INTERNATIONAL STANDARD ISO/IEC 9075:1992
Technical Corrigendum 3
ISO/IEC 9075-3:1995
Technical Corrigendum 1
ISO/IEC 9075-4:1996
Technical Corrigendum 1

TECHNICAL CORRIGENDUM

Published 1999-11-15

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

TECHNICAL CORRIGENDUM

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

RECTIFICATIF TECHNIQUE

Technical Corrigendum 3 to International Standard ISO/IEC 9075:1992, Technical Corrigendum 1 to International Standard ISO/IEC 9075-3:1995 and Technical Corrigendum 1 to International Standard ISO/IEC 9075-4:1996 were prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management services*.

Relation to previous technical corrigenda:

This Corrigendum contains the cumulative set of corrections to ISO/IEC 9075:1992, ISO/IEC 9075-3:1995 and ISO/IEC 9075-4:1996.
It completely subsumes and replaces all previous corrigenda for ISO/IEC 9075.

Statement of purpose for rationale:

A statement indicating the rationale for each change to ISO/IEC 9075 is included. This is to inform the users of that standard as to the reason why it was judged necessary to change the original wording. In many cases the reason is editorial or to clarify the wording; in some cases it is to correct an error or an omission in the original wording.

Notes on rule numbering:

This Corrigendum introduces some new Syntax, Access, General and Leveling Rules. The new Rules in this Corrigendum have been numbered as follows:
Rules inserted between, for example, Rules 7) and 8) (in ISO/IEC 9075:1992) are numbered 7.1), 7.2), etc. [or 7) a.1), 7) a.2), etc.]. Those inserted before Rule 1) are numbered 0.1), 0.2), etc.

ICS 35.060 **Ref. No. ISO/IEC 9075/Cor:1999(E)**

Contents

Page

ISO/IEC 9075:1992
Database Languages - SQL

Introduction

1. *Rationale: In the list of significant new features, the wording incorrectly implies that all the examples listed in item 10) are referential integrity facilities.*

On page xiv, in Significant new feature 10), replace "referential integrity" with "integrity".

2 Normative references

1. *Rationale: Editorial. The (non-extended) Pascal standard should be identified as ISO/IEC 7185 rather than ISO 7185. The designation was changed in 1990 when the standard was revised.*

Change "ISO 7185:1990" to "ISO/IEC 7185:1990"

2. *Rationale: Editorial.*

Add the following reference after the reference to "ISO 8601:1988":

— ISO 8649:1988, *Information Processing Systems — Open Systems Interconnection — Service Definition for the Association Control Service Element.*

3. *Rationale: The newly revised Ada language standard (ISO/IEC-8652:1995, Information technology — Programming languages — Ada) contains support for decimal-encoded numeric data and variable length character strings. The revised interface allows newly written applications in the revised Ada language access to these features of SQL; previously written Ada applications, conformant with the earlier Ada interface, are conformant with the revised interface.*

Replace the reference to ISO/IEC 8652:1987) with:

— ISO/IEC 8652:1995, *Information technology — Programming languages — Ada.*

4. *Rationale: Editorial.*

Add the following reference after the reference to ISO/IEC 8824:1990:

— ISO/IEC 9579-1:1993, *Information technology — Open Systems Interconnection — Remote Database Access, Part 1: Generic Model, Service, and Protocol.*

Add the following reference after the reference to ISO/IEC 9899:

— ISO/IEC 10026-21, *Information technology — Open Systems Interconnection — Distributed Transaction Processing — Part 2: Service Definition.*

3.3.4.3 Terms denoting rule requirements

1. *Rationale: The following unifies the SQLSTATE returned for the different ways of invoking an SQL statement.*

In the first and second paragraphs, replace "*syntax error or access rule violation* (if this situation occurs during dynamic execution of an SQL-statement, then the exception that is raised is *syntax error or access rule violation in dynamic SQL statement*; if the situation occurs during direct invocation of an SQL-statement, then the exception that is raised is *syntax or access rule violation in direct SQL statement*)" with "*syntax error or access rule violation*".

4.2 Character strings

1. *Rationale: Editorial.*

In the second paragraph, replace the last sentence with:

> Character sets defined by standards or by implementations reside in the Information Schema (named INFORMATION_SCHEMA) in each catalog, as do collations and translations defined by standards and collations, translations, and form-of-use conversions defined by implementations.

4.2.1 Character strings and collating sequences

1. *Rationale: The following changes make the definitions of character set and collation descriptors more precise.*

Replace the text on page 17 that occurs after the first paragraph with:

> A character set is described by a character set descriptor. A character set descriptor includes:
>
>
>
> — the name of the character set or character repertoire,
>
> — if the character set is a character repertoire, then the name of the form-of-use,
>
> — an indication of what characters are in the character set, and
>
> — whether or not the character set uses the DEFAULT collation for its character repertoire, and,
>
> — if the character set does not utilize the DEFAULT collation for its character repertoire, then the <translation name> contained in the character set's <translation collation>, if any, the <collation name> contained in the character set's <collate clause> or <limited collation definition>, if any, and, whether or not DESC was specified in the reference to the collation
>
> For every character set, there is at least one collation. A collation is described by a collation descriptor. A collation descriptor includes:
>
> — the name of the collation,
>
> — the name of the character repertoire on which the collation operates,
>
> — whether the collation has the NO PAD or the PAD SPACE attribute, and
>
> — whether or not this collation utilizes the DEFAULT collation for its character repertoire,

— if the collation does not utilize the DEFAULT collation for its character repertoire, then the <translation name> contained in the collation's <translation collation>, if any, the <collation name> contained in the collation's <collation source>, if any, and whether or not DESC was specified in the definition of the collation.

4.2.3 Rules determining collating sequence usage

1. *Rationale: Editorial. The second bullet of the first paragraph (When columns are involved (e.g.., comparing ...)) predicates the possibility of just one column being involved, but the sentence ends with words that imply at least two columns. In addition, it is not clear whether the intent is that default collating sequence of a column overrides the collating sequence (necessarily the default for the relevant repertoire) for a literal.*

Replace the second bullet with:

— When one or more columns are involved (e.g., comparing two columns, or comparing a column to a literal), then provided that all columns involved have the same default collating sequence and there is no explicit specification of a collating sequence, that default collating sequence is used.

4.3.1 Bit string comparison and assignment

1. *Rationale: The second paragraph can be read to mean that all source bits are assigned successively to the most significant bit position of the receiving (target) string. Note that there does not appear to be a corresponding rule for character string assignment in the relevant subclause of "Concepts". Relevant rules do exist in clause 9, but do not address the order in which bits (or characters) are assigned.*

Delete the second paragraph of the subclause.

4.4 Numbers

1. *Rationale: Clarification.*

Add the following sentence immediately before the heading of Subclause 4.4.1 Characteristics of Numbers:

A value described by a numeric data type descriptor is always signed.

4.4.1 Characteristics of numbers

1. *Rationale: In several paragraphs in this subclause, phrases similar to "values representable in the data type" are used when the meaning is "members of the data type". A data type is defined as a set of values, so the term is unnecessarily complicated. Since there are rules defining mappings from all numbers within the range of a data type onto that data type, the meaning of "representable" is anyway somewhat ambiguous. Simpify the convoluted wording used to specify that rounding is always towards zero. Use numeric instead of numerical for consistency.*

Replace the fourth, fifth, sixth, seventh and eighth paragraphs with:

An approximation obtained by truncation of a numeric value *N* for an <exact numeric type> *T* is a value *V* in *T* such that *N* is not closer to zero than is *V* and there is no value in *T* between *V* and *N*.

An approximation obtained by rounding of a numeric value N for an <exact numeric type> T is a value V in T such that the absolute value of the difference between N and the numeric value of V is not greater than half

the absolute value of the difference between two successive numeric values in T. If there are more than one such values V, then it is implementation-defined which one is taken.

All numeric values between the smallest and the largest value, inclusive, in a given exact numeric type have an approximation obtained by rounding or truncation for that type; it is implementation-defined which other numeric values have such approximations.

An approximation obtained by truncation or rounding of a numeric value N for an <approximate numeric type> T is a value V in T such that there is no value in T that lies between the V and N.

If there are more than one such values V then it is implementation-defined which one is taken. It is implementation-defined which numeric values have approximations obtained by rounding or truncation for a given approximate numeric type.

4.5 Datetimes and intervals

1. Rationale: Clarification.

Add the following sentence before the paragraph starting "Every datetime ...":

A value described by an interval data type descriptor is always signed.

4.5.1 Datetimes

1. Rationale: Editorial.

Replace the first paragraph with:

Table 4, "Fields in datetime items", specifies the fields that can make up a datetime value; a datetime value is made up of a subset of those fields. Not all of the fields shown are required to be in the subset, but every field that appears in the table between the first included primary field and the last included primary field shall also be included. If either timezone field is in the subset, then both of them shall be included.

2. Rationale: Clarify the treatment of time zones.

Replace the sixth, seventh, eighth and ninth paragraphs with:

A datetime data type that specifies WITH TIME ZONE is a data type that is *datetime with time zone*.

The surface of the earth is divided into zones, called time zones, in which every correct clock tells the same time, known as *local time*. Local time is equal to UTC (Coordinated Universal Time) plus the *time zone displacement*, which is a value of INTERVAL HOUR TO MINUTE, between -'12:59' and +'13:00'. The time zone displacement is constant throughout a time zone, changing at the beginning and end of Daylight Saving Time, where applicable.

A datetime value, of data type TIME or TIMESTAMP, may represent a local time or UTC. A data item may be defined to contain either a datetime value only, or a datetime value together with a time zone displacement.

For the convenience of users, whenever a datetime value with time zone is to be implicitly derived from one without (for example, in a simple assignment operation), SQL assumes the value without time zone to be local, subtracts the default session time zone displacement from it to give UTC, and associates that time zone displacement with the result.

Conversely, whenever a datetime value without time zone is to be implicitly derived from one with, SQL assumes the value with time zone to be UTC, adds the time zone displacement to it to give local, and the result, without any time zone displacement, is local.

4.5.2 Intervals

1. Rationale: Editorial

Replace the fifth paragraph with:

Within an item of type interval, the first field is constrained only by the <interval leading field precision> of the associated <interval qualifier>. Table 7, "Valid values for fields in INTERVAL items", specifies the constraints on subsequent field values.

2. Rationale: Editorial.

In Table 7, replace "<interval leading field precision" with "<interval leading field precision>" (two occurrences).

3. Rationale: Clarify the precision of interval fields.

Replace the sixth paragraph (that following Table 7) with:

Values in interval fields other than SECOND are integers, and have precision 2 when not the first field. SECOND, however, can be defined to have an <interval fractional seconds precision> that indicates the number of decimal digits maintained following the decimal point in the seconds value. When not the first field, SECOND has a precision of 2 places before the decimal point.

4. Rationale: The wording seemed to imply that extra fields effectively added to a day-time interval for the purposes of operations between two fields are all added at the same end. However, comparison of an HOUR interval with a DAY-MINUTE interval would require extension of the HOUR interval at both ends.

In the ninth paragraph, change "either the most significant end or the least significant end of one or both day-time intervals" to "either the most significant end of one interval, or the least significant end of one interval, or both".

4.5.3 Operations involving datetimes and intervals

1. Rationale: Editorial. Table 8, Valid operators involving datetimes and intervals, specifies not the results of arithmetic operations involving datetime and interval operands, but the result types of operations between operands of those types.

Replace the first paragraph with:

Table 8, "Valid operators involving datetimes and intervals", specifies the data types of the results of arithmetic operations involving datetime and interval operands.

4.6 Type conversion and mixing of data types

1. *Rationale: Correct an inconsistency in style between the third sentence of the third paragraph and similar statements elsewhere in the standard.Also correct the grammar of the last sentence.*

Replace the third paragraph with:

> Values corresponding to the data types BIT and BIT VARYING are always mutually comparable and are mutually assignable. If a store assignment would result in the loss of bits due to truncation, then an exception condition is raised. When values of unequal length are compared, if the shorter is a prefix of the longer, then the shorter is less than the longer; otherwise, the longer is effectively truncated to the length of the shorter for the purposes of comparison. When values of equal length are to be compared, then a bit-by-bit comparison is made. A 0-bit is less than a 1-bit.

4.8 Columns

1. *Rationale: Editorial.*

Replace the third paragraph and the lead-in to the bullet list of the fourth paragraph with:

> Every column has a nullability characteristic that indicates whether any attempt to store a null value into that column will inevitably cause an exception condition to be raised, and whether any attempt to retrieve a value from that column can ever result in a null value. The possible values of the nullability characteristic are known not nullable and possibly nullable.
>
> A column *C* with <column name> *CN* of a base table *T* has a nullability characteristic that is known not nullable if any only if either:

2. *Rationale: Editorial.*

In the penultimate paragraph on page 28, replace "<row value constructor expression>" with "<row value constructor element>".

4.9 Tables

1. *Rationale: Editorial.*

Replace the first sentence of the third paragraph with:

> A table is either a base table or a derived table.

2. *Rationale: There is no named derived table other than a viewed table.*

After the paragraph that begins with "A derived table descriptor describes a derived table.", delete the first item ("— if the table is named, then the name of the table;").

3. *Rationale: There is no named derived table other than a viewed table.*

After the paragraph that begins with "A view descriptor describes a view.", insert "— the name of the view, and" before the existing item.

4.10.2 Table constraints

1. *Rationale: Editorial.*

In the **Note**, replace "<match option>" with "<match type>".

2. *Rationale: Editorial.*

In the paragraph that begins with "A referential constraint is satisfied", replace "<match option>" with "<match type>".

4.18.1 Status parameters

1. *Rationale: To insure that the value returned to the user in SQLSTATE is representative of the actual state of the transaction or SQL-statement.*

Add the following as the last paragraph:

> For the purpose of choosing status parameter values to be returned, *exceptions* for transaction rollback have precedence over *exceptions* for statement failure. Similarly, completion condition *no data* has precedence over *warning*, which has precedence over *successful completion*. All *exceptions* have precedence over all completion conditions. The values assigned to SQLSTATE shall obey these precedence rules.

4.21 Cursors

1. *Rationale: Define "dynamic cursor" and "extended dynamic cursor".*

Add the following after the first sentence of the first paragraph:

> 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.

2. *Rationale: Correct concepts section regarding when cursors are destroyed.*

Replace the second paragraph with:

> For every <declare cursor> or <dynamic declare cursor> in a <module>, a cursor is effectively created when an SQL-transaction (see Subclause 4.28, "SQL-transactions") referencing the <module> is initiated, and destroyed when that SQL-transaction is terminated. An extended dynamic cursor is effectively created when an <allocate cursor statement> is executed within an SQL-transaction and destroyed when that SQL-transaction is terminated. In addition, a dynamic cursor is destroyed when a <deallocate prepared statement> is executed that deallocates the prepared statement on which the cursor is based.

4.22.6 SQL-statements and transaction states

1. *Rationale: No statement can be both transaction-initiating and not transaction-initiating.*

In the first dashed list (of transaction-initiating SQL-statements), in the bulleted sublist of SQL-data statements, delete the entry for <dynamic select statement>.

4.24 SQL dynamic statements

1. *Rationale: Editorial.*

In the fourth paragraph, replace "<target specification>s" with "<simple value specification>s".

2. *Rationale: Editorial.*

In the eighth paragraph, replace the first occurrence of "<SQL statement>s" with "<SQL procedure statement>s", and replace the second occurrence of "<SQL statement>s" with "<embedded SQL statement>s".

4.26 Privileges

1. *Rationale: Editorial.*

In the fourth paragraph on page 52, replace "<module authorization identifier> is" with "<schema authorization identifier> is".

2. *Rationale: Provide missing rules that cover the acquisition of the necessary privileges to acquire the WITH GRANT OPTION on views through a grant to PUBLIC.*

Add the following before the antepenultimate paragraph of this Subclause:

> The phrase *user privileges* refers to the set of privileges defined by the privilege descriptors whose grantee is either the identified <authorization identifier> or PUBLIC.

4.28 SQL-transactions

1. *Rationale: Clarification.*

In the paragraph that begins "In some environments (e.g., remote database access)", replace all occurrences of "SQL-environment" with "SQL-implementation".

4.29 SQL-connections

1. *Rationale: Editorial.*

Replace the second paragraph with :

> An SQL-connection is an *active SQL-connection* if any SQL-statement that initiates or requires an SQL-transaction has been executed at its SQL-server via that SQL-connection during the current SQL-transaction.

2. Rationale: Clarification.

In the last sentence of the penultimate paragraph, replace "SQL-environment" with "SQL-implementation".

4.31 Client-server operation

1. Rationale: Clarification.

Replace the first sentence with :

> As perceived by an SQL-agent, an SQL-implementation consists of one or more SQL-servers and one SQL-client through which SQL-connections can be made to them.

5.2 <token> and <separator>

1. Rationale: The maximum length of an <identifier> is intended to be 128 characters.

Replace Syntax Rule 8) with :

8) In a <regular identifier>, the number of <underscore>s plus the number of <identifier part>s shall be less than 128.

2. Rationale: A <regular identifier> shall not contain any <quote> or <double quote>. Thus, a <delimited identifier> with a <delimited identifier body> containing a <quote> or <double quote> is not equivalent to any <regular identifier>.

In Syntax Rule 13), delete the expression "(with all occurrences of <quote> replaced by <quote symbol> and all occurrences of <doublequote symbol> replaced by <double quote>)".

3. Rationale: Correct the incorrect references to "<quote>" and "<quote symbol>" and delete the redundant references to "<double quote>"s and "<double quote symbol>"s in Syntax Rule 14.

In Syntax Rule 14), delete "(with all occurrences of <quote> replaced by <quote symbol> and all occurrences of <doublequote symbol> replaced by <doublequote>)".

4. Rationale: A <character representation> does not appear in a <regular identifier> or in a <delimited identifier body>.

Replace Leveling Rule 2) a) with :

a) The number of <underscore>s plus the number of <identifier part>s contained in a <regular identifier> shall be less than 18.

Insert the following Leveling Rule 2) a.1):

a.1) The <delimited identifier body> of a <delimited identifier> shall not comprise more than 18 <delimited identifier part>s.