



**INTERNATIONAL STANDARD ISO/IEC 9075-1:1999/Amd.1:2001**  
**TECHNICAL CORRIGENDUM 1**

Published 2003-06-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION  
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

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

**Part 1:**  
**Framework (SQL/Framework)**

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

TECHNICAL CORRIGENDUM 1

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

*Partie 1: Charpente (SQL/Charpente)*

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

*RECTIFICATIF TECHNIQUE 1*

Technical Corrigendum 1 to ISO/IEC 9075-1:1999/Amd.1:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

## Contents

|                                                          | Page |
|----------------------------------------------------------|------|
| Introduction .....                                       | 3    |
| 3.3.2.1 Clause, Subclause, and Table relationships ..... | 3    |
| 4.1.1 Operations involving numbers .....                 | 3    |
| 5.1 <token> and <separator> .....                        | 4    |
| 6.1 <set function specification> .....                   | 4    |
| 6.2 <numeric value function>.....                        | 5    |
| 6.4 <value expression> .....                             | 7    |
| 7.0 <table reference> .....                              | 7    |
| 7.5 <window clause> .....                                | 8    |
| 7.6 <query specification> .....                          | 8    |
| 8.1 <aggregate function> .....                           | 8    |
| 13 Information Schema .....                              | 10   |
| Annex B Implementation-defined elements .....            | 10   |
| Annex F Incompatibilities with ISO/IEC 9075:1992 .....   | 10   |

# Information technology — Database languages — SQL —

## Part 1:

## Framework (SQL/Framework)

### AMENDMENT 1

### TECHNICAL CORRIGENDUM 1

## Introduction

1. *Rationale: Add description of new Annex.*

Insert the following point:

- 21) Annex F, “Incompatibilities with ISO/IEC 9075:1992”, is an informative Annex. It lists incompatibilities with the previous version of ISO/IEC 9075.

### 3.3.2.1 Clause, Subclause, and Table relationships

1. *Rationale: Add relationship for new Annex.*

Insert the following row into 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 |
|----------------------------------------------------------|-------------------------------------------------------------|--------------------------------|
| Annex F, “Incompatibilities with ISO/IEC 9075:1992”      | (none)                                                      | (none)                         |

### 4.1.1 Operations involving numbers

1. *Rationale: Editorial.*

Replace the 7<sup>th</sup> bullet of the 1<sup>st</sup> paragraph with:

- <width bucket function> 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 function> are illustrated in Figure 1, “Illustration of WIDTH\_BUCKET Semantics”.

## 5.1 <token> and <separator>

1. *Rationale: Correct the BNF of <non-reserved word> and <reserved word>.*

Replace the productions for <nonreserved word> and <reserved word> with:

```

<non-reserved word> ::=
    !! All alternatives from ISO/IEC 9075-2
  | !! All alternatives from ISO/IEC 9075-5
  | EXCLUDE
  | FOLLOWING
  | NULLS
  | OTHERS
  | PRECEDING
  | TIES
  | UNBOUNDED

<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
  | EXP
  | FILTER | FLOOR
  | LN
  | OVER
  | PARTITION | PERCENTILE_CONT | PERCENTILE_DISC | PERCENT_RANK | POWER
  | 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
  | VAR_POP | VAR_SAMP
  | WIDTH_BUCKET | WINDOW | WITHIN

```

## 6.1 <set function specification>

1. *Rationale: The definition of aggregated argument must overlay the definition in ISO/IEC 9075-2:1999 introduced by a correction elsewhere in this TC.*

Replace Syntax Rule, 5) with:

- 5) Replace SR 3.1) 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*.

## 6.2 <numeric value function>

### 1. Rationale: Clarify the declared type of <numeric value function>

Replace Syntax Rules 1), 2) and 3) with:

- 1) Insert this SR The declared type of the result of <natural logarithm> is an implementation-defined approximate numeric type.
- 2) Insert this SR The declared type of the result of <exponential function> is an implementation-defined approximate numeric type.
- 3) Insert this SR The declared type of the result of <power function> is an implementation-defined approximate numeric type.

### 2. Rationale: Correct the result data type of <floor function> and <ceiling function>.

Replace Syntax Rule 5) with:

- 5) Insert this SR If <floor function> is specified, then  
 Case:
  - a) If the declared type of the simply contained <numeric value expression> *NVE* is exact numeric, then the declared type of the result is exact numeric with implementation-defined precision, with the radix of *NVE*, and with scale 0 (zero).
  - b) Otherwise, the declared type of the result is approximate numeric with implementation-defined precision.

### 3. Rationale: Correct the result data type of <floor function> and <ceiling function>.

Replace Syntax Rule 6) with:

- 6) Insert this SR If <ceiling function> is specified, then  
 Case:
  - a) If the declared type of the simply contained <numeric value expression> *NVE* is exact numeric, then the declared type of the result is exact numeric with implementation-defined precision, with the radix of *NVE*, and with scale 0 (zero).
  - b) Otherwise, the declared type of the result is approximate numeric with implementation-defined precision.

### 4. Rationale: Correct the result data type of <floor function> and <ceiling function>.

Replace General Rule 4) with:

- 4) Insert this GR If <floor function> is specified, then let *V* be the value of the simply contained <numeric value expression> *NVE*.

Case:

- a) If  $V$  is the null value, then the result is the null value.
- b) Otherwise,

Case:

- i) If the most specific type of  $NVE$  is exact numeric, then the result is the greatest exact numeric value with scale 0 (zero) that is less than or equal to  $V$ . If this result is not representable by the result data type, then an exception condition is raised: *data exception — numeric value out of range*.
- ii) Otherwise, the result is the greatest whole number that is less than or equal to  $V$ . If this result is not representable by the result data type, then an exception condition is raised: *data exception — numeric value out of range*.

5. *Rationale: Correct the result data type of <floor function> and <ceiling function>.*

Replace General Rule 5) with:

- 5) Insert this GR If <ceiling function> is specified, then let  $V$  be the value of the simply contained <numeric value expression>  $NVE$ .

Case:

- a) If  $V$  is the null value, then the result is the null value.
- b) Otherwise,

Case:

- i) If the most specific type of  $NVE$  is exact numeric, then the result is the least exact numeric value with scale 0 (zero) that is greater than or equal to  $V$ . If this result is not representable by the result data type, then an exception condition is raised: *data exception — numeric value out of range*.
- ii) Otherwise, the result is the least whole number that is greater than or equal to  $V$ . If this result is not representable by the result data type, then an exception condition is raised: *data exception — numeric value out of range*.

6. *Rationale: Editorial.*

Replace General Rule 6) e) iii) with:

- 6) e) iii) Otherwise, the result is the greatest exact numeric value with scale 0 (zero) that is less than or equal to  $((WBC * (WBB1 - WBO)) / (WBB1 - WBB2) + 1)$ .

## 6.4 <value expression>

1. *Rationale: Define possibly non-deterministic <value expression>.*

Insert the following Syntax Rules:

- 2) Replace SR 9) f) An <aggregate function> that specifies MIN or MAX and that simply contains a <value expression> whose declared type is character string, user-defined type, or datetime with time zone.
- 3) Insert after SR 9) g) A <windowed table function> that specifies ROW\_NUMBER or whose associated <window specification> specifies ROWS.

## 7.0 <table reference>

1. *Rationale: The scope of a <table reference> must include <window clause>*

Insert the following Subclause

### 7.0 <table reference>

Function

Reference a table

Format

No additional format items

Syntax Rules

- 1) Replaces SR 3)a) If a <table reference> *TR* is contained in a <from clause> *FC* with no intervening <query expression>, then the *scope clause SC* of *TR* is the <select statement: single row> or innermost <query specification> that contains *FC*. The scope of the exposed <correlation name> or exposed <table or query name> of *TR* is the <select list>, <where clause>, <group by clause>, <having clause>, and <window clause> of *SC*, together with every <lateral derived table> that is simply contained in *FC* and is preceded by *TR*, and every <collection derived table> that is simply contained in *FC* and is preceded by *TR*, and the <join condition> of all <joined table>s contained in *SC* that contain *TR*. If *SC* is the <query specification> that is the <query expression body> of a simple table query *STQ*, then the scope of the exposed <correlation name> or exposed <table or query name> also includes the <order by clause> of *STQ*.

General Rules

No additional General Rules

Conformance Rules

No additional Conformance Rules

## 7.5 <window clause>

1. *Rationale: Define missing symbol.*

Replace Syntax Rule 4) with:

- 4) Each <column reference> contained in the <window partition clause> or <window order clause> of *WDEF* shall unambiguously reference a column of the derived table *T* that is the result of *TE*. A column referenced in a <window partition clause> is a *partitioning column*.  
NOTE 7 — If *T* is a grouped table, then the <column reference>s contained in <window partition clause> or <window order clause> must reference columns of the grouped table obtained by performing the syntactic transformation in Subclause 7.6, “<query specification>”

## 7.6 <query specification>

1. *Rationale: Correct symbol.*

Replace Syntax Rule 3) m) with:

- 4) m) If  $N_2 = 0$  (zero), then let *CRL* be a zero-length string; otherwise, let *CRL* be:

$$CR_1 \text{ AS } CRI_1, CR_2 \text{ AS } CRI_2, \dots, CR_{N_2} \text{ AS } CRI_{N_2}$$

2. *Rationale: Correct the definition of possibly non-deterministic.*

Replace Syntax Rule 4 with:

- 4) Insert after SR 11) d) The <window clause> contains a reference to a column *C* of *T* that has a data type of character string, user-defined type, TIME WITH TIME ZONE, or TIMESTAMP WITH TIME ZONE, and the functional dependency  $G \rightarrow C$ , where *G* is the set consisting of the grouping columns of *T*, holds in *T*.

## 8.1 <aggregate function>

1. *Rationale: Clarify the declared type of numeric set functions*

Replace Syntax Rule 6) with:

- 6) If COUNT is specified, then the declared type of the result is an implementation-defined exact numeric type with scale of 0 (zero).

Replace Syntax Rule 7) g) with:

- 7) g) If SUM or AVG is specified, then:
- i) *DT* shall be a numeric type or an interval type.
  - ii) If SUM is specified and *DT* is exact numeric with scale *S*, then the declared type of the result is an implementation-defined exact numeric type with scale *S*.

- iii) If AVG is specified and *DT* is exact numeric, then the declared type of the result is an implementation-defined exact numeric type with precision not less than the precision of *DT* and scale not less than the scale of *DT*.
- iv) If *DT* is approximate numeric, then the declared type of the result is an implementation-defined approximate numeric type with precision not less than the precision of *DT*.
- v) If *DT* is interval, then the declared type of the result is interval with the same precision as *DT*.

Replace Syntax Rule 7) h) with:

- 7) h) If VAR\_POP or VAR\_SAMP is specified, then the declared type of the result is an implementation-defined approximate numeric type. If *DT* is an approximate numeric type, then the precision of the result is not less than the precision of *DT*.

Replace Syntax Rule 9) c) with:

- 9) c) Case:
  - i) The declared type of REGR\_COUNT is an implementation-defined exact numeric type with scale of 0 (zero).
  - ii) Otherwise, the declared type of the result is an implementation-defined approximate numeric type. If the declared type *DTDVE* of *DVE* is an approximate numeric type, then the precision of the result is not less than the precision of *DTDVE*. If the declared type *DTIVE* of *I'VE* is an approximate numeric type, then the precision of the result is not less than the precision of *DTIVE*.

- 2. *Rationale: SQL-implementations might need to raise exception conditions during the computation of <aggregate function>s.*

Insert the following General Rule:

- 0) If during the computation of the result of *AF* an intermediate result is not representable in the declared type of the intermediate result, then an exception condition is raised: *data exception — numeric value out of range*.

- 3. *Rationale: Correct set operator.*

Replace General Rule 5) g) with:

- 5) g) The result is the result of the <scalar subquery>
 

```
( SELECT WIFTVAL
  FROM ( SELECT MARKER, WIFT() OVER ( ORDER BY WSP1, ... , WSPK )
        FROM ( SELECT 0, SK1, ... , SKK
              FROM TNAME
              UNION ALL
              VALUES ( 1, VE1, ... , VE_K )
              ) AS TXNAME ( MARKER, CN1, ... , CN_K )
        ) AS TEMPTABLE ( MARKER, WIFTVAL )
  WHERE MARKER = 1 )
```