INTERNATIONAL STANDARD

ISO/IEC 9075-1 9075-2 9075-5

First edition
1999-12-01

AMENDMENT 1
2001-03-15

Information technology — Database languages — SQL —

Part 1:
Framework (SQL/Framework)
Part 2:
Foundation (SQL/Foundation)
Part 5:
Host Language Bindings (SQL/Bindings)

AMENDMENT 1: On-Line Analytical Processing (SQL/OLAP)

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

Partie 1: Charpente (SQL/Charpente)

Partie 2: Fondations (SQL/Fondations)

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

AMENDEMENT 1: Traitement analytique en ligne (SQL/OLAP)

Reference number
ISO/IEC 9075 (parts 1, 2 and 5):1999/Amd1:2001(E)

© ISO/IEC 2001

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2001

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

Contents

Page

Foreword vii

Introduction viii

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 Syntactic containment 5
3.3.2 Relationships to other parts of ISO/IEC 9075 5
3.3.2.1 Clause, Subclause, and Table relationships 5

4 Concepts 9
4.1 Numbers 9
4.1.1 Operations involving numbers 9
4.2 Tables 10
4.2.1 Windowed tables 10
4.3 Data analysis operations (involving tables) 11
4.3.1 Group functions 11
4.3.2 Window functions 12
4.3.3 Aggregate functions 13

5 Lexical elements 17
5.1 <token> and <separator> 17
5.2 Names and identifiers 19

6 Scalar expressions 21
6.1 <set function specification> 21
6.2 <numeric value function> 23
6.3 <window function> 27
6.4 <value expression> 31

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

7 Query expressions 33
7.1 <table expression> 33
7.2 <joined table> 34
7.3 <where clause> 35
7.4 <having clause> 36
7.5 <window clause> 37
7.6 <query specification> 47

8 Additional common elements 51
8.1 <aggregate function> 51
8.2 <sort specification list> 61

9 Schema definition and manipulation 63
9.1 <drop table constraint definition> 63
9.2 <drop user-defined ordering statement> 64

10 SQL-client modules 65
10.1 Calls to an <externally-invoked procedure> 65

11 Data manipulation 67
11.1 <declare cursor> 67
11.2 <select statement: single row> 68

12 Dynamic SQL 69
12.1 <prepare statement> 69

13 Information Schema 71
13.1 Definition of SQL built-in functions 71

14 Status codes 73
14.1 SQLSTATE 73

15 Conformance 75
15.1 General conformance requirements 75

Annex A SQL conformance summary 77

Annex B Implementation-defined elements 81

Annex C Implementation-dependent elements 85

Annex D SQL feature and package taxonomy 87

Annex E SQL Packages 89
E.1 OLAP 89

Index 91

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

FIGURES

Figure **Page**

1 Illustration of WIDTH_BUCKET Semantics 9

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

TABLES

Tables **Page**

1 Clause, Subclause, and Table relationships 5
2 SQLSTATE class and subclass values 73
3 Implied feature relationships 75
4 SQL/OLAP feature taxonomy for features outside Core SQL 87
5 SQL Packages 89

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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.

Attention is drawn to the possibility that some of the elements of this Amendment may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to parts 1, 2 and 5 of ISO/IEC 9075:1999 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

Introduction

The organization of this Amendment is as follows:

1) Clause 1, “Scope”, specifies the scope of this Amendment.

2) Clause 2, “Normative references”, identifies additional standards that, through reference in this Amendment, constitute provisions of this Amendment.

3) Clause 3, “Definitions, notations, and conventions”, defines the notations and conventions used in this Amendment.

4) Clause 4, “Concepts”, presents concepts used in the definition of On-Line Analytical Processing facilities.

5) Clause 5, “Lexical elements”, defines a number of lexical elements used in the definition of On-Line Analytical Processing facilities.

6) Clause 6, “Scalar expressions”, defines a number of scalar expressions used in the definition of On-Line Analytical Processing facilities.

7) Clause 7, “Query expressions”, defines the elements of the language that produce rows and tables of data as used in On-Line Analytical Processing facilities.

8) Clause 8, “Additional common elements”, defines additional common elements used in the definition of On-Line Analytical Processing facilities.

9) Clause 9, “Schema definition and manipulation”, defines the schema definition and manipulation statements associated with the definition of On-Line Analytical Processing facilities.

10) Clause 10, “SQL-client modules”, defines SQL-client modules and externally-invoked procedures.

11) Clause 11, “Data manipulation”, defines data manipulation operations associated with On-Line Analytical Processing facilities.

12) Clause 12, “Dynamic SQL”, defines the SQL dynamic statements.

13) Clause 13, “Information Schema”, defines viewed tables that contain schema information related to On-Line Analytical Processing facilities.

14) Clause 14, “Status codes”, defines SQLSTATE values related to On-Line Analytical Processing facilities.

15) Clause 15, “Conformance”, defines the criteria for conformance to this Amendment.

16) Annex A, “SQL conformance summary”, is an informative Annex. It summarizes the conformance requirements of the SQL language.

17) Annex B, “Implementation-defined elements”, is an informative Annex. It lists those features for which the body of this Amendment 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.

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

18) Annex C, “Implementation-dependent elements”, is an informative Annex. It lists those features for which the body of this Amendment 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.

19) Annex D, “SQL feature and package taxonomy”, is an informative Annex. It identifies features of the SQL language specified in this Amendment 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.

20) Annex E, “SQL Packages”, is an informative Annex. It specifies a package of SQL language features.

In the text of this Amendment, Clauses begin a new odd-numbered page, and in Clause 5, “Lexical elements”, through Clause 15, “Conformance”, Subclauses begin a new page. Any resulting blank space is not significant.

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

Information technology — Database languages — SQL —

Part 1: Framework (SQL/Framework)
Part 2: Foundation (SQL/Foundation)
Part 5: Host Language Bindings (SQL/Bindings)

AMENDMENT 1:
On-Line Analytical Processing (SQL/OLAP)

1 Scope

This Amendment specifies the syntax and semantics of database language facilities that support on-line analytical processing.

The database language facilities that support on-line analytical processing include:

— Rank functions.

— Distribution functions.

— Inverse distribution functions (percentiles).

— Hypothetical set functions.

— Cumulative and other forms of moving aggregates.

— Variance, standard deviation, covariance, correlation, and linear regression functions.

This amendment also incidentally defines several new numeric functions.

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

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

(Blank page)

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this Amendment. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this Amendment are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9075-1:1999, *Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework).*

ISO/IEC 9075-2:1999, *Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation).*

ISO/IEC 9075-5:1999, *Information technology — Database languages — SQL — Part 5: Host Language Bindings (SQL/Bindings).*

© ISO/IEC 2001 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

(Blank page)

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

3 Definitions, notations, and conventions

3.1 Definitions

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

3.2 Notation

Insert this paragraph The syntax notation used in this Amendment is an extended version of BNF ("Backus Normal Form" or "Backus Naur Form"). This version of BNF is fully described in Subclause 3.2, "Notation", of ISO/IEC 9075-1.

3.3 Conventions

Insert this paragraph Except as otherwise specified in this Amendment the conventions used in this Amendment 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 Syntactic containment

Replace 2nd paragraph *A1 directly contains B1* if *A1* contains *B1* without an intervening <subquery>, <within group specification>, or <set function specification> that is not an <ordered set function>.

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 3.2, “Notation”	Subclause 3.2, "Notation"	ISO/IEC 9075-2

© ISO/IEC 2001 – All rights reserved

Documents Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

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 3.3, “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, “Syntactic containment”	Subclause 3.3.1.1, "Syntactic containment"	ISO/IEC 9075-2
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)*
Clause 4, “Concepts”	Clause 4, "Concepts"	ISO/IEC 9075-2
Subclause 4.1, “Numbers”	Subclause 4.5, "Numbers"	ISO/IEC 9075-2
Subclause 4.1.1, “Operations involving numbers”	Subclause 4.5.2, "Operations involving numbers"	ISO/IEC 9075-2
Subclause 4.2, “Tables”	Subclause 4.16, "Tables"	ISO/IEC 9075-2
Subclause 4.2.1, “Windowed tables”	*(none)*	*(none)*
Subclause 4.3, “Data analysis operations (involving tables)”	*(none)*	*(none)*
Subclause 4.3.1, “Group functions”	*(none)*	*(none)*
Subclause 4.3.2, “Window functions”	*(none)*	*(none)*
Subclause 4.3.3, “Aggregate functions”	*(none)*	*(none)*
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
Subclause 6.1, “<set function specification>”	Subclause 6.16, "<set function specification>"	ISO/IEC 9075-2
Subclause 6.2, “<numeric value function>”	Subclause 6.17, "<numeric value function>"	ISO/IEC 9075-2
Subclause 6.3, “<window function>”	*(none)*	*(none)*
Subclause 6.4, “<value expression>”	Subclause 6.23, "<value expression>"	ISO/IEC 9075-2
Clause 7, “Query expressions”	Clause 7, "Query expressions"	ISO/IEC 9075-2
Subclause 7.1, “<table expression>”	Subclause 7.4, "<table expression>"	ISO/IEC 9075-2
Subclause 7.2, “<joined table>”	Subclause 7.7, "<joined table>"	ISO/IEC 9075-2

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

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 7.3, “<where clause>”	Subclause 7.8, "<where clause>"	ISO/IEC 9075-2
Subclause 7.4, “<having clause>”	Subclause 7.10, "<having clause>"	ISO/IEC 9075-2
Subclause 7.5, “<window clause>”	*(none)*	*(none)*
Subclause 7.6, “<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, “<aggregate function>”	*(none)*	*(none)*
Subclause 8.2, “<sort specification list>”	*(none)*	*(none)*
Clause 9, “Schema definition and manipulation”	Clause 11, "Schema definition and manipulation"	ISO/IEC 9075-2
Subclause 9.1, “<drop table constraint definition>”	Subclause 11.19, "<drop table constraint definition>"	ISO/IEC 9075-2
Subclause 9.2, “<drop user-defined ordering statement>”	Subclause 11.55, "<drop user-defined ordering statement>"	ISO/IEC 9075-2
Clause 10, “SQL-client modules”	Clause 13, "SQL-client modules"	ISO/IEC 9075-2
Subclause 10.1, “Calls to an <externally-invoked procedure>”	Subclause 13.4, "Calls to an <externally-invoked procedure>"	ISO/IEC 9075-2
Clause 11, “Data manipulation”	Clause 14, "Data manipulation"	ISO/IEC 9075-2
Subclause 11.1, “<declare cursor>”	Subclause 14.1, "<declare cursor>"	ISO/IEC 9075-2
Subclause 11.2, “<select statement: single row>”	Subclause 14.5, "<select statement: single row>"	ISO/IEC 9075-2
Clause 12, “Dynamic SQL”	Clause 15, "Dynamic SQL"	ISO/IEC 9075-5
Subclause 12.1, “<prepare statement>”	Subclause 15.6, "<prepare statement>"	ISO/IEC 9075-5
Clause 13, “Information Schema”	Clause 20, "Information Schema"	ISO/IEC 9075-2
Subclause 13.1, “Definition of SQL built-in functions”	Subclause 20.70, "Definition of SQL built-in functions"	ISO/IEC 9075-2
Clause 14, “Status codes”	Clause 22, "Status codes"	ISO/IEC 9075-2
Subclause 14.1, “SQLSTATE”	Subclause 22.1, "SQLSTATE"	ISO/IEC 9075-2
Clause 15, “Conformance”	Clause 23, "Conformance"	ISO/IEC 9075-2
Subclause 15.1, “General conformance requirements”	Subclause 23.1, "General conformance requirements"	ISO/IEC 9075-2
Annex A, “SQL conformance summary”	Annex A, "SQL Conformance Summary"	ISO/IEC 9075-2

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

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
Annex B, “Implementation-defined elements”	Annex B, "Implementation-defined elements"	ISO/IEC 9075-2
Annex C, “Implementation-dependent elements”	Annex C, "Implementation-dependent elements"	ISO/IEC 9075-2
Annex D, “SQL feature and package taxonomy”	Annex F, "SQL feature and package taxonomy"	ISO/IEC 9075-2
Annex E, “SQL Packages”	Annex B, "SQL Packages"	ISO/IEC 9075-1
Subclause E.1, “OLAP”	*(none)*	*(none)*
Figure 1, “Illustration of WIDTH_BUCKET Semantics”	*(none)*	*(none)*
Table 1, “Clause, Subclause, and Table relationships”	*(none)*	*(none)*
Table 2, “SQLSTATE class and subclass values”	Table 27, "SQLSTATE class and subclass values"	ISO/IEC 9075-2
Table 3, “Implied feature relationships”	Table 30, "Implied feature relationships"	ISO/IEC 9075-2
Table 4, “SQL/OLAP feature taxonomy for features outside Core SQL”	Table 32, "SQL/Foundation feature taxonomy for features outside Core SQL"	ISO/IEC 9075-2
Table 5, “SQL Packages”	Table 2, "SQL Packages"	ISO/IEC 9075-1

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

4 Concepts

4.1 Numbers

4.1.1 Operations involving numbers

Insert this paragraph The following are also functions that return numbers:

— <natural logarithm> computes the natural logarithm of its argument.

— <exponential function> computes the exponential function, that is, *e*, (the base of natural logarithms) raised to the power equal to its argument.

— <power function> raises its first argument to the power of its second argument.

— <square root> computes the square root of its argument.

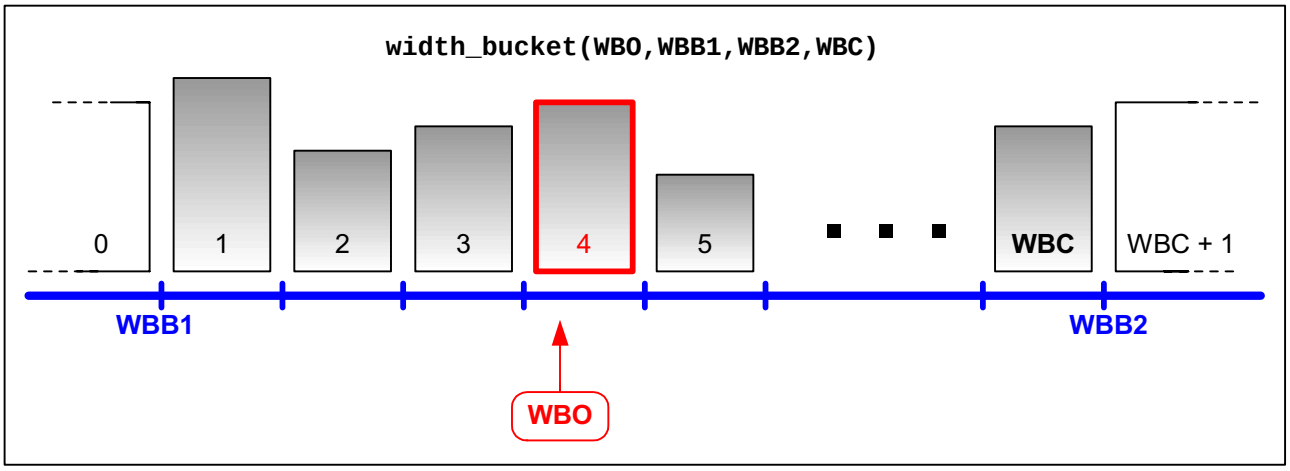
— <floor function> computes the greatest integer less than or equal to its argument.

— <ceiling function> computes the least integer greater than or equal to its argument.

— <width bucket> is a function of four arguments, returning an integer between 0 (zero) and the value of the final argument plus 1 (one), by assigning the first argument to an equi-width partitioning of the range of numbers between the second and third arguments. Values outside the range between the second and third arguments are assigned to either 0 (zero) or the value of the final argument plus 1 (one).

NOTE 2 – The semantics of <width bucket> are illustrated in Figure 1, "Illustration of WIDTH_BUCKET Semantics".

Figure 1—Illustration of WIDTH_BUCKET Semantics



© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

4.2 Tables

4.2.1 Windowed tables

A *windowed table* is a table together with one or more windows. A *window* is a transient data structure associated with a <table expression>. A window is defined explicitly by a <window definition> or implicitly by an <inline window specification>. Implicitly defined windows have an implementation-dependent window name. A window is used to specify window partitions and window frames, which are multisets of rows used in the definition of <window function>s.

Every window defines a *window partitioning* of the rows of the <table expression>. The window partitioning is specified by a list of columns. Window partitioning is similar to forming groups of a grouped table. However, unlike grouped tables, each row is retained in the result of the <table expression>. The *window partition* of a row *R* is the multiset of rows *R2* that are not distinct from *R*, for all columns enumerated in the window partitioning clause. The window partitioning clause is optional; if omitted, there is a single window partition consisting of all the rows in the result.

If a <table expression> is grouped and also has a window, then there is a syntactic transformation that segregates the grouping into a <derived table>, so that the window partitions consist of rows of the <derived table> rather than groups of rows.

A window may define a *window ordering* of rows within each window partition defined by the window. The window ordering of rows within window partitions is specified by a list of <value expression>s, followed by ASC (for ascending order) or DESC (for descending order). In addition, NULLS FIRST or NULLS LAST may be specified, to indicate whether a null value should appear before or after all non-null values in the ordered sequence of each <value expression>.

Optionally, a window may define a *window frame* for each row *R*. A window frame is always defined relative to the current row. A window frame is specified by up to four syntactic elements:

— The choice of RANGE, to indicate a logical definition of the window frame by offsetting forward or backward from the current row by an increment or decrement to the sort key; or ROWS, to indicate a physical definition of the window frame, by counting rows forward or backward from the current row.

— A starting row, which may be the first row of the window partition of *R*, the current row, or some row determined by a logical or physical offset from the current row.

— An ending row, which may be the last row of the window partition of *R*, the current row, or some row determined by a logical or physical offset from the current row.

— A <window frame exclusion>, indicating whether to exclude the current row and/or its peers (if not already excluded by being prior to the starting row or after the ending row).

A window is described by a *window structure descriptor*, including:

— The window name.

— Optionally, the ordering window name—that is, the name of another window, called the *ordering window*, that is used to define the partitioning and ordering of the present window.

— The window partitioning clause—that is, a <window partition clause>, if any is specified in either the present <window specification> or in the window descriptor of the ordering window.

— The window ordering clause—that is, a <window order clause>, if any is specified in either the present <window specification> or in the window descriptor of the ordering window.

 © ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

— The window framing clause—that is, a <window frame clause>, if any.

In general, two <window function>s are computed independently, each one performing its own sort of its data, even if they use the same data and the same <sort specification list>. Since sorts may specify partial orderings, the computation of <window function>s is inevitably non-deterministic to the extent that the ordering is not total. Nevertheless, the user may desire that two <window function>s be computed using the same ordering, so that, for example, two moving aggregates move through the rows of a partition in precisely the same order. Two <window function>s are computed using the same (possibly non-deterministic) window ordering of the rows if any of the following are true:

— The <window function>s identify the same window structure descriptor.

— The <window function>s' window structure descriptors have window partitioning clauses that enumerate the same number of column references, and those column references are pairwise equivalent in their order of occurrence; and their window structure descriptors have window ordering clauses with the same number of <sort key>s, and those <sort key>s are all column references, and those column references are pairwise equivalent in their order of occurrence, and the <sort specification>s pairwise specify or imply <collate clause>s that specify equivalent <collation name>s, the same <ordering specification> (ASC or DESC), and the same <null ordering> (NULLS FIRST or NULLS LAST).

— The window structure descriptor of one <window function> is the ordering window of the other <window function>, or both window structure descriptors identify the same ordering window.

4.3 Data analysis operations (involving tables)

A data analysis function is a function that returns a value derived from a number of rows in the result of a <table expression>. A data analysis function may only be invoked as part of a <query specification> or <select statement: single row>, and then only in certain contexts, identified below. A data analysis function is one of:

— A group function, which is invoked on a grouped table and computes a grouping operation or an aggregate function from a group of the grouped table.

— A window function, which is invoked on a windowed table and computes a rank, row number or window aggregate function.

4.3.1 Group functions

A group function may only appear in the <select list>, <having clause> or <window clause> of a <query specification> or <select statement: single row>, or in the <order by clause> of a cursor that is a simple table query.

A group function is one of:

— The *grouping operation*.

— A *group aggregate function*.

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

4.3 Data analysis operations (involving tables)

The grouping operation is of the form `GROUPING(<column reference>)`. The result of such an invocation is 1 (one) in the case of a row whose values are the results of aggregation over that <column reference> during the execution of a grouped query containing CUBE, ROLLUP, or GROUPING SET, and 0 (zero) otherwise.

4.3.2 Window functions

A window function is a function whose result for a given row is derived from the window frame of that row as defined by a window structure descriptor of a windowed table. Window functions may only appear in the <select list> of a <query specification> or <select statement: single row>, or the <order by clause> of a simply table query.

A window function is one of:

— A rank function.

— A distribution function.

— The row number function.

— A window aggregate function.

The rank functions compute the ordinal rank of a row *R* within the window partition of *R* as defined by a window structure descriptor, according to the window ordering of those rows, also specified by the same window structure descriptor. Rows that are not distinct with respect to the window ordering within their window partition are assigned the same rank. There are two variants, indicated by the keywords RANK and DENSE_RANK.

— If RANK is specified, then the rank of row *R* is defined as 1 (one) plus the number of rows that precede *R* and are not peers of *R*.

  NOTE 3 – This implies that if two or more rows are not distinct with respect to the window ordering, then there will be one or more gaps in the sequential rank numbering.

— If DENSE_RANK is specified, then the rank of row *R* is defined as the number of rows preceding and including *R* that are distinct with respect to the window ordering.

  NOTE 4 – This implies that there are no gaps in the sequential rank numbering of rows in each window partition.

The distribution functions compute a relative rank of a row *R* within the window partition of *R* defined by a window structure descriptor, expressed as an approximate numeric ratio between 0.0 and 1.0. There are two variants, indicated by the keywords PERCENT_RANK and CUME_DIST.

— If PERCENT_RANK is specified, then the relative rank of a row *R* is defined as $(RK-1)/(NR-1)$, where *RK* is defined to be the RANK of *R* and *NR* is defined to be the number of rows in the window partition of *R*.

— If CUME_DIST is specified, then the relative rank of a row *R* is defined as NP/NR, where *NP* is defined to be the number of rows preceding or peer with *R* in the window ordering of the window partition of *R* and *NR* is defined to be the number of rows in the window partition of *R*.

The ROW_NUMBER function computes the sequential row number, starting with 1 (one) for the first row, of the row within its window partition according to the window ordering of the window.

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

The window aggregate functions compute an aggregate value (COUNT, SUM, AVG, *etc.*), the same as a group aggregate function, except that the computation aggregates over the window frame of a row rather than over a group of a grouped table. The hypothetical set functions are not permitted as window aggregate functions.

4.3.3 Aggregate functions

An aggregate function is a function whose result is derived from an aggregation of rows defined by one of:

— The grouping of a grouped table, in which case the aggregate function is a group aggregate function, or set function, and for each group there is one aggregation, which includes every row in the group.

— The window frame of a row *R* of a windowed table relative to a particular window structure descriptor, in which case the aggregate function is a window aggregate function, and the aggregation consists of every row in the window frame of *R*, as defined by the window structure descriptor.

Optionally, the multiset of rows in an aggregation may be filtered, retaining only those rows that satisfy a <search condition> that is specified by a <filter clause>.

The result of the aggregate function COUNT (*) is the number of rows in the aggregation.

Every other aggregate function may be classified as a *unary aggregate function*, a *binary aggregate function*, an *inverse distribution*, or a *hypothetical set function*.

Every unary aggregate function takes an arbitrary <value expression> as the argument; most unary aggregate functions can optionally be qualified with either DISTINCT or ALL. Of the rows in the aggregation, the following do not qualify:

— If DISTINCT is specified, then redundant duplicates.

— Every row in which the <value expression> evaluates to the null value.

If no row qualifies, then the result of COUNT is 0 (zero), and the result of any other aggregate function is the null value.

Otherwise (*i.e.*, at least one row qualifies), the result of the aggregate function is:

— If COUNT <value expression> is specified, then the number of rows that qualify.

— If SUM is specified, then the sum of <value expression> evaluated for each row that qualifies.

— If AVG is specified, then the average of <value expression> evaluated for each row that qualifies.

— If MAX is specified, then the maximum value of <value expression> evaluated for each row that qualifies.

— If MIN is specified, then the minimum value of <value expression> evaluated for each row that qualifies.

— If EVERY is specified, then true if the <value expression> evaluates to *true* for every row that qualifies, otherwise, *false* .

— If ANY or SOME is specified, then *true* if the <value expression> evaluates to *true* for at least one row remaining in the group; otherwise, *false* .

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

4.3 Data analysis operations (involving tables)

— If VAR_POP is specified, then the population variance of <value expression> evaluated for each row remaining in the group, defined as the sum of squares of the difference of <value expression> from the mean of <value expression>, divided by the number of rows remaining.

— If VAR_SAMP is specified, then the sample variance of <value expression> evaluated for each row remaining in the group, defined as the sum of squares of the difference of <value expression> from the mean of <value expression>, divided by the number of rows remaining minus 1 (one).

— If STDDEV_POP is specified, then the population standard deviation of <value expression> evaluated for each row remaining in the group, defined as the square root of the population variance.

— If STDDEV_SAMP is specified, then the sample standard deviation of <value expression> evaluated for each row remaining in the group, defined as the square root of the sample variance.

Neither DISTINCT nor ALL are allowed to be specified for VAR_POP, VAR_SAMP, STDDEV_POP or STDDEV_SAMP; redundant duplicates are not removed when computing these functions.

The binary aggregate functions take a pair of arguments, the <dependent variable expression> and the <independent variable expression>, which are both <numeric value expression>s. Any row in which either argument evaluates to the null value is removed from the group. If there are no rows remaining in the group, then the result of REGR_COUNT is 0 (zero), and the other binary aggregate functions result in the null value. Otherwise, the computation concludes and the result is:

— If REGR_COUNT is specified, then the number of rows remaining in the group.

— If COVAR_POP is specified, then the population covariance, defined as the sum of products of the difference of <independent variable expression> from its mean times the difference of <dependent variable expression> from its mean, divided by the number of rows remaining.

— If COVAR_SAMP is specified, then the sample covariance, defined as the sum of products of the difference of <independent variable expression> from its mean times the difference of <dependent variable expression> from its mean, divided by the number of rows remaining minus 1 (one).

— If CORR is specified, then the correlation coefficient, defined as the ratio of the population covariance divided by the product of the population standard deviation of <independent variable expression> and the population standard deviation of <dependent variable expression>.

— If REGR_R2 is specified, then the square of the correlation coefficient.

— If REGR_SLOPE is specified, then the slope of the least-squares-fit linear equation determined by the (<independent variable expression>, <dependent variable expression>) pairs.

— If REGR_INTERCEPT is specified, then the y-intercept of the least-squares-fit linear equation determined by the (<independent variable expression>, <dependent variable expression>) pairs.

— If REGR_SXX is specified, then the sum of squares of <independent variable expression>.

— If REGR_SYY is specified, then the sum of squares of <dependent variable expression>.

— If REGR_SXY is specified, then the sum of products of <independent variable expression> times <dependent variable expression>.

— If REGR_AVGX is specified, then the average of <independent variable expression>.

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

— If REGR_AVGY is specified, then the average of <dependent variable expression>.

There are two inverse distribution functions, PERCENTILE_CONT and PERCENTILE_DISC. Both inverse distribution functions specify an argument and an ordering of a value expression. The value of the argument should be between 0 (zero) and 1 (one) inclusive. The value expression is evaluated for each row of the group, nulls are discarded, and the remaining rows are ordered. The computation concludes:

— If PERCENTILE_CONT is specified, by considering the pair of consecutive rows that are indicated by the argument, treated as a fraction of the total number of rows in the group, and interpolating the value of the value expression evaluated for these rows.

— If PERCENTILE_DISC is specified, by treating the group as a window partition of the CUME_DIST window function, using the specified ordering of the value expression as the window ordering, and returning the first value expression whose cumulative distribution value is greater than or equal to the argument.

The hypothetical set functions are related to the window functions RANK, DENSE_RANK, PERCENT_RANK and CUME_DIST, and use the same names, though with a different syntax. These functions take an argument *A* and an ordering of a value expression *VE*. *VE* is evaluated for all rows of the group. This multiset of values is augmented with *A*; the resulting collection is treated as a window partition of the corresponding window function whose window ordering is the ordering of the value expression. The result of the hypothetical set function is the value of the eponymous window function for the hypothetical "row" that contributes *A* to the collection.

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

(Blank page)

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

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

       | CEIL | CEILING | CORR | COVAR_POP | COVAR_SAMP | CUME_DIST

       | DENSE_RANK

       | EXCLUDE | EXP

       | FILTER | FLOOR | FOLLOWING

       | LN

       | NULLS

       | OTHERS | OVER

       | PARTITION | PERCENTILE_CONT | PERCENTILE_DISC | PERCENT_RANK | POWER | PRECEDING


       | RANGE | RANK | REGR_AVGX | REGR_AVGY | REGR_COUNT | REGR_INTERCEPT
       | REGR_R2 | REGR_SLOPE | REGR_SXX | REGR_SXY | REGR_SYY | ROW_NUMBER

       | SQRT | STDDEV_POP | STDDEV_SAMP

       | TIES

       | UNBOUNDED

       | VAR_POP | VAR_SAMP

       | WIDTH_BUCKET

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

       | WINDOW
```

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

5.1 <token> and <separator>

Syntax Rules

No additional Syntax Rules.

Access Rules

No additional Access Rules.

General Rules

No additional General Rules.

Conformance Rules

No additional Conformance Rules.

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

5.2 Names and identifiers

Function

Specify names.

Format

```
<window name> ::= <identifier>
```

Syntax Rules

No additional Syntax Rules.

Access Rules

No additional Access Rules.

General Rules

1) Insert this GR A <window name> identifies a window.

Conformance Rules

1) Insert this CR Without Feature T612, “Advanced OLAP operators”, conforming SQL language shall not specify <window name>.

© ISO/IEC 2001 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

(Blank page)

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

6 Scalar expressions

6.1 <set function specification>

Function

Specify a value derived by the application of a function to an argument.

Format

```
<set function specification> ::=
         <aggregate function>
       | <grouping operation>
```

Delete the definition of <general set function>

Delete the definition of <set function type>

Delete the definition of <computational operation>

Delete the definition of <set quantifier>

NOTE 5 – The deleted definitions are now placed in Subclause 8.1, "<aggregate function>".

Syntax Rules

1) Delete SR 1) through 3)

2) Insert this SR If <aggregate function> specifies a <filter clause>, then the <filter condition> shall not contain a <set function specification>.

3) Replace SR 4) If <aggregate function> specifies a <general set function>, then the <value expression> simply contained in the <general set function> shall not contain a <set function specification> or a <subquery>. If the <value expression> contains a column reference that is an outer reference, then that outer reference shall be the only column reference contained in the <value expression>.

4) Insert this SR If <aggregate function> specifies <binary set function>, then neither the <dependent variable expression> nor the <independent variable expression> simply contained in the <binary set function> shall contain a <set function specification> or a <subquery>. If the <dependent variable expression> or <independent variable expression> contains a column reference that is an outer reference, then that outer reference shall be the only column reference contained in the <dependent variable expression> or <independent variable expression>.

5) Insert this SR A <value expression> *VE* simply contained in a <set function specification> *SFE* is an *aggregated argument* of *SFE* if either *SFE* is not an <ordered set function> or *VE* is simply contained in a <within group specification>; otherwise, *VE* is a *non-aggregated argument* of *SFE*.

© ISO/IEC 2001 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: ISO_IEC 9075-5_1999_Amd 1_2001

6.1 <set function specification>

6) Replace SR 5) If an aggregated argument of a <set function specification> contains a column reference that is an outer reference, then the <set function specification> shall be contained in either:

   a) A <select list>.

   b) A <window clause>.

   c) A <subquery> of a <having clause>, in which case the qualifying table of the <column reference> shall be the table referenced by a <table reference> that is directly contained in the <table expression> that directly contains the <having clause>.

7) Insert this SR If <aggregate function> is specified, then the declared type of the result is the declared type of the <aggregate function>. If the declared type is character string, then the collating sequence and coercibility characteristic are the collating sequence and coercibility characteristic of the <aggregate function>.

8) Delete SRs 6) through 14)

9) Delete SR 16)

Access Rules

No additional Access Rules.

General Rules

1) Delete GR 1) and GR 2)

2) Delete GRs 3)a) through 3)g)

3) Insert this GR If <aggregate function> is specified, then the result is the value of the <aggregate function>.

Conformance Rules

1) Delete CRs 1) through 6)

2) Delete CRs 8) and 9)

© ISO/IEC 2001 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 9075-5_1999_Amd 1_2001