BRITISH STANDARD

BS ISO/IEC 9075-5:1999

Incorporating Corrigenda Nos. 1 & 2

Information technology — Database languages — SQL —

Part 5: Host Language Bindings (SQL/Bindings)

ICS 35.060

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

National foreword

This British Standard reproduces verbatim ISO/IEC 9075-5:1999 and implements it as the UK national standard. Together with BS ISO/IEC 9075-1:1999 and BS ISO/IEC 9075-2:1999 it supersedes BS ISO/IEC 9075:1992 which is withdrawn.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 and i-3, pages ii to x, pages 1 to 243, and a back cover.

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

Sidelining in this document indicates the most recent changes by amendment.

Amendments issued since publication

Amd. No.	Date	Comments
10945 Corrigendum No. 1	May 2000	
11005 Corrigendum No. 2	July 2000	See National Foreword

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 April 2000

© BSI 07-2000

ISBN 0 580 34628 5

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

INTERNATIONAL STANDARD

ISO/IEC 9075-5

First edition
1999-12-01

Information technology — Database languages — SQL —

Part 5:
Host Language Bindings (SQL/Bindings)

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

Partie 5: Liants de langage d'hôte (SQL/Liants)

Reference number
ISO/IEC 9075-5:1999(E)

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Contents

Page

Foreword . . . vii

Introduction . . . ix

1 Scope . . . 1

2 Normative references . . . 3

3 Definitions, notations, and conventions . . . 5
3.1 Definitions . . . 5
3.2 Notation . . . 5
3.3 Conventions . . . 5
3.3.1 Use of terms . . . 5
3.3.1.1 Other terms . . . 5
3.3.2 Relationships to other parts of ISO/IEC 9075 . . . 5
3.3.2.1 Clause, Subclause, and Table relationships . . . 5
3.4 Object identifier for Database Language SQL . . . 12

4 Concepts . . . 15
4.1 Catalogs . . . 15
4.2 SQL-client modules . . . 15
4.3 SQL-invoked routines . . . 15
4.4 Locators . . . 15
4.5 Cursors . . . 15
4.6 SQL-statements . . . 16
4.6.1 Classes of SQL-statements . . . 16
4.6.2 SQL-statements classified by function . . . 16
4.6.3 SQL-statements and transaction states . . . 18
4.6.4 Embeddable SQL-statements . . . 19
4.6.5 Preparable and immediately executable SQL-statements . . . 20
4.6.6 Directly executable SQL-statements . . . 22
4.7 Standard programming languages . . . 23
4.8 Embedded syntax . . . 24
4.9 SQL dynamic statements . . . 24
4.10 Direct invocation of SQL . . . 27
4.11 Privileges and roles . . . 27
4.12 SQL-transactions . . . 27

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

4.13 SQL-connections . . . 27
4.14 SQL-sessions . . . 28
4.15 Client-server operation . . . 29

5 Lexical elements . . . 31
5.1 <token> and <separator> . . . 31
5.2 <literal> . . . 33
5.3 Names and identifiers . . . 34

6 Scalar expressions . . . 37
6.1 <value specification> and <target specification> . . . 37
6.2 <column reference> . . . 39
6.3 <interval value expression> . . . 40

7 Query expressions . . . 41
7.1 <table reference> . . . 41
7.2 <query specification> . . . 42

8 Additional common elements . . . 43
8.1 <routine invocation> . . . 43

9 Data assignment rules and routine determination . . . 45
9.1 Retrieval assignment . . . 45
9.2 Store assignment . . . 46
9.3 Data types of results of aggregations . . . 47

10 Schema definition and manipulation . . . 49
10.1 <check constraint definition> . . . 49
10.2 <view definition> . . . 50
10.3 <assertion definition> . . . 51
10.4 <trigger definition> . . . 52
10.5 <SQL-invoked routine> . . . 53

11 SQL-client modules . . . 55
11.1 <SQL-client module definition> . . . 55
11.2 Calls to an <externally-invoked procedure> . . . 57
11.3 <SQL procedure statement> . . . 59

12 Data manipulation . . . 61
12.1 <select statement: single row> . . . 61
12.2 <free locator statement> . . . 62

13 Transaction management . . . 63
13.1 <commit statement> . . . 63

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

14 Session management 65
14.1 <set catalog statement> 65
14.2 <set schema statement> 66
14.3 <set names statement> 67
14.4 <set path statement> 68
14.5 <set transform group statement> 69

15 Dynamic SQL 71
15.1 Description of SQL descriptor areas 71
15.2 <allocate descriptor statement> 80
15.3 <deallocate descriptor statement> 82
15.4 <get descriptor statement> 83
15.5 <set descriptor statement> 86
15.6 <prepare statement> 91
15.7 <deallocate prepared statement> 100
15.8 <describe statement> 101
15.9 <input using clause> 107
15.10 <output using clause> 111
15.11 <execute statement> 116
15.12 <execute immediate statement> 118
15.13 <dynamic declare cursor> 119
15.14 <allocate cursor statement> 121
15.15 <dynamic open statement> 123
15.16 <dynamic fetch statement> 124
15.17 <dynamic single row select statement> 125
15.18 <dynamic close statement> 126
15.19 <dynamic delete statement: positioned> 127
15.20 <dynamic update statement: positioned> 129
15.21 <preparable dynamic delete statement: positioned> 131
15.22 <preparable dynamic update statement: positioned> 132

16 Embedded SQL 133
16.1 <embedded SQL host program> 133
16.2 <embedded exception declaration> 145
16.3 <embedded SQL Ada program> 149
16.4 <embedded SQL C program> 154
16.5 <embedded SQL COBOL program> 161
16.6 <embedded SQL Fortran program> 167
16.7 <embedded SQL MUMPS program> 172
16.8 <embedded SQL Pascal program> 176
16.9 <embedded SQL PL/I program> 181

17 Direct invocation of SQL 187
17.1 <direct SQL statement> 187
17.2 <direct select statement: multiple rows> 191

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

18 Diagnostics management 193
18.1 <get diagnostics statement> 193

19 Definition Schema 197
19.1 SQL_LANGUAGES base table 197

20 Status codes 199
20.1 SQLSTATE 199

21 Conformance 201
21.1 General conformance requirements 201
21.2 Claims of conformance 201

Annex A SQL Conformance Summary 203

Annex B Implementation-defined elements 217

Annex C Implementation-dependent elements 221

Annex D Deprecated features 223

Annex E Incompatibilities with ISO/IEC 9075:1992 225

Annex F SQL feature and package taxonomy 227

Index 229

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

TABLES

Tables **Page**

1 Clause, Subclause, and Table relationships 5
2 Data types of <key word>s used in the header of SQL descriptor areas 75
3 Data types of <key word>s used in SQL item descriptor areas 75
4 Codes used for SQL data types in Dynamic SQL 77
5 Codes associated with datetime data types in Dynamic SQL 78
6 Codes used for <interval qualifier>s in Dynamic SQL 78
7 Codes used for input/output SQL parameter modes in Dynamic SQL 79
8 <identifier>s for use with <get diagnostics statement> 193
9 SQL-statement codes 194
10 SQLSTATE class and subclass values 199
11 Implied feature relationships 201
12 SQL/Bindings feature taxonomy and definition for Core SQL 228
13 SQL/Bindings feature taxonomy for features outside Core SQL 228

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

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 9075-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

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.

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

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 SQL modules.

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

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

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

8) Clause 8, “Additional common elements”, defines additional language elements that are used in various parts of the language.

9) Clause 9, “Data assignment rules and routine determination”, specifies the rules for assignments that retrieve data from or store data into SQL-data, and formation rules for set operations.

10) Clause 10, “Schema definition and manipulation”, defines facilities for creating and managing a schema.

11) Clause 11, “SQL-client modules”, defines modules and procedures.

12) Clause 12, “Data manipulation”, defines the data manipulation statements.

13) Clause 13, “Transaction management”, defines the SQL-transaction management statements.

14) Clause 14, “Session management”, defines the SQL-session management statements.

15) Clause 15, “Dynamic SQL”, defines the SQL dynamic statements.

16) Clause 16, “Embedded SQL”, defines the host language embeddings.

17) Clause 17, “Direct invocation of SQL”, defines direct invocation of SQL language.

18) Clause 18, “Diagnostics management”, defines the diagnostics management facilities.

19) Clause 19, “Definition Schema”, defines base tables on which the viewed tables containing schema information depend.

20) Clause 20, “Status codes”, defines values that identify the status of the execution of SQL-statements and the mechanisms by which those values are returned.

21) Clause 21, “Conformance”, defines the criteria for conformance to this part of ISO/IEC 9075.

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

22) Annex A, “SQL Conformance Summary”, is an informative Annex. It summarizes the conformance requirements of the SQL language.

23) 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.

24) 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.

25) 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.

26) Annex E, “Incompatibilities with ISO/IEC 9075:1992”, is an informative Annex. It lists the incompatibilities between this version of ISO/IEC 9075 and ISO/IEC 9075:1992.

27) Annex F, “SQL feature and package 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 21, “Conformance”, Subclauses begin a new page. Any resulting blank space is not significant.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Information technology — Database languages — SQL —

Part 5:
Host Language Bindings (SQL/Bindings)

1 Scope

This part of ISO/IEC 9075 specifies:

— Syntax for embedding SQL-statements in a compilation unit that otherwise conforms to the standard for a particular programming language (host language).

— How an equivalent compilation unit may be derived that conforms to the particular programming language standard. In that equivalent compilation unit, each embedded SQL-statement has been replaced by one or more statements in the host language, some of which invoke an SQL externally-invoked procedure that, when executed, has an effect equivalent to executing the SQL-statement.

— Syntax for direct invocation of SQL-statements.

— SQL language to support dynamic preparation and execution of SQL-statements.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Documents
Document Sharing Hub

2 Host Language Bindings (SQL/Bindings)

Full document details: BS ISO/IEC 9075-5:1999

2 Normative references

Insert this paragraph 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 1539-1:1997, *Information technology — Programming languages — Fortran — Part 1: Base language.*

ISO 1989:1985, *Programming languages — COBOL.*
(Endorsement of ANSI X3.23-1985).

ISO 6160:1979, *Programming languages — PL/I*
(Endorsement of ANSI X3.53-1976).

ISO/IEC 7185:1990, *Information technology — Programming languages—Pascal.*

ISO 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 9075-3:1995, *Information technology — Database languages — SQL — Part 3: Call-Level Interface (SQL/CLI).*

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

ISO/IEC 9899:1990, *Programming languages — C.*

ISO/IEC 9899:1990/Amendment 1:1995, *Amendment to ISO/IEC 9899:1990 — C Integrity.*

ISO/IEC 10206:1991, *Information technology — Programming languages—Extended Pascal.*

ISO/IEC 11756:1992, *Information technology—Programming languages — MUMPS.*

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

4 Host Language Bindings (SQL/Bindings)

Full document details: BS ISO/IEC 9075-5:1999

3 Definitions, notations, and conventions

3.1 Definitions

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 Notation

Insert this paragraph The notation used in this part of ISO/IEC 9075 is defined in ISO/IEC 9075-1.

3.3 Conventions

Insert this paragraph The conventions used in this part of ISO/IEC 9075 is defined in ISO/IEC 9075-1.

3.3.1 Use of terms

3.3.1.1 Other terms

Insert this paragraph An SQL-statement *S1* is said to be executed as a *direct result of executing an SQL-statement* if *S1* is the value of an <SQL statement variable> referenced by an <execute immediate statement> contained in an <externally-invoked procedure> that has been executed, or if *S1* was the value of the <SQL statement variable> that was associated with an <SQL statement name> by a <prepare statement> and that same <SQL statement name> is referenced by an <execute statement> contained in an <externally-invoked procedure> that has been executed.

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

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

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 3.1, “Definitions”	Subclause 3.1, "Definitions"	ISO/IEC 9075-1
Subclause 3.2, “Notation”	Subclause 6.1, "Notation"	ISO/IEC 9075-1
Subclause 3.3, “Conventions”	Subclause 6.2, "Conventions"	ISO/IEC 9075-1
Subclause 3.3.1, “Use of terms”	Subclause 6.2.3, "Use of terms"	ISO/IEC 9075-1
Subclause 3.3.1.1, “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, “Catalogs”	Subclause 4.2.6.1, "Catalogs"	ISO/IEC 9075-1
Subclause 4.2, “SQL-client modules”	Subclause 4.21, "SQL-client modules"	ISO/IEC 9075-2
Subclause 4.3, “SQL-invoked routines”	Subclause 4.23, "SQL-invoked routines"	ISO/IEC 9075-2
Subclause 4.4, “Locators”	Subclause 4.26.4, "Locators"	ISO/IEC 9075-2
Subclause 4.5, “Cursors”	Subclause 4.29, "Cursors"	ISO/IEC 9075-2
Subclause 4.6, “SQL-statements”	Subclause 4.30, "SQL-statements"	ISO/IEC 9075-2
Subclause 4.6.1, “Classes of SQL-statements”	Subclause 4.30.1, "Classes of SQL-statements"	ISO/IEC 9075-2
Subclause 4.6.2, “SQL-statements classified by function”	Subclause 4.30.2, "SQL-statements classified by function"	ISO/IEC 9075-2
Subclause 4.6.3, “SQL-statements and transaction states”	Subclause 4.30.3, "SQL-statements and transaction states"	ISO/IEC 9075-2
Subclause 4.6.4, “Embeddable SQL-statements”	*none*	*none*
Subclause 4.6.5, “Preparable and immediately executable SQL-statements”	*none*	*none*
Subclause 4.6.6, “Directly executable SQL-statements”	*none*	*none*
Subclause 4.7, “Standard programming languages”	Subclause 4.28, "Standard programming languages"	ISO/IEC 9075-2
Subclause 4.8, “Embedded syntax”	*none*	*none*

Documents
Document Sharing Hub

6 Host Language Bindings (SQL/Bindings)

Full document details: BS ISO/IEC 9075-5:1999

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 4.9, “SQL dynamic statements”	*none*	*none*
Subclause 4.10, “Direct invocation of SQL”	*none*	*none*
Subclause 4.11, “Privileges and roles”	Subclause 4.31, "Basic security model"	ISO/IEC 9075-2
Subclause 4.12, “SQL-transactions”	Subclause 4.32, "SQL-transactions"	ISO/IEC 9075-2
Subclause 4.13, “SQL-connections”	Subclause 4.33, "SQL-connections"	ISO/IEC 9075-2
Subclause 4.14, “SQL-sessions”	Subclause 4.34, "SQL-sessions"	ISO/IEC 9075-2
Subclause 4.15, “Client-server operation”	Subclause 4.36, "Client-server operation"	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, “<literal>”	Subclause 5.3, "<literal>"	ISO/IEC 9075-2
Subclause 5.3, “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
Subclause 6.1, “<value specification> and <target specification>”	Subclause 6.3, "<value specification> and <target specification>"	ISO/IEC 9075-2
Subclause 7.1, “<table reference>”	Subclause 7.6, "<table reference>"	ISO/IEC 9075-2
Subclause 6.2, “<column reference>”	Subclause 6.6, "<column reference>"	ISO/IEC 9075-2
Subclause 6.3, “<interval value expression>”	Subclause 6.29, "<interval value expression>"	ISO/IEC 9075-2
Clause 7, “Query expressions”	Clause 7, "Query expressions"	ISO/IEC 9075-2
Subclause 7.2, “<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
Clause 9, “Data assignment rules and routine determination”	Clause 9, "Data assignment rules and routine determination"	ISO/IEC 9075-2
Subclause 9.1, “Retrieval assignment”	Subclause 9.1, "Retrieval assignment"	ISO/IEC 9075-2
Subclause 9.2, “Store assignment”	Subclause 9.2, "Store assignment"	ISO/IEC 9075-2

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

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
Clause 10, “Schema definition and manipulation”	Clause 11, "Schema definition and manipulation"	ISO/IEC 9075-2
Subclause 10.1, “<check constraint definition>”	Subclause 11.9, "<check constraint definition>"	ISO/IEC 9075-2
Subclause 10.2, “<view definition>”	Subclause 11.21, "<view definition>"	ISO/IEC 9075-2
Subclause 10.3, “<assertion definition>”	Subclause 11.36, "<assertion definition>"	ISO/IEC 9075-2
Subclause 10.4, “<trigger definition>”	Subclause 11.38, "<trigger definition>"	ISO/IEC 9075-2
Subclause 10.5, “<SQL-invoked routine>”	Subclause 11.49, "<SQL-invoked routine>"	ISO/IEC 9075-2
Clause 11, “SQL-client modules”	Clause 13, "SQL-client modules"	ISO/IEC 9075-2
Subclause 11.1, “<SQL-client module definition>”	Subclause 13.1, "<SQL-client module definition>"	ISO/IEC 9075-2
Subclause 11.2, “Calls to an <externally-invoked procedure>”	Subclause 13.4, "Calls to an <externally-invoked procedure>"	ISO/IEC 9075-2
Subclause 11.3, “<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, “<select statement: single row>”	Subclause 14.5, "<select statement: single row>"	ISO/IEC 9075-2
Subclause 12.2, “<free locator statement>”	Subclause 14.12, "<free locator statement>"	ISO/IEC 9075-2
Clause 13, “Transaction management”	Clause 16, "Transaction management"	ISO/IEC 9075-2
Subclause 13.1, “<commit statement>”	Subclause 16.6, "<commit statement>"	ISO/IEC 9075-2
Clause 14, “Session management”	Clause 18, "Session management"	ISO/IEC 9075-2
Subclause 14.1, “<set catalog statement>”	*none*	*none*
Subclause 14.2, “<set schema statement>”	*none*	*none*
Subclause 14.3, “<set names statement>”	*none*	*none*
Subclause 14.4, “<set path statement>”	*none*	*none*
Subclause 14.5, “<set transform group statement>”	*none*	*none*

8 Host Language Bindings (SQL/Bindings)

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

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 15, "Dynamic SQL"	*none*	*none*
Subclause 15.1, "Description of SQL descriptor areas"	*none*	*none*
Subclause 15.2, "<allocate descriptor statement>"	*none*	*none*
Subclause 15.3, "<deallocate descriptor statement>"	*none*	*none*
Subclause 15.4, "<get descriptor statement>"	*none*	*none*
Subclause 15.5, "<set descriptor statement>"	*none*	*none*
Subclause 15.6, "<prepare statement>"	*none*	*none*
Subclause 15.7, "<deallocate prepared statement>"	*none*	*none*
Subclause 15.8, "<describe statement>"	*none*	*none*
Subclause 15.9, "<input using clause>"	*none*	*none*
Subclause 15.10, "<output using clause>"	*none*	*none*
Subclause 15.11, "<execute statement>"	*none*	*none*
Subclause 15.12, "<execute immediate statement>"	*none*	*none*
Subclause 15.13, "<dynamic declare cursor>"	*none*	*none*
Subclause 15.14, "<allocate cursor statement>"	*none*	*none*
Subclause 15.15, "<dynamic open statement>"	*none*	*none*
Subclause 15.16, "<dynamic fetch statement>"	*none*	*none*
Subclause 15.17, "<dynamic single row select statement>"	*none*	*none*
Subclause 15.18, "<dynamic close statement>"	*none*	*none*
Subclause 15.19, "<dynamic delete statement: positioned>"	*none*	*none*

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

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 15.20, “<dynamic update statement: positioned>”	*none*	*none*
Subclause 15.21, “<preparable dynamic delete statement: positioned>”	*none*	*none*
Subclause 15.22, “<preparable dynamic update statement: positioned>”	*none*	*none*
Clause 16, “Embedded SQL”	*none*	*none*
Subclause 16.1, “<embedded SQL host program>”	*none*	*none*
Subclause 16.2, “<embedded exception declaration>”	*none*	*none*
Subclause 16.3, “<embedded SQL Ada program>”	*none*	*none*
Subclause 16.4, “<embedded SQL C program>”	*none*	*none*
Subclause 16.5, “<embedded SQL COBOL program>”	*none*	*none*
Subclause 16.6, “<embedded SQL Fortran program>”	*none*	*none*
Subclause 16.7, “<embedded SQL MUMPS program>”	*none*	*none*
Subclause 16.8, “<embedded SQL Pascal program>”	*none*	*none*
Subclause 16.9, “<embedded SQL PL/I program>”	*none*	*none*
Clause 17, “Direct invocation of SQL”	*none*	*none*
Subclause 17.1, “<direct SQL statement>”	*none*	*none*
Subclause 17.2, “<direct select statement: multiple rows>”	*none*	*none*
Clause 18, “Diagnostics management”	Clause 19, "Diagnostics management"	ISO/IEC 9075-2
Subclause 18.1, “<get diagnostics statement>”	Subclause 19.1, "<get diagnostics statement>"	ISO/IEC 9075-2
Clause 19, “Definition Schema”	Clause 21, "Definition Schema"	ISO/IEC 9075-2
Subclause 19.1, “SQL_LANGUAGES base table”	Subclause 21.37, "SQL_LANGUAGES base table"	ISO/IEC 9075-2

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

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 20, “Status codes”	Clause 22, "Status codes"	ISO/IEC 9075-2
Subclause 20.1, “SQLSTATE”	Subclause 22.1, "SQLSTATE"	ISO/IEC 9075-2
Clause 21, “Conformance”	Clause 23, "Conformance"	ISO/IEC 9075-2
Subclause 21.2, “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 and package taxonomy”	Appendix F, "SQL feature and package taxonomy"	ISO/IEC 9075-2
Table 1, “Clause, Subclause, and Table relationships”	*none*	*none*
Table 2, “Data types of <key word>s used in the header of SQL descriptor areas”	*none*	*none*
Table 3, “Data types of <key word>s used in SQL item descriptor areas”	*none*	*none*
Table 4, “Codes used for SQL data types in Dynamic SQL”	*none*	*none*
Table 5, “Codes associated with datetime data types in Dynamic SQL”	*none*	*none*
Table 6, “Codes used for <interval qualifier>s in Dynamic SQL”	*none*	*none*
Table 7, “Codes used for input/output SQL parameter modes in Dynamic SQL”	*none*	*none*
Table 8, “<identifier>s for use with <get diagnostics statement>”	Table 25, "<identifier>s for use with <get diagnostics statement>"	ISO/IEC 9075-2
Table 9, “SQL-statement codes”	Table 26, "SQL-statement codes"	ISO/IEC 9075-2
Table 10, “SQLSTATE class and subclass values”	Table 27, "SQLSTATE class and subclass values"	ISO/IEC 9075-2

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

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 12, “SQL/Bindings feature taxonomy and definition for Core SQL”	Table 31, "SQL/Foundation feature taxonomy and definition for Core SQL"	ISO/IEC 9075-2
Table 13, “SQL/Bindings feature taxonomy for features outside Core SQL”	Table 32, "SQL/Foundation feature taxonomy for features outside Core SQL"	ISO/IEC 9075-2

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 5 yes> ::=
      <Part 5 conformance>
      <Part 5 direct> <Part 5 embedded>

<Part 5 conformance> ::=
      5 | sqlbindings199x <left paren> 5 <right paren>

<Part 5 direct> ::=
        <Part 5 direct yes>
      | <Part 5 direct no>

<Part 5 direct yes> ::=
      1 | directyes <left paren> 1 <right paren>

<Part 5 direct no> ::=
      0 | directno <left paren> 0 <right paren>

<Part 5 embedded> ::=
        <Part 5 embedded no>
      | <Part 5 embedded languages>...

<Part 5 embedded no> ::=
      0 | embeddedno <left paren> 0 <right paren>

<Part 5 embedded languages> ::=
        <Part 5 embedded Ada>
      | <Part 5 embedded C>
      | <Part 5 embedded COBOL>
      | <Part 5 embedded Fortran>
      | <Part 5 embedded MUMPS>
      | <Part 5 embedded Pascal>
      | <Part 5 embedded PL/I>

<Part 5 embedded Ada> ::=
      1 | embeddedAda <left paren> 1 <right paren>

<Part 5 embedded C> ::=
```


Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

```
       2 | embeddedC <left paren> 2 <right paren>

<Part 5 embedded COBOL> ::=
       3 | embeddedCOBOL <left paren> 3 <right paren>

<Part 5 embedded Fortran> ::=
       4 | embeddedFortran <left paren> 4 <right paren>

<Part 5 embedded MUMPS> ::=
       5 | embeddedMUMPS <left paren> 5 <right paren>

<Part 5 embedded Pascal> ::=
       6 | embeddedPascal <left paren> 6 <right paren>

<Part 5 embedded PL/I> ::=
       7 | embeddedPLI <left paren> 7 <right paren>
```

Syntax Rules

1) Specification of <Part 5 Yes> implies that conformance to this part of ISO/IEC 9075 is claimed.

2) Specification of <Part 5 direct yes> implies that conformance to <direct SQL statement> is claimed.

3) Specification of <Part 5 direct no> implies that conformance to <direct SQL statement> is not claimed.

4) Specification of <Part 5 embedded *language*> implies that conformance to <embedded SQL host program> for the specified *language* is claimed.

5) Specification of <Part 5 embedded no> implies that conformance to <embedded SQL host program> is not claimed.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

14 Host Language Bindings (SQL/Bindings)

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

4 Concepts

4.1 Catalogs

Insert this paragraph The default catalog name for <preparable statement>s that are dynamically prepared in the current SQL-session through the execution of <prepare statement>s and <execute immediate statement>s is initially implementation-defined but may be changed by the use of <set catalog statement>s.

4.2 SQL-client modules

Insert this paragraph SQL-session modules are implicitly-created modules for prepared SQL-statements (see Subclause 4.34, "SQL-sessions", in ISO/IEC 9075-2).

Insert this paragraph Each <SQL-client module definition> *M* that is associated with an SQL-session, other than an SQL-session module, may have associated with it a *shadow module M1* that has an implementation-dependent name not equivalent to the <module name> of any other <SQL-client module definition> in the same SQL-session and whose <module authorization clause> specifies "SCHEMA *SN*", where *SN* is the explicit or implicit <schema name> of the <module authorization clause> of *M*. The <language clause>, the SQL-path, if specified, and the <module character set specification> of *M1* are identical to the corresponding characteristic of *M*. When *M* contains a <module authorization clause> that specifies "FOR STATIC ONLY", this shadow module effectively contains one <externally-invoked procedure> for each SQL-statement prepared by <prepared statement>s or <execute immediate statement>s contained in *M*.

4.3 SQL-invoked routines

Augment the 6th paragraph The <SQL procedure statement> forming the <routine body> must exclude <SQL dynamic statement>.

4.4 Locators

Augment the 1st paragraph A host variable may be specified to be a *locator* by specifying AS LOCATOR.

Augment the 1st paragraph A host variable specified as a locator may be further specified to be a *holdable locator*.

4.5 Cursors

Insert this paragraph A cursor is also specified by a <dynamic declare cursor> or an <allocate cursor statement>. A cursor specified by a <dynamic declare cursor> is a *declared dynamic cursor*. A cursor specified by an <allocate cursor statement> is an *extended dynamic cursor*. A *dynamic cursor* is either a declared dynamic cursor or an extended dynamic cursor.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

4.5 Cursors

Insert this paragraph For every <dynamic declare cursor> in an <SQL-client module definition>, a cursor is effectively created when an SQL-transaction (see Subclause 4.32, "SQL-transactions", in ISO/IEC 9075-2) referencing the <SQL-client module definition> is initiated. An extended dynamic cursor is also effectively created when an <allocate cursor statement> is executed within an SQL-session and destroyed when that SQL-session is terminated.

Insert this paragraph In addition, a dynamic cursor is destroyed when a <deallocate prepared statement> is executed that deallocates the prepared statement on which the dynamic cursor is based.

Insert this paragraph A cursor is also placed in the open state by a <dynamic open statement> and returned to the closed state by a <dynamic close statement>.

Insert this paragraph A <dynamic fetch statement> positions an open cursor on a specified row of the cursor's ordering and retrieves the values of the columns of that row. A <dynamic update statement: positioned> updates the current row of the cursor. A <dynamic delete statement: positioned> deletes the current row of the cursor.

4.6 SQL-statements

4.6.1 Classes of SQL-statements

Insert this paragraph Most SQL-statements can be prepared for execution and executed in additional ways. These are:

— In an embedded SQL host program, in which case they are prepared when the embedded SQL host program is preprocessed (see Subclause 4.8, "Embedded syntax").

— Being prepared and executed by the use of SQL-dynamic statements (which are themselves executed in an SQL-client module or an embedded SQL host program—see Subclause 4.9, "SQL dynamic statements").

— Direct invocation, in which case they are effectively prepared immediately prior to execution (see Subclause 4.10, "Direct invocation of SQL").

Insert this paragraph There are at least three additional ways of classifying SQL-statements:

— According to whether or not they may be embedded.

— According to whether they may be dynamically prepared and executed.

— According to whether or not they may be directly executed.

4.6.2 SQL-statements classified by function

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

— SQL-dynamic statements

— SQL embedded exception declaration

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

— <dynamic declare cursor>

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

— <allocate cursor statement>

— <dynamic select statement>

— <dynamic open statement>

— <dynamic close statement>

— <dynamic fetch statement>

— <direct select statement: multiple rows>

— <dynamic single row select statement>

Insert this paragraph The following are additional SQL-data change statements:

— <dynamic delete statement: positioned>

— <preparable dynamic delete statement: positioned>

— <dynamic update statement: positioned>

— <preparable dynamic update statement: positioned>

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

— <set catalog statement>

— <set schema statement>

— <set names statement>

— <set path statement>

— <set transform group statement>

Insert this paragraph 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>

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

4.6 SQL-statements

Insert this paragraph The following is the SQL embedded exception declaration:

— <embedded exception declaration>

4.6.3 SQL-statements and transaction states

Insert this paragraph Whether or not an <execute immediate statement> starts a transaction depends on the content of the <SQL statement variable> referenced by the <execute immediate statement> at the time it is executed. Whether or not an <execute statement> starts a transaction depends on the content of the <SQL statement variable> referenced by the <prepare statement> at the time the prepared statement referenced by the <execute statement> was prepared. In both cases, if the content of the <SQL statement variable> was a transaction-initiating SQL-statement, then the <execute immediate statement> or <execute statement> is treated as a transaction-initiating statement; otherwise it is not treated as a transaction-initiating statement.

Insert this paragraph The following additional SQL-statements are transaction-initiating SQL-statements, *i.e.*, if there is no current transaction, and a statement of this class is executed, a transaction is initiated:

— <allocate cursor statement>

— <dynamic open statement>

— <dynamic close statement>

— <dynamic fetch statement>

— <direct select statement: multiple rows>

— <dynamic single row select statement>

— <dynamic delete statement: positioned>

— <preparable dynamic delete statement: positioned>

— <dynamic update statement: positioned>

— <preparable dynamic update statement: positioned>

— The following SQL-dynamic statements:

- <describe input statement>
- <describe output statement>
- <allocate descriptor statement>
- <deallocate descriptor statement>
- <get descriptor statement>
- <set descriptor statement>
- <prepare statement>
- <deallocate prepared statement>

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Insert this paragraph The following additional SQL-statements are not transaction-initiating SQL-statements, *i.e.*, if there is no current transaction, and a statement of this class is executed, no transaction is initiated.

— SQL embedded exception declarations

— The following SQL-data statements:

- <dynamic declare cursor>
- <dynamic select statement>

4.6.4 Embeddable SQL-statements

The following SQL-statements are embeddable in an embedded SQL host program, and may be the <SQL procedure statement> in an <externally-invoked procedure> in an <SQL-client module definition>:

— All SQL-schema statements

— All SQL-transaction statements

— All SQL-connection statements

— All SQL-session statements

— All SQL-dynamic statements

— All SQL-diagnostics statements

— The following SQL-data statements:

- <allocate cursor statement>
- <open statement>
- <dynamic open statement>
- <close statement>
- <dynamic close statement>
- <fetch statement>
- <dynamic fetch statement>
- <select statement: single row>
- <insert statement>
- <delete statement: searched>
- <delete statement: positioned>
- <dynamic delete statement: positioned>
- <update statement: searched>

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

4.6 SQL-statements

- <update statement: positioned>
- <dynamic update statement: positioned>
- <free locator statement>

— The following SQL-control statements:

- <call statement>

The following SQL-statements are embeddable in an embedded SQL host program, and may occur in an <SQL-client module definition>, though not in an <externally-invoked procedure>:

— <temporary table declaration>

— <declare cursor>

— <dynamic declare cursor>

The following SQL-statements are embeddable in an embedded SQL host program, but may not occur in an <SQL-client module definition>:

— SQL embedded exception declarations

Consequently, the following SQL-data statements are not embeddable in an embedded SQL host program, nor may they occur in an <SQL-client module definition>, nor be the <SQL procedure statement> in an <externally-invoked procedure> in an <SQL-client module definition>:

— <dynamic select statement>

— <dynamic single row select statement>

— <direct select statement: multiple rows>

— <preparable dynamic delete statement: positioned>

— <preparable dynamic update statement: positioned>

4.6.5 Preparable and immediately executable SQL-statements

The following SQL-statements are preparable:

— All SQL-schema statements

— All SQL-transaction statements

— All SQL-session statements

— The following SQL-data statements:

- <delete statement: searched>
- <dynamic select statement>
- <dynamic single row select statement>
- <insert statement>

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

- <update statement: searched>
- <preparable dynamic delete statement: positioned>
- <preparable dynamic update statement: positioned>
- <preparable implementation-defined statement>
- <free locator statement>

— The following SQL-control statements:

- <call statement>

Consequently, the following SQL-statements are not preparable:

— All SQL-connection statements

— All SQL-dynamic statements

— All SQL-diagnostics statements

— SQL embedded exception declarations

— The following SQL-data statements:

- <allocate cursor statement>
- <open statement>
- <dynamic open statement>
- <close statement>
- <dynamic close statement>
- <fetch statement>
- <dynamic fetch statement>
- <select statement: single row>
- <delete statement: positioned>
- <dynamic delete statement: positioned>
- <update statement: positioned>
- <dynamic update statement: positioned>
- <direct select statement: multiple rows>
- <temporary table declaration>
- <declare cursor>
- <dynamic declare cursor>

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

4.6 SQL-statements

Any preparable SQL-statement can be executed immediately, with the exception of:

— <dynamic select statement>

— <dynamic single row select statement>

4.6.6 Directly executable SQL-statements

The following SQL-statements may be executed directly:

— All SQL-schema statements

— All SQL-transaction statements

— All SQL-connection statements

— All SQL-session statements

— The following SQL-data statements:

- <temporary table declaration>
- <direct select statement: multiple rows>
- <insert statement>
- <delete statement: searched>
- <update statement: searched>

Consequently, the following SQL-statements may not be executed directly:

— All SQL-dynamic statements

— All SQL-diagnostics statements

— SQL embedded exception declarations

— The following SQL-data statements:

- <declare cursor>
- <dynamic declare cursor>
- <allocate cursor statement>
- <open statement>
- <dynamic open statement>
- <close statement>
- <dynamic close statement>
- <fetch statement>
- <dynamic fetch statement>

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

- <select statement: single row>
- <dynamic select statement>
- <dynamic single row select statement>
- <delete statement: positioned>
- <dynamic delete statement: positioned>
- <preparable dynamic delete statement: positioned>
- <update statement: positioned>
- <dynamic update statement: positioned>
- <preparable dynamic update statement: positioned>

— <free locator statement>

— <hold locator statement>

4.7 Standard programming languages

Insert this paragraph This part of ISO/IEC 9075 specifies a mechanism whereby SQL language may be embedded in programs that otherwise conform to any of the same specified programming language standards.

NOTE 1 – In this part of ISO/IEC 9075, for the purposes of interfacing with programming languages, the data types DATE, TIME, TIMESTAMP, and INTERVAL must be converted to or from character strings in those programming languages by means of a <cast specification>. It is anticipated that future evolution of programming language standards will support data types corresponding to these four SQL data types; this standard will then be amended to reflect the availability of those corresponding data types.

The data types CHARACTER, CHARACTER VARYING, and CHARACTER LARGE OBJECT are also mapped to character strings in the programming languages. However, because the facilities available in the programming languages do not provide the same capabilities as those available in SQL, there must be agreement between the host program and SQL regarding the specific format of the character data being exchanged. Specific syntax for this agreement is provided in this part of ISO/IEC 9075. For standard programming languages C, COBOL, Fortran, and Pascal, bit strings are mapped to character variables in the host language in a manner described in Subclause 16.1, “<embedded SQL host program>”. For standard programming languages Ada and PL/I, bit string variables are directly supported.

For standard programming languages C and COBOL, BOOLEAN values are mapped to integer variables in the host language. For standard programming languages Ada, Fortran, Pascal, and PL/I, BOOLEAN variables are directly supported.

For the purposes of interfacing with programming languages, the data type ARRAY must be converted to a locator (see Subclause 4.26.4, "Locators", in ISO/IEC 9075-2).

For the purposes of interfacing with programming languages, user-defined types must be handled with a locator (see Subclause 4.26.4, "Locators", in ISO/IEC 9075-2) or transformed to another SQL data type that has a defined mapping to the host language (see Subclause 4.8.5, "Transforms for user-defined types", in ISO/IEC 9075-2).

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

4.8 Embedded syntax

An <embedded SQL host program> (<embedded SQL Ada program>, <embedded SQL C program>, <embedded SQL COBOL program>, <embedded SQL Fortran program>, <embedded SQL MUMPS program>, <embedded SQL Pascal program>, or <embedded SQL PL/I program>) is a compilation unit that consists of programming language text and SQL text. The programming language text shall conform to the requirements of a specific standard programming language. The SQL text shall consist of one or more <embedded SQL statement>s and, optionally, one or more <embedded SQL declare section>s, as defined in this International Standard. This allows database applications to be expressed in a hybrid form in which SQL-statements are embedded directly in a compilation unit. Such a hybrid compilation unit is defined to be equivalent to a standard compilation unit in which the SQL-statements have been replaced by standard procedure or subroutine calls of <externally-invoked procedure>s in a separate SQL-client module, and in which each has been removed and the declarations contained therein have been suitably transformed into standard host-language syntax.

If an <embedded SQL host program> contains an <embedded authorization declaration>, then it shall be the first statement or declaration in the <embedded SQL host program>. The <embedded authorization declaration> is not replaced by a procedure or subroutine call of an <externally-invoked procedure>, but is removed and replaced by syntax associated with the <SQL-client module definition>'s <module authorization clause>.

An implementation may reserve a portion of the name space in the <embedded SQL host program> for the names of procedures or subroutines that are generated to replace SQL-statements and for program variables and branch labels that may be generated as required to support the calling of these procedures or subroutines; whether this reservation is made is implementation-defined. They may similarly reserve name space for the <module name> and <procedure name>s of the generated <SQL-client module definition> that may be associated with the resulting standard compilation unit. The portion of the name space to be so reserved, if any, is implementation-defined.

4.9 SQL dynamic statements

In many cases, the SQL-statement to be executed can be coded into an <SQL-client module definition> or into a compilation unit using the embedded syntax. In other cases, the SQL-statement is not known when the program is written and will be generated during program execution. An <execute immediate statement> can be used for a one-time preparation and execution of an SQL-statement. A <prepare statement> is used to prepare the generated SQL-statement for subsequent execution. A <deallocate prepared statement> is used to deallocate SQL-statements that have been prepared with a <prepare statement>. A description of the input dynamic parameters for a prepared statement can be obtained by execution of a <describe input statement>. A description of the resultant columns of a <dynamic select statement> or <dynamic single row select statement> can be obtained by execution of a <describe output statement>. A description of the output dynamic parameters of a statement that is neither a <dynamic select statement> nor a <dynamic single row select statement> can be obtained by execution of a <describe output statement>. For a statement other than a <dynamic select statement>, an <execute statement> is used to associate parameters with the prepared statement and execute it as though it had been coded when the program was written. For a <dynamic select statement>, the prepared <cursor specification> is associated with a cursor via a <dynamic declare cursor> or <allocate cursor statement>. The cursor can be opened and dynamic parameters can be associated with the cursor with a <dynamic open statement>. A <dynamic fetch statement> positions an open cursor on a specified row and retrieves the values of the columns of that row. A <dynamic close statement> closes a cursor that was opened with a <dynamic open statement>. A <dynamic delete statement: positioned> is used to delete rows through

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

a dynamic cursor. A <dynamic update statement: positioned> is used to update rows through a dynamic cursor. A <preparable dynamic delete statement: positioned> is used to delete rows through a dynamic cursor when the precise format of the statement isn't known until runtime. A <preparable dynamic update statement: positioned> is used to update rows through a dynamic cursor when the precise format of the statement isn't known until runtime.

The interface for input dynamic parameters and output dynamic parameters for a prepared statement and for the resulting values from a <dynamic fetch statement> or the execution of a prepared <dynamic single row select statement> can be either a list of dynamic parameters or embedded variables or an SQL descriptor area. An SQL descriptor area consists of one or more item descriptor areas, together with a header that includes a count of the number of those item descriptor areas. The header of an SQL descriptor area consists of the fields in Table 2, "Data types of <key word>s used in the header of SQL descriptor areas", in Subclause 15.1, "Description of SQL descriptor areas". Each item descriptor area consists of the fields specified in Table 3, "Data types of <key word>s used in SQL item descriptor areas", in Subclause 15.1, "Description of SQL descriptor areas". The SQL descriptor area is allocated and maintained by the system with the following statements: <allocate descriptor statement>, <deallocate descriptor statement>, <set descriptor statement>, and <get descriptor statement>.

An SQL descriptor area is identified by a <descriptor name>, which is a <simple value specification> whose value is an <identifier>. Two <descriptor name>s identify the same SQL descriptor area if their values, with leading and trailing <space>s removed, are equivalent according to the rules for <identifier> comparisons in Subclause 5.2, "<token> and <separator>", in ISO/IEC 9075-2.

Dynamic statements can be identified by <statement name>s or by <extended statement name>s. Similarly, dynamic cursors can be identified by <cursor name>s and by <extended cursor name>s. The non-extended names are <identifier>s. The extended names are <target specification>s whose values are <identifier>s used to identify the statement or cursor. Two extended names are equivalent if their values, with leading and trailing <space>s removed, are equivalent according to the rules for <identifier> comparison in Subclause 5.2, "<token> and <separator>", in ISO/IEC 9075-2.

An SQL descriptor area name may be defined as global or local. Similarly, an extended statement name or extended cursor name may be global or local. The scope of a global name is the SQL-session. The scope of a local name is the <SQL-client module definition> in which it appears. A reference to an entity in which one specifies a global scope is valid only if the entity was defined as global and if the reference is from the same SQL-session in which it was defined. A reference to an entity in which one specifies a local scope is valid only if the entity was defined as local and if the reference is from the same <SQL-client module definition> in which it was defined. (The scope of non-extended statement names and non-extended cursor names is always local.)

NOTE 2 – The namespace of non-extended statement names and non-extended cursor names is different from the namespace of extended statement names that are specified to be LOCAL and extended cursor names that are specified to be LOCAL, respectively.

The relationships between prepared statements and <SQL-client module definition>s are:

— Within an SQL-session, all global prepared statements (prepared statements with global statement names) belong to the SQL-session module.

— Within an SQL-session, each local prepared statement (a prepared statement with a local statement name) prepared in an <SQL-client module definition> containing a <module authorization clause> that does not specify "AUTHORIZATION <module authorization identifier>" belongs to the <SQL-client module definition> that contains the <prepare statement> or <execute immediate statement> with which it is prepared.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

4.9 SQL dynamic statements

— Within an SQL-session, each local prepared statement (a prepared statement with a local statement names) prepared in an <SQL-client module definition> containing a <module authorization clause> that contains "AUTHORIZATION <module authorization identifier>" and does not contain "FOR STATIC ONLY" belongs to the <SQL-client module definition> that contains the <prepare statement> or <execute immediate statement> with which it is prepared.

— Within an SQL-session, each local prepared statement (a prepared statement with a local statement names) prepared in an <SQL-client module definition> *M* containing a <module authorization clause> that contains "AUTHORIZATION <module authorization identifier> FOR STATIC ONLY" belongs to a *shadow module* that has an implementation-dependent name not equivalent to the <module name> of any other <SQL-client module definition> in the same SQL-session and whose <module authorization clause> specifies "SCHEMA *SN*", where *SN* is the explicit or implicit <schema name> of the <module authorization clause> of *M*.

NOTE 3 – The SQL-session module is defined in Subclause 4.34, "SQL-sessions", in ISO/IEC 9075-2.

Dynamic execution of SQL-statements can generally be accomplished in two different ways. Statements can be *prepared* for execution and then later executed one or more times; when the statement is no longer needed for execution, it can be *released* by the use of a <deallocate prepared statement>. Alternatively, a statement that is needed only once can be executed without the preparation step—it can be *executed immediately* (not all SQL-statements can be executed immediately).

Many SQL-statements can be written to use "parameters" (which are manifested in static execution of SQL-statements as host parameters in <SQL procedure statement>s contained in <externally-invoked procedure>s in <SQL-client module definition>s or as host variables in <embedded SQL statement>s contained in <embedded SQL host program>s). In SQL-statements that are executed dynamically, the parameters are called dynamic parameters (<dynamic parameter specification>s) and are represented in SQL language by a <question mark> (?).

In many situations, an application that generates an SQL-statement for dynamic execution knows in detail the required characteristics (*e.g.*, <data type>, <length>, <precision>, <scale>, *etc.*) of each of the dynamic parameters used in the statement; similarly, the application may also know in detail the characteristics of the values that will be returned by execution of the statement. However, in other cases, the application may not know this information to the required level of detail; it is possible in some cases for the application to ascertain the information from the Information Schema, but in other cases (*e.g.*, when a returned value is derived from a computation instead of simply from a column in a table, or when dynamic parameters are supplied) this information is not generally available except in the context of preparing the statement for execution.

To provide the necessary information to applications, SQL permits an application to request the SQL-server to *describe* a prepared statement. The description of a statement identifies the number of input dynamic parameters (*describe input*) and their data type information or it identifies the number of output dynamic parameters or values to be returned (*describe output*) and their data type information. The description of a statement is placed into the SQL descriptor areas already mentioned.

Many, but not all, SQL-statements can be prepared and executed dynamically.

NOTE 4 – The complete list of statements that may be dynamically prepared and executed is defined in Subclause 4.6.5, "Preparable and immediately executable SQL-statements".

Certain "set statements" (<set catalog statement>, <set schema statement>, <set names statement>, and <set path statement>) have no effect other than to set up default information (catalog name, schema name, character set, and SQL path, respectively) to be applied to other SQL-statements that are prepared or executed immediately or that are invoked directly.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Syntax errors and Access Rule violations caused by the preparation or immediate execution of <preparable statement>s are identified when the statement is prepared (by <prepare statement>) or when it is executed (by <execute statement> or <execute immediate statement>); executed (by <execute statement> or <execute immediate statement>); such violations are indicated by the raising of an exception condition.

4.10 Direct invocation of SQL

Direct invocation of SQL is a mechanism for executing direct SQL-statements, known as <direct SQL statement>s. In direct invocation of SQL, the method of invoking <direct SQL statement>s, the method of raising conditions that result from the execution of <direct SQL statement>s, the method of accessing the diagnostics information that results from the execution of <direct SQL statement>s, and the method of returning the results are implementation-defined.

4.11 Privileges and roles

Insert this paragraph For direct SQL, the SQL-session user identifier is always the current authorization identifier.

4.12 SQL-transactions

Insert this paragraph The operations comprising an SQL-transaction may also be performed by the direct invocation of SQL.

Insert this paragraph It is implementation-defined whether or not the dynamic execution of an <SQL dynamic data statement> is permitted to occur within the same SQL-transaction as the dynamic execution of an SQL-schema statement. If it does occur, then the effect on any open cursor, prepared dynamic statement, or deferred constraint is implementation-defined. There may be additional implementation-defined restrictions, requirements, and conditions. If any such restrictions, requirements, or conditions are violated, then an implementation-defined exception condition or a completion condition *warning* with an implementation-defined subclass code is raised.

Insert this paragraph Each direct invocation of SQL that executes an SQL-statement of an SQL-transaction is associated with that SQL-transaction. An SQL-transaction is initiated when no SQL-transaction is currently active by direct invocation of SQL that results in the execution of a transaction-initiating <direct SQL statement>.

4.13 SQL-connections

Insert this paragraph An SQL-connection may also be terminated by the last execution of a <direct SQL statement> through the direct invocation of SQL. The mechanism and rules by which an SQL-implementation determines whether a direct invocation of SQL is the last execution of a <direct SQL statement> are implementation-defined.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

4.14 SQL-sessions

Replace 3rd paragraph Within an SQL-session, module local temporary tables are effectively created by <temporary table declaration>s. Module local temporary tables are accessible only to invocations of <externally-invoked procedure>s in the SQL-client module or SQL-session module in which they are created. The definitions of module local temporary tables persist until the end of the SQL-session.

Insert this paragraph An *SQL-session* also spans the execution of a sequence of consecutive SQL-statements invoked by the direct invocation of SQL.

Insert this paragraph An SQL-session has an SQL-session module that is different from any other <SQL-client module definition> that exists simultaneously in the SQL-environment. The SQL-session module contains the global prepared SQL-statements that belong to the SQL-session. The SQL-session module contains a <module authorization clause> that specifies SCHEMA <schema name>, where the value of <schema name> is implementation-dependent.

Insert this paragraph An SQL-session has a default catalog name that is used to effectively qualify unqualified <schema name>s that are contained in <preparable statement>s when those statements are prepared in the current SQL-session by either an <execute immediate statement> or a <prepare statement> or are contained in <direct SQL statement>s when those statements are invoked directly. The default catalog name is initially set to an implementation-defined value but can subsequently be changed by the successful execution of a <set catalog statement> or <set schema statement>.

Insert this paragraph An SQL-session has a default unqualified schema name that is used to effectively qualify unqualified <schema qualified name>s that are contained in <preparable statement>s when those statements are prepared in the current SQL-session by either an <execute immediate statement> or a <prepare statement> or are contained in <direct SQL statement>s when those statements are invoked directly. The default unqualified schema name is initially set to an implementation-defined value but can subsequently be changed by the successful execution of a <set schema statement>.

Insert this paragraph An SQL-session has a SQL-path that is used to effectively qualify unqualified <routine name>s that are immediately contained in <routine invocation>s that are contained in <preparable statement>s when those statements are prepared in the current SQL-session by either an <execute immediate statement> or a <prepare statement> or are contained in <direct SQL statement>s when those statements are invoked directly. The SQL-path is initially set to an implementation-defined value, but can subsequently be changed by the successful execution of a <set path statement>.

Insert this paragraph The text defining the SQL-path can be referenced by using the <general value specification> CURRENT_PATH.

Insert this paragraph An SQL-session has a default transform group name and one or more user-defined type name-transform group name pairs that are used to identify the group of transform functions for every user-defined type that is referenced in <preparable statement>s when those statements are prepared in the current SQL-session by either an <execute immediate statement> or a <prepare statement> or are contained in <direct SQL statement>s when those statements are invoked directly. The transform group name for a given user-defined type name is initially set to an implementation-defined value but can subsequently be changed by the successful execution of a <set transform group statement>.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

Insert this paragraph The text defining the transform group names associated with the SQL-session can be referenced using two mechanisms: the <general value specification> "CURRENT_TRANSFORM_GROUP_FOR_TYPE <user-defined type>", which evaluates to the name of the transform group associated with the specified data type, and the <general value specification> "CURRENT_DEFAULT_TRANSFORM_GROUP", which evaluates to the name of the transform group associated with all types that have no type-specific transform group specified for them.

Insert this paragraph An SQL-session has a default character set name that is used to identify the character set in which <preparable statement>s are represented when those statements are prepared in the current SQL-session by either an <execute immediate statement> or a <prepare statement>. The default character set name is initially set to an implementation-defined value but can subsequently be changed by the successful execution of a <set names statement>.

Augment the 4th paragraph Within an SQL-session, locators are effectively created whenever a host variable that is specified as a binary large object locator, character large object locator, array locator, or user-defined type locator is assigned a value of binary large object type, character large object type, array type, or user-defined type, respectively. A host variable that is a locator may be *holdable* or *nonholdable*.

Insert this paragraph The SQL-session context also comprises:

— The SQL-session module.

— The current default catalog name.

— The current default unqualified schema name.

— The current character set name substitution value.

— The text defining the SQL-path.

— The contents of all SQL dynamic descriptor areas.

— The text defining the default transform group name.

— The text defining the user-defined type name-transform group name pair for each user-defined type explicitly set by the user.

4.15 Client-server operation

Insert this paragraph <direct SQL statement>s containing an <SQL connection statement> or an <SQL diagnostics statement> are processed by the SQL-client. Other <direct SQL statement>s are processed by the SQL-server.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

30 Host Language Bindings (SQL/Bindings)

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

5 Lexical elements

5.1 <token> and <separator>

Function

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

Format

```
<delimiter token> ::=
       !! All alternatives from ISO/IEC 9075-2
       | <double period>

<double period> ::= ..

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

       | DEGREE | DYNAMIC_FUNCTION | DYNAMIC_FUNCTION_CODE

       | RETURNED_CARDINALITY

       | STRUCTURE

       | TOP_LEVEL_COUNT

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

       | CURRENT_DEFAULT_TRANSFORM_GROUP
       | CURRENT_TRANSFORM_GROUP_FOR_TYPE

       | DYNAMIC

       | NESTING
```

Syntax Rules

None.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

5.1 <token> and <separator>

Conformance Rules

No additional Conformance Rules.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

5.2 <literal>

Function

Specify a non-null value.

Format

No additional Format items.

Syntax Rules

1) Augments SR8) Case:

a) If a <character set specification> is not specified in a <character string literal>, then the set of characters contained in the <character string literal> shall be wholly contained in the character set of the SQL-client module or SQL-session module that contains the <character string literal>.

b) Otherwise, there shall be no <separator> between the <introducer> and the <character set specification>, and the set of characters contained in the <character string literal> shall be wholly contained in the character set specified by the <character set specification>.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

Conformance Rules

No additional Conformance Rules.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 9075-5:1999

5.3 Names and identifiers

Function

Specify names.

Format

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

<statement name> ::= <identifier>

<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> ::=
       [ <scope option> ] <simple value specification>

<scope option> ::=
         GLOBAL
       | LOCAL
```

Syntax Rules

1) Insert before SR4)a) If the <local or schema qualified name> is contained, without an intervening <schema 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 SR11)a) If the <schema qualified name> is contained, without an intervening <schema 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.

3) Insert before SR12)a) If the <unqualified schema name> is contained 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 catalog name for the SQL-session is implicit.

4) Insert this SR The <simple value specification> of <extended statement name> or <extended cursor name> shall not be a <literal>.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

5) Insert this SR The declared type of the <simple value specification> of <extended statement name> shall be character string with an implementation-defined character set and shall have an octet length of 128 octets or less.

6) Insert this SR The declared type of the <simple value specification> of <extended cursor name> shall be character string with an implementation-defined character set and shall have an octet length of 128 octets or less.

7) Insert this SR The declared type of the <simple value specification> of <descriptor name> shall be character string with an implementation-defined character set and shall have an octet length of 128 octets or less.

8) Insert this SR In a <descriptor name>, <extended statement name>, or <extended cursor name>, if a <scope option> is not specified, then a <scope option> of LOCAL is implicit.

Access Rules

No additional Access Rules.

General Rules

1) Insert this GR The value of an <extended statement name> identifies a statement prepared by the execution of a <prepare statement>. If a <scope option> of GLOBAL is specified, then the scope of the <extended statement name> is the current SQL-session. If a <scope option> of LOCAL is specified or implicit, then the scope of the <extended statement name> is further restricted to the <SQL-client module definition> in which the <extended statement name> appears.

2) Insert this GR A <dynamic cursor name> identifies a cursor in an <SQL dynamic statement>.

3) Insert this GR A <descriptor name> identifies a descriptor area.

4) Insert this GR A <statement name> identifies a statement prepared by the execution of a <prepare statement>. The scope of a <statement name> is the <SQL-client module definition> in which it appears and the current SQL-session.

5) Insert this GR The value of an <extended cursor name> identifies a cursor created by the execution of an <allocate cursor statement>. If a <scope option> of GLOBAL is specified, then the scope of the <extended cursor name> is the current SQL-session. If a <scope option> of LOCAL is specified or implicit, then the scope of the <extended cursor name> is further restricted to the <SQL-client module definition> in which the <extended cursor name> appears.

6) Insert this GR A <descriptor name> identifies an SQL descriptor area created by the execution of an <allocate descriptor statement>. If a <scope option> of GLOBAL is specified, then the scope of the <descriptor name> is the current SQL-session. If a <scope option> of LOCAL is specified or implicit, then the scope of the <descriptor name> is further restricted to the <SQL-client module definition> in which the <descriptor name> appears.

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999

5.3 Names and identifiers

Conformance Rules

1) Insert this CR Without Feature B032, "Extended dynamic SQL", conforming SQL language shall not contain any <extended statement name> or <extended cursor name>.

2) Insert this CR Without Feature B031, "Basic dynamic SQL", conforming SQL language shall not contain any <SQL statement name>, <dynamic cursor name>, or <descriptor name>.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 9075-5:1999