disambiguate a <period>-separated chain of identifiers.

Format

No additional Format items.

Syntax Rules

- 1) Replace introductory paragraph of SR7) For at most one j between 1 and M , PIC_j is called the *basis* of IC , and j is called the *basis length* of IC . The *referent* of the basis is a column C of a table, an SQL parameter SP , or an SQL variable SV . The basis, basis length, basis scope and basis referent of IC are determined as follows:
- 2) Replace SR7)a) If $N = 1$, then IC shall be contained within the scope of one or more exposed <table or query name>s or <correlation name>s whose associated tables include a column whose <identifier> is equivalent to I_1 or within the scope of a <routine name> whose associated <SQL parameter declaration list> includes an SQL parameter whose <identifier> is equivalent to I_1 or within the scope of one or more <beginning label>s whose associated <local declaration list> includes an SQL variable whose <identifier> is equivalent to I_1 . Let the phrase *possible scope tags* denote those exposed <table name>s, <correlation name>s, and <routine name>s.
- 3) Insert after SR7)a)i)2) If the innermost possible qualifier is a <beginning label>, then let SV be the SQL variable whose <identifier> is equivalent to I_1 . PIC_1 is the basis of IC , the basis length is 1 (one), the basis scope is the scope of SP , and the basis referent is SV .
- 4) Insert before SR7)b)ii) If IC is contained within the scope of a <beginning label> whose associated <local declaration list> includes an SQL variable SV whose <identifier> is equivalent to I_1 , then PIC_1 is a candidate basis of IC , the scope of PIC_1 is the scope of SV , and the referent of PIC_1 is SV .
- 5) Replace SR9) If $BL < N$, then let TIC be the <value expression primary>:

$$(PIC_{BL}) <period> I_{BL+1} <period> \dots <period> I_N$$

The Syntax Rules of Subclause 6.23, "<value expression>", are applied to TIC , yielding a column reference, an SQL parameter reference, or an SQL variable reference, and $(N-BL)$ <field reference>s, <method invocation>s, <modified field reference>s, and/or <mutator reference>s.

NOTE 5 – In this transformation, (PIC_{BL}) is interpreted as a <value expression primary> of the form <left paren> <value expression> <right paren>. PIC_{BL} is a <value expression> that is a <value expression primary> that is an <unsigned value specification> that is either a <column reference> or an <SQL parameter reference>. The identifiers $I_{BL+1}, \dots, I_N$ are parsed using the Syntax Rules of <field reference> and <method invocation>. Alternatively, on the left-hand side of an <assignment statement>, (PIC_{BL}) is interpreted as "<left paren> <target specification> <right paren>", and the identifiers $I_{BL+1}, \dots, I_N$ are parsed using the Syntax Rules of <modified field reference> and <mutator reference>.

- 6) Insert after SR12) A <basic identifier chain> whose basis referent is an SQL variable is an *SQL variable reference*.

Access Rules

None.

General Rules

- 1) Insert this GR If *BIC* is an SQL variable reference, then *BIC* references the SQL variable *SV* of a given execution of the <compound statement> whose <local declaration list> contains the <SQL variable declaration> that declares *SV*.

6.3 <SQL variable reference>

Function

Reference an SQL variable.

Format

```
<SQL variable reference> ::=  
    <basic identifier chain>
```

Syntax Rules

- 1) An <SQL variable reference> shall be a <basic identifier chain> that is an SQL variable reference.

Access Rules

None.

General Rules

None.

7 Query expressions

7.1 <query specification>

Function

Specify a table derived from the result of a <table expression>.

Format

No additional Format items.

Syntax Rules

- 1) Insert after SR7)f)i) If IC is contained within the scope of a <beginning label> whose associated <local declaration list> includes an SQL variable SV whose <identifier> is equivalent to I_1 , then PIC_1 is a candidate basis of IC and the scope of PIC_1 is the scope of SP .
- 2) Insert after SR2) in Part 5 An SQL variable.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

8 Additional common elements

8.1 <routine invocation>

Function

Invoke an SQL-invoked routine.

Format

No additional Format items.

Syntax Rules

- 1) Replace SR5) An SQL-invoked routine *R* is an *executable routine* if and only if *R* is a possibly candidate routine and

Case:

- a) If *RI* is contained in an <SQL schema statement>, then

Case:

- i) If *RI* is contained in an <SQL-server module definition> *M*, then the applicable privileges of the <authorization identifier> that owns the containing schema include EXECUTE on *M*.
- ii) Otherwise, the applicable privileges of the <authorization identifier> that owns the containing schema include EXECUTE on *R*.

- b) Otherwise,

Case:

- i) If *RI* is contained in an <SQL-server module definition> *M*, then the current privileges include EXECUTE on *M*.
- ii) Otherwise, the current privileges include EXECUTE on *R*.

NOTE 6 – “applicable privileges” and “current privileges” are defined in Subclause 10.5, “<privileges>”, in ISO/IEC 9075-2.

- 2) Insert before SR7)b)i)1)C)i) If *RI* is contained in an <SQL-server module definition>, then let *DP* be the SQL-path of that <SQL-server module definition>.
- 3) Replace SR7)b)i)1)C)i) If *RI* is contained in a <schema definition> without an intervening <SQL-server module definition>, then let *DP* be the SQL-path of that <schema definition>.
- 4) Insert before SR8)b)i)1)C)i) If *RI* is contained in an <SQL-server module definition>, then let *DP* be the SQL-path of that <SQL-server module definition>.

ISO/IEC 9075-4:1999 (E)

8.1 <routine invocation>

- 5) Replace SR8)b)i)1)C)i) If RI is contained in a <schema definition> without an intervening <SQL-server module definition>, then let DP be the SQL-path of that <schema definition>.
- 6) Replace SR8)c)i)4)B) If A_i is an <SQL variable name> or the <SQL parameter name> of an SQL parameter of an SQL-invoked routine, then P_i shall be assignable to A_i , according to the Syntax Rules of Subclause 9.2, "Store assignment", in ISO/IEC 9075-2, with A_i and P_i as *TARGET* and *VALUE*, respectively.

Access Rules

No additional Access Rules.

General Rules

- 1) Insert after GR6)c)i)
NOTE 7 – The identities of declared local temporary tables that are defined in <SQL-server module>s are not removed.
- 2) Replace SR10)b)i) If TS_i is an <SQL variable name> or a <host parameter specification>, then CPV_i is assigned to TS_i according to the rules of Subclause 9.1, "Retrieval assignment", in ISO/IEC 9075-2.
- 3) Replace SR10)b)i) If TS_i is an <SQL variable name> or the <SQL parameter name> of an SQL parameter of an SQL-invoked routine, then CPV_i is assigned to TS_i according to the rules of Subclause 9.2, "Store assignment", in ISO/IEC 9075-2.

8.2 <privileges>

Function

Specify privileges.

Format

```
<object name> ::=  
    !! All alternatives from ISO/IEC 9075-2  
    | !! All alternatives from ISO/IEC 9075-5  
    | MODULE <module name>
```

Syntax Rules

- 1) Replace SR7 If the object identified by <object name> of the <grant statement> or <revoke statement> is an SQL-invoked routine or an SQL-server module, then <privileges> shall specify EXECUTE; otherwise, EXECUTE shall not be specified.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

8.3 <sqlstate value>

Function

Specify an SQLSTATE value.

Format

```
<sqlstate value> ::=  
    SQLSTATE [ VALUE ] <character string literal>
```

Syntax Rules

- 1) Let *L* be the <character string literal> contained in <sqlstate value>.
- 2) The implicit or explicit character set of *L* shall be the implementation-defined character set in which SQLSTATE parameter values are returned.
- 3) Let *V* be the character string that is the value of
 TRIM (BOTH ' ' FROM *L*)
- 4) *V* shall comprise either:
 - a) Five characters of which the first two have the form of a standard-defined class value and the last three have the form of a standard-defined subclass value.
 - b) Five characters of which the first two have the form of a standard-defined class value and the last three have the form of an implementation-defined subclass value.
 - c) Five characters of which the first two have the form of an implementation-defined class value and the last three have the form of either a standard-defined subclass value or an implementation-defined subclass value.
- 5) *V* shall not be the SQLSTATE value for the condition *successful completion*.
- 6) The SQLSTATE value defined by the <sqlstate value> is *V*.

Access Rules

None.

General Rules

None.