INTERNATIONAL STANDARD ISO/IEC 9075-2:2016
TECHNICAL CORRIGENDUM 2

Published 2022-06

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 2: Foundation (SQL/Foundation)

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Langages de base de données — SQL — Partie 2: Fondations (SQL/Fondations)

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 2 to ISO/IEC 9075-2:2016 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

ICS 35.060 **Ref. No. ISO/IEC 9075-2:2016/Cor.2:2022(E)**

Published in Switzerland

2 Normative references

2.2 Other international standards

1. *Rationale: Replace invalid URI.*

   Replace the normative reference for RFC 7159 with:

   > [RFC7159] Internet Engineering Task Force, RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014; `https://datatracker.ietf.org/doc/rfc7159/`

3 Definitions, notations, and conventions

3.1 Definitions

3.1.6 Definitions provided in Part 2

1. *Rationale: Correct the definition of whitespace.*

   Replace definition 3.1.6.77 with:

   3.1.6.1 white space
   sequence of one or more characters that have the Unicode property White_Space

   > NOTE 1 — White space is typically used to separate <nondelimiter token>s from one another in SQL text, and is always permitted between two tokens in SQL text.

4 Concepts

4.3 Character strings

4.3.1 Introduction to character strings

1. *Rationale: Correct the definition of truncating whitespace.*

   Replace the 6th paragraph with:

   With two exceptions, a character string expression is assignable only to sites of a character string type whose character set is the same. The exceptions are as specified in Subclause 4.2.8, “Universal character sets”, and such other cases as may be implementation-defined. If a store assignment would result in the loss of non-<truncating whitespace> characters due to truncation, then an exception condition is raised. If a retrieval assignment or evaluation of a <cast specification> would result in the loss of characters due to truncation, then a warning condition is raised.

4.6 Datetimes and intervals

4.6.2 Datetimes

1. *Rationale: Clarify the precision of timestamps.*

   Add the following paragraph after the 13th paragraph:

   When a timestamp value is called for as part of a descriptor or context definition in this Clause (for example, "The creation timestamp" in a descriptor), and no additional information as to the specific type or precision is supplied, an implementation-defined timestamp type is implicit.

4.7 User-defined types

4.7.3 Structured types

4.7.3.3 Constructors

1. *Rationale: Use the correct term.*

   Replace Note 33 with:

   > NOTE 33 — SQL-invoked constructor methods are original methods that cannot be overloaded. An SQL-invoked constructor method and a regular SQL-invoked function may exist such that they have equivalent routine names, the types of the first parameter of the method's augmented SQL parameter declaration list and the function's parameter list are the same, and the types of the corresponding remaining parameters (if any) are identical according to the Syntax Rules of Subclause 9.24, “Data type identity”.

4.23 Integrity constraints

4.23.3 Table constraints

4.23.3.2 Unique constraints

1. *Rationale: Clarify uniqueness in the presence of nulls.*

   Replace the 2nd, 3rd and 4th paragraphs with:

   Let *T* be a table and let *R1* and *R2* be two rows of *T*. If there is some column for which the corresponding values from *R1* and *R2* are not equal, *R1* and *R2* are *unique with nulls distinct*. If there is some column for which the corresponding values from *R1* and *R2* are distinct, *R1* and *R2* are *unique with nulls not distinct*. If all column values of both *R1* and *R2* are the null value, it is implementation-defined whether *R1* and *R2* are unique with nulls distinct, or unique with nulls not distinct. It is implementation-defined whether the *implementation uniqueness rule* is unique with nulls distinct or unique with nulls not distinct.

   If the table descriptor for base table *T* includes a unique constraint descriptor indicating that the unique constraint was defined with PRIMARY KEY, then the columns of that unique constraint constitute the *primary key* of *T*. A table that has a primary key cannot have a proper supertable.