



**International
Standard**

ISO/IEC 9075-2

**Information technology — Database
languages SQL —**

**Part 2:
Foundation (SQL/Foundation)**

TECHNICAL CORRIGENDUM 1

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

Partie 2: Fondations (SQL/Fondations)

RECTIFICATIF TECHNIQUE 1

**Sixth edition
2023-06**

**TECHNICAL
CORRIGENDUM 1
2026-08**

Sample Document

get full document from standards.iteh.ai



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2026

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Technical Corrigendum 1 to ISO/IEC 9075-2:2023 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

Sample Document

get full document from standards.iteh.ai

Sample Document

get full document from standards.iteh.ai

4 Concepts

4.15 Columns, fields, and attributes

1. *Rationale:* Correctly identify implementation-defined aspects that are dependent on Feature T101, “Enhanced nullability determination”, and remove the duplication of those aspects.

Replace the 5th bullet of the 4th paragraph with:

- The SQL-implementation supports Feature T101, “Enhanced nullability determination”, and is able to deduce that the <boolean value expression> “C IS NULL” can never be *True* when applied to a row in *T* through some implementation-defined (IE011) rule or rules;

NOTE 22 — The notions “known not nullable” and “known-not-null condition” are examples of such additional rules that an SQL-implementation can adopt.

4.17 Tables

4.17.14 Grouped tables

1. *Rationale:* Correct undesired over-restriction.

Replace the 1st paragraph with:

A *grouped table* is a set of groups derived during the evaluation of a <group by clause>. A group *G* is a collection of rows in which, for every grouping column *GC*, if the value of *GC* in some row is *GV*, then the value of *GC* in every row is not distinct from *GV*; moreover, if *R1* is a row in group *G1* of grouped table *GT* and *R2* is a row in *GT* such that for every grouping column *GC* the value of *GC* in *R1* is not distinct from the value of *GC* in *R2*, then *R2* is in *G1*. Every row in *GT* is in exactly one group. A group may be considered as a table. Set functions operate on groups.

4.18 Data analysis operations

4.18.2 Group functions

1. *Rationale:* Reword to avoid possible confusion of interpretation.

Replace the 3rd paragraph with:

The grouping operation is of the form GROUPING(<value expression>). The result of such an invocation is 1 (one) in the case of a row whose values are the results of aggregation that omits that <value expression> from the <group by clause> in a branch of the UNION ALL or UNION DISTINCT that is generated by the syntactic transforms in Subclause 7.13, “<group by clause>” of a grouped query containing CUBE, ROLLUP, or GROUPING SETS, and 0 (zero) otherwise.

5 Lexical elements

5.2 <token> and <separator>

1. *Rationale:* The definition of <whitespace> is any consecutive sequence of whitespace characters. This can be an arbitrary set so <truncating whitespace> needs to use the definition of whitespace, not <whitespace>.

Replace Syntax Rule 4) with:

- 4) <truncating whitespace> is an implementation-defined (IV074) subset of characters each of which satisfies the definition of whitespace found in Clause 3, “Terms and definitions”; the subset shall always include at least <space>.

5.4 Names and identifiers

1. *Rationale:* Correct an incorrect BNF term.

Replace Conformance Rule 25) with:

- 25) Without Feature B209, “PTF extended names”, conforming SQL language shall not contain a <PTF cursor name> or a <PTF descriptor name>.

6 Scalar expressions

6.1 <data type>

1. *Rationale:* Reference the correct type descriptor.

Replace the lead text of General Rule 20) with:

- 20) If <data type> is a <collection type>, then a collection type descriptor is created. Let *KC* be the kind of collection specified by <collection type>. Let *ET* be the element type of <collection type>. Let *ETD* be the type designator of *ET*. The collection type descriptor includes the type designator *ETD KC*, an indication of *KC*, the descriptor of *ET*.

6.10 <window function>

1. *Rationale:* Ensure that symbols are properly defined.

Add the following Syntax Rules:

- 2.1) If <lead or lag function> is specified, then:
 - a) Let *VE1* be <lead or lag extent> and let *DT* be the declared type of *VE1*.
 - b) Case:
 - i) If <offset> is specified, then let *OFF* be <offset>.
 - ii) Otherwise, let *OFF* be 1 (one).

- c) Case:
 - i) If <default expression> is specified, then let $VE2$ be <default expression>. The declared type of $VE2$ shall be compatible with DT .
 - ii) Otherwise, let $VE2$ be $CAST (NULL AS DT)$.
 - d) If <window function null treatment> is specified, then let $NTREAT$ be the <window function null treatment>; otherwise, let $NTREAT$ be $RESPECT NULLS$.
 - e) The declared type of the result is DT .
- 2.2) Let WNS be the <window name or specification>. Let WDX be a window structure descriptor that describes the window defined by WNS .
- 2.3) A <measure name> that is a <window row pattern measure> shall be equivalent to the <measure name> contained in a <row pattern measure definition> $RPMC$ in the <row pattern measures> of WDX . The declared type of the <window function> is the declared type of the <value expression> contained in $RPMC$.

6.12 <case expression>

1. *Rationale: Correct problem with symbol definition.*

Insert the following Syntax Rule:

- 1) ...
- b.1) Let NV be the number of <value expression>s directly contained in the <case abbreviation> and for i , $1 \text{ (one)} \leq i \leq NV$, let V_i be the i -th such <value expression>.

2. *Rationale: Correct problem with symbol definition.*

Replace Syntax Rules 2) e), 2) f), and 2) g) with:

- 2) ...
- e) For i , $1 \text{ (one)} \leq i \leq N$:
- i) Let WOL_i be the <when operand list> of the i -th <simple when clause>.
 - ii) Let R_i be the <result> of the i -th <simple when clause>.
 - iii) Let $M(i)$ be the number of <when operand>s simply contained in WOL_i .
 - iv) For j , $1 \text{ (one)} \leq j \leq M(i)$:
 - 1) Let WO_{ij} be the j -th <when operand> simply contained in WOL_i .
 - 2) Case:
 - A) If WO_{ij} is a <row value predicand>, then let EWO_{ij} be
 $= WO_{i,j}$
 - B) Otherwise, let EWO_{ij} be WO_{ij} .