

Information technology — Database languages — SQL

TECHNICAL CORRIGENDUM 1

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

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to International Standard ISO/IEC 9075:1992 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open systems interconnection, data management and open distributed processing*.

Relation to previous technical corrigenda:

This Corrigendum contains the cumulative set of corrections to ISO/IEC 9075:1992. It cancels and replaces Technical Corrigendum 1:1994.

Statement of purpose for rationale:

A statement indicating the rationale for each change to ISO/IEC 9075:1992 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.

Contents

	Page
Introduction	5
2 Normative references	5
3.3.4.3 Terms denoting rule requirements	5
4.2 Character strings	6
4.2.1 Character strings and collating sequences	6
4.4 Numbers	7
4.5 Datetimes and intervals	7
4.5.1 Datetimes	7
4.5.2 Intervals	7
4.8 Columns	7
4.9 Tables	7
4.10.2 Table constraints	8
4.18.1 Status parameters	8
4.22.6 SQL-statements and transaction states	8
4.24 SQL dynamic statements	8
4.26 Privileges	8
4.28 SQL-transactions	9
4.29 SQL-connections	9
4.31 Client-server operation	9
5.2 <token> and <separator>	9
5.3 <literal>	10
5.4 Names and identifiers	11
6.1 <data type>	11
6.2 <value specification> and <target specification>	11
6.3 <table reference>	12
6.6 <numeric value function>	12
6.8 <datetime value function>	13
6.10 <cast specification>	13
6.12 <numeric value expression>	15
6.14 <datetime value expression>	15
6.15 <interval value expression>	16
7.1 <row value constructor>	16
7.4 <from clause>	16
7.5 <joined table>	17
7.6 <where clause>	18
7.9 <query specification>	18
7.10 <query expression>	19
7.11 <scalar subquery>, <row subquery>, and <table subquery>	19
8.3 <between predicate>	20
8.7 <quantified comparison predicate>	20
9.1 Retrieval assignment	20
9.2 Store assignment	21
10.1 <interval qualifier>	21
10.4 <character set specification>	22
10.5 <collate clause>	22
10.6 <constraint name definition> and <constraint attributes>	22
11.1 <schema definition>	22
11.2 <drop schema statement>	22
11.4 <column definition>	23
11.5 <default clause>	23
11.6 <table constraint definition>	25
11.8 <referential constraint definition>	25
11.9 <check constraint definition>	25
11.11 <add column definition>	25

11.16 <add table constraint definition>	26
11.17 <drop table constraint definition>	26
11.18 <drop table statement>	26
11.19 <view definition>	26
11.21 <domain definition>	27
11.25 <add domain constraint definition>	28
11.26 <drop domain constraint definition>	28
11.27 <drop domain statement>	28
11.28 <character set definition>	28
11.29 <drop character set statement>	28
11.30 <collation definition>	29
11.31 <drop collation statement>	29
11.32 <translation definition>	30
11.34 <assertion definition>	30
11.36 <grant statement>	30
11.37 <revoke statement>	32
12.3 <procedure>	37
12.4 Calls to a <procedure>	38
13.1 <declare cursor>	47
13.2 <open statement>	48
13.3 <fetch statement>	48
13.4 <close statement>	48
13.5 <select statement: single row>	49
13.6 <delete statement: positioned>	49
13.7 <delete statement: searched>	49
13.8 <insert statement>	49
13.9 <update statement: positioned>	50
13.10 <update statement: searched>	50
14.1 <set transaction statement>	50
14.2 <set constraints mode statement>	50
14.3 <commit statement>	51
15.1 <connect statement>	51
15.2 <set connection statement>	51
15.3 <disconnect statement>	51
16.5 <set local time zone statement>	51
17.1 Description of SQL item descriptor areas	51
17.2 <allocate descriptor statement>	52
17.3 <deallocate descriptor statement>	52
17.4 <get descriptor statement>	52
17.5 <set descriptor statement>	52
17.6 <prepare statement>	53
17.9 <using clause>	56
17.10 <execute statement>	56
17.11 <execute immediate statement>	56
17.15 <dynamic fetch statement>	56
17.18 <dynamic update statement: positioned>	56
18.1 <get diagnostics statement>	56
19.1 <embedded SQL host program>	58
19.5 <embedded SQL COBOL program>	58
20.1 <direct SQL statement>	59
21.1 Introduction	59
21.2.2 INFORMATION_SCHEMA_CATALOG_NAME base table	59
21.2.3 INFORMATION_SCHEMA_CATALOG_NAME_CARDINALITY assertion	60
21.2.5 DOMAINS view	60
21.2.6 DOMAIN_CONSTRAINTS view	60

21.2.9 COLUMNS view	60
21.2.17 ASSERTIONS view	61
21.2.27 SQL_IDENTIFIER domain	61
21.3.5 DATA_TYPE_DESCRIPTOR base table	61
21.3.6 DOMAINS base table	64
21.3.8 TABLES base table	64
21.3.10 COLUMNS base table	64
21.3.11 VIEW_TABLE_USAGE base table	65
21.3.12 VIEW_COLUMN_USAGE base table	65
21.3.13 TABLE_CONSTRAINTS base table	65
21.3.15 REFERENTIAL_CONSTRAINTS base table	66
21.3.17 CHECK_TABLE_USAGE base table	66
21.3.18 CHECK_COLUMN_USAGE base table	66
21.3.21 COLUMN_PRIVILEGES base table	66
21.3.22 USAGE_PRIVILEGES base table	66
21.3.23 CHARACTER_SETS base table	67
21.3.24 COLLATIONS base table	67
21.3.26 SQL_LANGUAGES base table	67
22.1 SQLSTATE	67
22.2 SQLCODE	69
22.3 Remote Database Access SQLSTATE Subclasses	69
23.2 Claims of Conformance	72
23.3 Extensions and options	72
Annex A.1 Intermediate SQL Specifications	72
A.2 Entry SQL Specifications	74
Annex B: Implementation-defined elements	77
Annex C Implementation-dependent elements	78
Annex E Incompatibilities with ISO/IEC 9075:1989	79
Annex F Maintenance and interpretation of SQL	80

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.*

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.*
2. *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 the following:

- ISO/IEC 8652:1995, *Information technology—Programming languages—Ada.*
3. *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 10026-2:1996, *Information technology—Open Systems Interconnection—Distributed Transaction Processing—Part 2: OSI TP Service.*

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 the following:

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 the following:

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.

4.4 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.

4.5 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.*

In the penultimate paragraph on page 24, replace "Universal Coordinated Time" with "Coordinated Universal Time".

4.5.2 Intervals

1. *Rationale: Editorial.*

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

4.8 Columns

1. *Rationale: Editorial.*

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

4.9 Tables

1. *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;").

2. *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.

4.22.6 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 the following:

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 the following:

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 the following:

- 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 the following:

- 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.

5.3 <literal>

1. *Rationale: Support changes to Subclause 6.10.*

In Format, replace the BNF for <date string>, <time string>, <timestamp string>, <interval string> with the following:

<date string> ::= <quote> <unquoted date string> <quote>

<time string> ::= <quote> <unquoted time string> <quote>

<timestamp string> ::= <quote> <unquoted timestamp string> <quote>

<interval string> ::= <quote> <unquoted interval string> <quote>

Add the following to Format:

<unquoted date string> ::= <date value>

<unquoted time string> ::= <time value> [<time zone interval>]

<unquoted timestamp string> ::= <unquoted date string> <space> <unquoted time string>

<unquoted interval string> ::= [<sign>] { <year-month literal> | <day-time literal> }

Add the following sentence at the end of General Rule 5):

If a <sign> is specified in both possible locations in an <interval literal>, the sign of the literal is determined by the normal mathematical interpretation of multiple sign operators, i.e., double negation results in a positive literal.

2. *Rationale: Ellipses were placed both in the definition of <separator> and in the usage of <separator> in several types of string literals. This leads to a small ambiguity when two separator characters follow one another. The following changes remove the ellipses in the latter of these two places:*

In Format, in each of the productions for <character string literal>, <national character string literal>, <bit string literal>, <hex string literal>, replace "<separator> ..." with "<separator>".

In Syntax Rules 1) - 3) replace "<separator> ..." with "<separator>".

5.4 Names and identifiers

1. *Rationale: Editorial.*

In Syntax Rule 1), replace "identified" with "indicated".

2. *Rationale: Add the rule for the comparison of <character set name>s missing since <character set name> was decoupled from <qualified name>.*

Add the following Syntax Rule:

- 10.5) Two <character set names> are equal if they have the same <SQL language identifier> and the same <schema name>, regardless of whether the <schema name>s are implicit or explicit.

6.1 <data type>

1. *Rationale: Clarify the error condition that should be returned if a datetime overflow occurs.*

Delete General Rule 6).

2. *Rationale: Clarification.*

Add the following Note after Syntax Rule 28):

Note: The length of interval data types is specified in the General Rules of 10.1, "<interval qualifier>".

6.2 <value specification> and <target specification>

1. *Rationale: Identify possible upward incompatibility.*

Add the following at the end of Syntax Rule 3):

Note: In an environment where the SQL-implementation conforms to Entry SQL, conforming SQL language that contains either:

- a) a specified or implied <comparison predicate> that compares the <value specification> USER with a <value specification> other than USER, or
- b) a specified or implied assignment in which the "value" (as defined in Subclause 9.2 Store assignment) contains the <value specification> USER

will become non-conforming in an environment where the SQL-implementation conforms to Intermediate SQL or Full SQL, unless the character repertoire of the implementation-defined character set in that environment is identical to the character repertoire of SQL_TEXT.

6.3 <table reference>

1. *Rationale: Editorial.*

In Syntax Rule 2) a), replace "with no intervening" with "without an intervening".

2. *Rationale: Editorial.*

In Syntax Rule 2) a), replace "The scope clause of the exposed" with "The scope of the exposed".

3. *Rationale: Editorial.*

In Syntax Rule 2) b), replace "with no intervening" with "without an intervening".

4. *Rationale: Clarify that references to non-existing objects is only allowed within the same <schema definition>.*

Add the following Syntax Rule:

- 8.1) Let *T* be the table identified by the <table name> immediately contained in <table reference>. If the <table reference> is not contained in a <schema definition>, then the schema identified by the explicit or implicit qualifier of the <table name> shall include the descriptor of *T*. If the <table reference> is contained in a <schema definition> *S*, then the schema identified by the explicit or implicit qualifier of the <table name> shall include the descriptor of *T*, or *S* shall include a <schema element> that creates the descriptor of *T*.

5. *Rationale: Editorial.*

Delete the first sentence of Access Rule 1).

6.6 <numeric value function>

1. *Rationale: Correct Format for <position expression> to support <bit value expression>s in addition to <character value expression>s.*

In the Format replace the production for <position expression> with the following:

```
<position expression> ::=
    POSITION <left paren> <string value expression>
    IN <string value expression> <right paren>
```

Replace Syntax Rule 1) with the following:

- 1) If <position expression> is specified, then both <string value expression>s shall be <bit value expression>s or both shall be <character value expression>s having the same character repertoire.

Replace General Rules 1) and 2) with the following:

- 1) If <position expression> is specified and neither <string value expression> is the null value, then
Case:
 - a) If the first <string value expression> has a length 0, then the result is 1.
 - b) If the value of the first <string value expression> is equal to an identical-length substring of contiguous characters or bits from the value of the second <string value expression>, then the result is 1 greater than the number of characters or bits within the value of the second <string value expression> preceding the start of the first such substring.
 - c) Otherwise, the result is 0.
- 2) If <position expression> is specified and either <string value expression> is the null value, then the result is the null value.

6.8 <datetime value function>

1. *Rationale: The following change clarifies that <datetime value function> is effectively evaluated only once per SQL statement.*

Replace General Rule 3) with the following:

- 3) If an SQL-statement causes the evaluation of one or more <datetime value function>s, then all such evaluations are effectively performed simultaneously. The time of evaluation of the <datetime value function> during the execution of the SQL-statement is implementation-dependent.

6.10 <cast specification>

1. *Rationale: Clarify that references to non-existing objects is only allowed within the same <schema definition>.*

Add the following Syntax Rule:

- 8.1) If <domain name> is specified, then let *D* be the domain identified by the <domain name>. If the <cast specification> is not contained in a <schema definition>, then the schema identified by the explicit or implicit qualifier of the <domain name> shall include the descriptor of *D*. If the <cast specification> is contained in a <schema definition> *S*, then the schema identified by the explicit or implicit qualifier of the <domain name> shall include the descriptor of *D*, or *S* shall include a <schema element> that creates the descriptor of *D*.

2. *Rationale: Legalize a simpler format for character strings being cast to temporal data types.*

In General Rule 9) a) i), replace "rules for <literal>" with "rules for <literal> or for <unquoted date string>".

3. *Rationale: Clarify the error condition that should be returned if a datetime overflow occurs.*

Replace General Rule 9) a) ii) with the following:

- ii) Otherwise,
 - 1) If a <datetime value> does not conform to the natural rules for dates or times according to the Gregorian calendar, an exception condition is raised: *data exception—invalid datetime format*.
 - 2) Otherwise, an exception condition is raised: *data exception—invalid character value for cast*.

4. *Rationale: Legalize a simpler format for character strings being cast to temporal data types.*

In General Rule 10) a) i), replace "rules for <literal>" with "rules for <literal> or for <unquoted time string>".

5. *Rationale: Clarify the error condition that should be returned if a datetime overflow occurs.*

Replace General Rule 10) a) ii) with the following:

- ii) Otherwise,
 - 1) If a <datetime value> does not conform to the natural rules for dates or times according to the Gregorian calendar, an exception condition is raised: *data exception—invalid datetime format*.
 - 2) Otherwise, an exception condition is raised: *data exception—invalid character value for cast*.

6. *Rationale: Legalize a simpler format for character strings being cast to temporal data types.*

In General Rule 11) a) i), replace "rules for <literal>" with "rules for <literal> or for <unquoted timestamp string>".

7. *Rationale: Clarify the error condition that should be returned if a datetime overflow occurs.*

Replace General Rule 11) a) ii) with the following:

- ii) Otherwise,
 - 1) If a <datetime value> does not conform to the natural rules for dates or times according to the Gregorian calendar, an exception condition is raised: *data exception—invalid datetime format*.
 - 2) Otherwise, an exception condition is raised: *data exception—invalid character value for cast*.

8. *Rationale: Legalize a simpler format for character strings being cast to temporal data types.*

In General Rule 12) b) i), replace "rules for <literal>" with "rules for <literal> or for <unquoted interval string>".

9. *Rationale: Clarify the error condition that should be returned if a datetime overflow occurs.*

Replace General Rule 12) b) ii) with the following:

- ii) Otherwise,
 - 1) If a <datetime value> does not conform to the natural rules for intervals according to the Gregorian calendar, an exception condition is raised: *data exception—invalid interval format*.
 - 2) Otherwise, an exception condition is raised: *data exception—invalid character value for cast*.

6.12 <numeric value expression>

1. *Rationale: Editorial.*

In General Rule 2), replace "then the value of" with "then the result of".

6.14 <datetime value expression>

1. *Rationale: The expression "X AT TIME ZONE Y - Z" may be parsed as either "(X AT TIME ZONE Y) - Z" or "X AT TIME ZONE (Y - Z)". The following clarifies that it should be the former.*

Replace the Format for <time zone specifier> with the following:

<time zone specifier>::=
LOCAL
| TIME ZONE <interval primary>

2. *Rationale: Editorial.*

In Syntax Rule 4), replace "<interval value expression>" with "<interval primary>".

3. *Rationale: Clarify the meaning of the list of data types given in Syntax Rule 5).*

Add a Note following Syntax Rule 5) b) as follows:

Note: In the preceding Syntax Rule, TIME does not include TIME WITH TIME ZONE and TIMESTAMP does not include TIMESTAMP WITH TIME ZONE.

4. *Rationale: If <time zone> is implied, then <time zone specifier> shall not be specified.*

In General Rule 2), replace "specified or implied" with "specified".

5. *Rationale: Editorial.*

In General Rule 2), replace "<interval value expression>" with "<interval primary>".

In General Rule 4) b), replace "<interval value expression>" with "<interval primary>".

6.15 <interval value expression>

1. *Rationale: Clarification of the data type of a result.*

Replace Syntax Rule 3) with the following:

- 3) Case:
 - a) If the <interval value expression> simply contains an <interval qualifier> *IQ*, then the data type of the result is
INTERVAL *IQ*
 - b) If the <interval value expression> is an <interval term>, then the result of the <interval value expression> contains the same interval fields as the <interval primary>
 - c) If <interval term.1> is specified, then the result contains every interval field that is contained in the result of either <interval value expression 1> or <interval term 1>, and, if both contain a seconds field, then the fractional seconds precision of the result is the greater of the two fractional seconds precisions.

Note: Interval fields are effectively defined by Table 5 - Fields in year-month INTERVAL items, and Table 6 Fields in day-time INTERVAL items.

7.1 <row value constructor>

1. *Rationale: To clarify the value of a <default specification> by reference to the General Rules of 11.5 <default clause> as is also done in 13.9.*

Replace General Rule 2) with the following:

- 2) The value of a <default specification> is determined according to the General Rules of Subclause 11.5, "<default clause>".

2. *Rationale: Editorial.*

In Leveling Rule 1) a), replace "<table value constructor>" with "<table value constructor> or an <overlaps predicate>".

3. *Rationale: The 1989 standard prohibited the use of a <row subquery> in a predicate. A leveling rule was omitted that would only allow this use of <row subquery> in Full level.*

Add the following to Leveling Rule 1):

- a.1) A <row value constructor> shall not be a <row subquery>.

7.4 <from clause>

1. Rationale: Editorial.

In Syntax Rule 1)a), replace "with no intervening" with "without an intervening".

In Syntax Rule 1)b), replace "with no intervening" with "without an intervening".

In General Rule 1)a), replace "with no intervening" with "without an intervening".

In General Rule 1)b), replace "with no intervening" with "without an intervening".

2. Rationale: Editorial.

In Leveling Rule 2) a), replace "<table name>" with "<table reference>".

7.5 <joined table>

1. Rationale: Syntax Rule 6) [If NATURAL ...] does not adequately address the following:

- i) the case when the set of corresponding columns is empty,
- ii) the cases when either set of non-corresponding columns is empty,
- iii) the case when a set of non-corresponding join columns contains columns with the same <column name>.

Syntax Rule 6) also conflicts with Syntax Rules 7), 8) and 9) with respect to the definition of the nullability characteristics of a column.

In Syntax Rule 6) d), replace "Let" by "If there is at least one corresponding join column, then let"

Replace Syntax Rule 6) e) with the following.

- e) 1) If T_1 contains at least one column that is not a corresponding join column, then let SLT_1 be a <select list> of <derived column>s of the form:
 $TA.C$
for every column C of T_1 that is not a corresponding join column, taken in order of their ordinal position in T_1 .
- 2) If T_2 contains at least one column that is not a corresponding join column, then let SLT_2 be a <select list> of <derived column>s of the form:
 $TB.C$
for every column C of T_2 that is not a corresponding join column, taken in order of their ordinal position in T_2 .

Replace Syntax Rule 6) f) with the following:

- f) Let the <select list> SL be defined as:
Case:
 - i) If all the columns of T_1 and T_2 are corresponding join columns, then let SL be $SLCC$.
 - ii) If T_1 contains no corresponding join columns and T_2 contains no corresponding join columns, then let SL be SLT_1, SLT_2 .
 - iii) If T_1 contains no columns other than corresponding join columns, then let SL be $SLCC, SLT_2$.

- iv) If T_2 contains no columns other than corresponding join columns, then let SL be $SLCC$, SLT_1
- v) Otherwise, let SL be $SLCC$, SLT_1 , SLT_2

The descriptors of the columns of the result of the <joined table>, with the possible exception of the nullability characteristics of the column, are the same as the descriptors of the columns of the result of
 SELECT SL FROM TR_1 , TR_2

2. *Rationale: General Rule 6) a) [If NATURAL is ...] uses the <select list> definitions $SLCC$, SLT_1 and SLT_2 which are not applicable to the table SN .*

Replace General Rule 6) a) with the following:

- a) If NATURAL is specified or a <named columns join> is specified, then:
 - i) Let CS_i be a name for the i -th column of S . Column CS_i of S corresponds to the i -th column of T_1 if i is less than or equal to D_1 . Column CS_j of S corresponds to the $(j-D_1)$ -th column of T_2 for j greater than D_1 .
 - ii) Let SLN be the <select list> derived from SL by replacing each <derived column> of the form
 $TA.C$ or $TB.C$
 in SL by the name CS_i of the column of S that corresponds to the column C of T_1 or T_2 , respectively.
 - iii) The result of the <joined table> is the multiset of rows resulting from
 SELECT SLN FROM SN

7.6 <where clause>

1. *Rationale: Remove the ambiguity in Leveling Rule 2).*

Replace Leveling Rule 2) with:

- 2) The following restrictions apply for Entry SQL in addition to any Intermediate SQL restriction:
 - a) A <value expression> directly contained in the <search condition> shall not contain a <column reference> that references a <derived column> that generally contains a <set function specification>.

7.9 <query specification>

1. *Rationale: The rules for expansion of "<qualifier>.*" need to limit the expansion to the table identified by the <qualifier>, rather than including the entire <table expression>.*

Replace Syntax Rule 4) with the following:

- 4) If the <select sublist> "<qualifier>.*" is specified, then let Q be the <qualifier> of that <select sublist>. Q shall be a <table name> or <correlation name> exposed by a <table reference> immediately contained in the <from clause> of T . Let TQ be the table associated with Q . That <select sublist> is equivalent to a <value expression> sequence in which each <value expression> is a <column reference> CR that references a column of TQ that is not a common column of a <joined table>. Each column of TQ that is not a common column of a <joined table> shall be referenced exactly once. The columns shall be referenced in the ascending order of their ordinal positions with TQ .
Note: common column of a <joined table> is defined in Subclause 7.5, "<joined table>".

2. *Rationale: Clarification of the definition of outer reference.*

Replace Syntax Rule 7) with the following:

- 7) If *T* is a grouped table, then in each <value expression>, each <column reference> that references a column of *T* shall reference a grouping column or be specified within a <set function specification>. If *T* is not a grouped table and any <value expression> contains a <set function specification> that contains a reference to a column of *T* or any <value expression> directly contains a <set function specification> that does not contain an outer reference, then in each <value expression>, each <column reference> that references a column of *T* shall be specified within a <set function specification>.

3. *Rationale: Editorial.*

In Syntax Rule 9) a), replace the two occurrences of "C" with "CN".

In Syntax Rule 9) b), replace "C" with "the <column name> of the column designated by the <column reference>".

4. *Rationale: A <dynamic parameter specification> is possibly nullable.*

In Syntax Rule 10), insert " a <dynamic parameter specification>," after "an <indicator variable>,".

5. *Rationale: Editorial.*

In General Rule 1) a) ii) 1), replace "the table is the result of the <query specification>" with "the result of the <query specification> is that table".

In General Rule 1) b) ii) 1), replace "the table is the result of the <query specification>" with "the result of the <query specification> is that table".

7.10 <query expression>

1. *Rationale: Editorial.*

Replace Syntax Rule 14) with the following:

- 14) The *simply underlying tables* of a <query expression> are the tables identified by those <table name>s, <query specification>s, and <derived table>s contained in the <query expression> without an intervening <derived table> or an intervening <join condition>.

7.11 <scalar subquery>, <row subquery>, and <table subquery>

1. *Rationale: Editorial.*

Replace Leveling Rule 1) with the following:

- 1) The following restrictions apply for Intermediate SQL:

None.

2. *Rationale: SQL-89 prohibited the use of a subquery with degree greater than one as the argument of an <exists predicate>. It was intended to relax this restriction in Intermediate SQL, but the required Leveling Rule for Entry SQL was inadvertently omitted.*

Add the following Leveling Rule:

- b.1) If a <table subquery> is simply contained in an <exists predicate>, then the <select list> of the <query specification> directly contained in the <table subquery> shall comprise either an <asterisk> or a single <derived column>.

8.3 <between predicate>

1. *Rationale: Editorial.*

In Syntax Rule 2), replace "two" with "three".

8.7 <quantified comparison predicate>

1. *Rationale: Editorial.*

In Syntax Rule 2), replace "columns" with "corresponding columns".

9.1 Retrieval assignment

1. *Rationale: Clarification of the treatment of datetimes.*

Replace General Rule 3) 1) with the following:

- 1) If the data type of *T* is datetime, then:
Case:
 - i) If *V* is a member of the data type of *T*, then *T* is set to *V*.
 - ii) If a member of the data type of *T* can be obtained from *V* by rounding, then *T* is set to that value.
 - iii) Otherwise, an exception condition is raised: *data exception—datetime field overflow*.

9.2 Store assignment

1. *Rationale: Clarification of the treatment of datetimes.*

Replace General Rule 3) l) with the following:

- l) If the data type of *T* is datetime, then:

Case:

- i) If *V* is a member of the data type of *T*, then *T* is set to *V*.
- ii) If a member of the data type of *T* can be obtained from *V* by rounding, then *T* is set to that value.
- iii) Otherwise, an exception condition is raised: *data exception—datetime field overflow*.

10.1 <interval qualifier>

1. *Rationale: Editorial.*

Replace "Syntax rules" with "Syntax Rules".

2. *Rationale: Clarification of the definition of "significance" of fractional seconds precision.*

Replace Syntax Rule 1) with:

- 1) There is a significance of ordering of <datetime field>s. In order from most significant to least significant, the ordering is: YEAR, MONTH, DAY, HOUR, MINUTE, and SECOND. A <start field> or <single datetime field> with an <interval leading field precision> *i* is more significant than a <start field> or <single datetime field> with an <interval leading field precision> *j* if *i* < *j*. An <end field> or <single datetime field> with an <interval fractional seconds precision> *i* is less significant than an <end field> or <single datetime field> with an <interval fractional seconds precision> *j* if *i* > *j*.
- 3. *Rationale: There is a contradiction between Syntax Rule 2) a) and General Rule 4. The Syntax Rule requires the <start field> to be more significant than the <end field> while the General Rule allows them to be the same. The Syntax Rule is modified to remove the contradiction.*

Replace Syntax Rule 2) with the following:

- 2) If TO is specified, then:
 - a) <start field> shall be more significant than, or equally significant to, the <end field>.
 - b) if <start field> specified MONTH, then <end field> shall specify MONTH, and
 - c) if <start field> specified YEAR, then <end field> shall specify either YEAR or MONTH.

10.4 <character set specification>

1. *Rationale: Clarify that references to non-existing objects is only allowed within the same <schema definition>.*

Add the following Syntax Rule:

- 3.1) Let *C* be the <character set name> contained in the <character set specification>. If the <character set specification> is not contained in a <schema definition>, then the schema identified by the explicit or implicit qualifier of the <character set name> shall include the descriptor of *C*. If the <character set specification> is contained in a <schema definition> *S*, then the schema identified by the explicit or implicit qualifier of the <character set name> shall include the descriptor of *C* or *S* shall include a <schema element> that creates the descriptor of *C*.

10.5 <collate clause>

1. *Rationale: Clarify that references to non-existing objects are only allowed within the same <schema definition>.*

Add the following Syntax Rule:

- 0.1) Let *C* be the <collation name> contained in the <collate clause>. If the <collate clause> is not contained in a <schema definition>, then the schema identified by the explicit or implicit qualifier of the <collation name> shall include the descriptor of *C*. If the <collate clause> is contained in a <schema definition> *S*, then the schema identified by the explicit or implicit qualifier of the <collation name> shall include the descriptor of *C*, or *S* shall include a <schema element> that creates the descriptor of *C*.

10.6 <constraint name definition> and <constraint attributes>

1. *Rationale: Editorial.*

In the **Note** following General Rule 3), replace "<SQL statement>" with "SQL-statement".

2. *Rationale: Editorial.*

In Leveling Rule 2) a), replace "Intermediate" with "Entry".

11.1 <schema definition>

1. *Rationale: Editorial.*

In Leveling Rule 2) a), replace "Intermediate" with "Entry".

11.2 <drop schema statement>

1. *Rationale: Editorial.*

In General Rule 5), replace "collation definition" with "collating sequence".

11.4 <column definition>

1. *Rationale: Clarify that references to non-existing objects is only allowed within the same <schema definition>.*

Add the following Syntax Rule:

- 4.1) If the <column definition> is not contained in a <schema definition>, then the schema identified by the explicit or implicit qualifier of the <domain name> shall include the descriptor of *D*. If the <column definition> is contained in a <schema definition> *S*, then the schema identified by the explicit or implicit qualifier of the <domain name> shall include the descriptor of *D*, or *S* shall include a <schema element> that creates the descriptor of *D*.

2. *Rationale: Editorial.*

In Access Rule 1), replace "*D*." with "*D*".

3. *Rationale: Editorial.*

In Leveling Rule 2) c), replace "Intermediate" with "Entry".

11.5 <default clause>

1. *Rationale: Editorial.*

In Syntax Rule 1), replace "<alter domain definition>" with "<alter domain statement>".

2. *Rationale: The following changes correct the fact that the current rules cause the value of, for example, CURRENT_USER to be evaluated at the time of the <table definition> or <column definition> and forever more used as the default value for the associated column in the table. The value of CURRENT_USER, etc., and the <datetime value function>s are being put directly into the column or domain descriptor at the time that the <default clause> is processed. It was intended that those things be evaluated at the time that the row was inserted into the table (or updated in the table). Additionally the new rule 2.1 defines how a default value of null is represented in COLUMN_DEFAULT in the DOMAINS and COLUMNS base table. The new Case c) of General Rule 2) covers the case of a derived table.*

Replace General Rules 1) and 2) with the following General Rules:

- 1) The default value inserted in the column descriptor, if the <default clause> is to apply to a column, or in the domain descriptor, if the <default clause> is to apply to a domain, is the <default option>. If the subject data type is bit string with fixed length, the <default clause> specifies a <bit string literal>, and the length of the <bit string literal> is less than the fixed length of the column, then a completion condition is raised: *warning—implicit zero-bit padding*.
- 2) The default value of a column is
Case:
 - a) If the column descriptor of a column indicates a default value derived from a <default option>, then the value in the column descriptor;
 - b) If the column descriptor includes a domain name that identifies a domain descriptor that includes a default value derived from a <default option>, then the value in the domain descriptor;

- c) If the default value is for a column *C* of a candidate row for insertion into or update of a derived table *DT* and *C* has a single counterpart column *CC* in a leaf generally underlying table of *DT*, then the default value of *CC* obtained by applying the General Rules of this Subclause.
 - d) Otherwise, the null value.
- 2.1) When a default value is required for a column in a row of a table, the default value for the column is derived from the default value in the appropriate column descriptor or domain descriptor. Let *D* be that descriptor.

Case:

- a) If *D* contains NULL, then the null value.
- b) If *D* contains a <literal>, then

Case:

- i) If the subject data type is numeric, then the numeric value of the <literal>.
 - ii) If the subject data type is character string with variable length, then the value of the <literal>.
 - iii) If the subject data type is character string with fixed length, then the value of the <literal>, extended as necessary on the right with <space>s to the length in characters of the subject data type.
 - iv) If the subject data type is bit string with variable length, then the value of the <literal>.
 - v) If the subject data type is bit string with fixed length, then the value of the <literal> extended as necessary on the right with 0-valued bits to the length of the subject data type.
 - vi) If the subject data type is datetime or interval, then the value of the <literal>.
- c) If the column or domain descriptor specifies CURRENT_USER, SESSION_USER, or SYSTEM_USER, then

Case:

- i) If the subject data type is character string with variable length, then the value obtained by an evaluation of CURRENT_USER, SESSION_USER, or SYSTEM_USER at the time that the default value is required.
 - ii) If the subject data type is character string with fixed length, then the value obtained by an evaluation of CURRENT_USER, SESSION_USER, or SYSTEM_USER at the time that the default value is required, extended as necessary on the right with <space>s to the length in characters of the subject data type.
- d) If the column or domain descriptor contains a <datetime value function>, then the value of an evaluation of the <datetime value function> at the time that the default value is required.
- e) Otherwise, the null value.

3. *Rationale: The following helps define the action taken when a default value cannot be represented in the Information Schema.*

Add the following General Rule:

- 3.1) If the <default clause> is contained in an <schema definition>, and if the character representation of the <default option> cannot be represented in the Information Schema without truncation, then a completion indication is raised: *warning—default option too long for information schema*.

11.6 <table constraint definition>

1. *Rationale: Editorial.*

In Leveling Rule 2) a), replace "Intermediate" with "Entry".

11.8 <referential constraint definition>

1. *Rationale: Clarify that references to non-existing objects are only allowed within the same <schema definition>.*

Add the following Syntax Rule:

- 9.1) Let *T* be the referenced table. If the <referential constraint definition> is not contained in a <schema definition>, then the schema identified by the explicit or implicit qualifier of the <table name> shall include the descriptor of *T*. If the <referential constraint definition> is contained in a <schema definition> *S*, then the schema identified by the explicit or implicit qualifier of the <table name> shall include the descriptor of *T*, or *S* shall include a <schema element> that creates the descriptor of *T*.

11.9 <check constraint definition>

1. *Rationale: Editorial.*

Replace Syntax Rule 1) with the following:

- 1) The <search condition> shall not contain a <parameter specification>, a <variable specification>, or a <dynamic parameter specification>.

11.11 <add column definition>

1. *Rationale: Editorial.*

Add the following General Rule:

- 0.1) Let *T* be the table identified by the <table name> immediately contained in the containing <alter table statement>.