



INTERNATIONAL STANDARD ISO/IEC 9075-9:2001 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 9: Management of External Data (SQL/MED)

TECHNICAL CORRIGENDUM 1

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

Partie 9: Gestion des données externes (SQL/MED)

RECTIFICATIF TECHNIQUE 1

ITeH STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 9075-9:2001/Cor 1:2003](#)

Technical Corrigendum to ISO/IEC 9075-9:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 32, Data management and interchange.

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 numbering:

Where this Corrigendum introduces new Syntax, Access, General and Conformance Rules, the new rules have been numbered as follows:

Rules inserted between, for example, Rules 7) and 8) 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.

Where this Corrigendum introduces new Subclauses, the new subclauses have been numbered as follows:

Subclauses inserted between, for example, Subclause 4.3.2 and 4.3.3 are numbered 4.3.2a, 4.3.2b, etc.

Those inserted before, for example, 4.3.1 are numbered 4.3.0, 4.3.0a, etc.

Contents

	Page
4.5 User mappings	3
4.8 Datalinks	3
4.9 Type conversions and mixing of data types	4
4.18.1 Handles	4
4.18.4 Return codes	4
4.18.7 Foreign-data wrapper descriptor areas	4
5.2 <token> and <separator>	5
6.2 <column reference>	5
6.5 <datalink value>	6
7.1 <table reference>	6
13.1 <foreign server definition>	7
13.4 <foreign-data wrapper definition>	8
14.1 <revoke statement>	8
14.3 <alter user mapping>	9
18.4 <describe statement>	9
18.6 <output using clause>	9
22.9 Tables used with SQL/MED	9
23.2 <foreign-data wrapper interface routine> invocation	10
23.3.1 AllocWrapperEnv	10
23.3.3 ConnectServer	10
23.3.18 InitRequest	10
23.3.20 Open	11
23.3.22 TransmitRequest	12
23.4.1 AllocDescriptor	13
23.4.2 FreeDescriptor	13
23.4.4 GetDescriptor	13
23.4.36 SetDescriptor	14
23.5.1 GetDiagnostics	16
25.1 ATTRIBUTES view	16
25.2 COLUMN_OPTIONS view	17
25.3 COLUMNS view	17
25.4 FOREIGN_DATA_WRAPPER_OPTIONS view	18
25.5 FOREIGN_DATA_WRAPPERS view	18
25.6 FOREIGN_SERVER_OPTIONS view	19
25.7 FOREIGN_SERVERS view	19
25.10 USER_MAP_OPTIONS view	20
25.12 Short name views	20
Annex B Implementation-defined elements	21
Annex C Implementation-dependent elements	22
Annex E Incompatibilities with ISO/IEC 9075:1992	22

iTech STANDARD PREVIEW
(standards.itech.ai)

Information technology — Database languages — SQL —

Part 9:

Management of External Data (SQL/MED)

TECHNICAL CORRIGENDUM 1

4.5 User mappings

1. *Rationale: Do not describe non-schema objects as being in a catalog.*

Replace the 2nd paragraph with:

A user mapping is defined by invoking an <user mapping definition>. Invocation of an <user mapping definition> results in the creation of a user mapping descriptor in the SQL-environment. A user mapping descriptor consists of:

- An authorization identifier.
- A foreign server name, identifying a foreign server descriptor.
- A generic options descriptor.

4.8 Datalinks

[ISO/IEC 9075-9:2001/Cor 1:2003](https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-db80553af42e/iso-iec-9075-9-2001-cor-1-2003)

<https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-db80553af42e/iso-iec-9075-9-2001-cor-1-2003>

1. *Rationale: Clarify the semantics of built-in functions.*

Replace the 2nd paragraph with:

The File Reference of a datalink is accessible by invoking operators defined in this part of ISO/IEC 9075. The character set of the File Reference, referred to as the *datalink character set* is implementation-defined.

Replace the 7th paragraph with:

With the function provided by datalinks and the datalinker, it is possible to specify that access to the files should be mediated by the SQL-server rather than by the external data manager. When access to the files is mediated by an SQL-server, any request to access a file must operate on an SQL-mediated datalink to obtain a character string with which to reference the file, using one of the operators provided for that purpose. This character string is constructed by combining the File Reference of a datalink value with an encrypted value called an *access token*. The generation of the access token and the method of combining it with the File Reference is implementation-dependent. When the application uses the returned character string value to access a file, the datalinker checks to see if the access token is *valid*. If it is valid, then the application is allowed to access the file pointed to by the File Reference. Every attempt by an application to access, without a valid access token, a file referenced by an SQL-mediated datalink is unsuccessful. The time at which a valid access token ceases to be valid is implementation-defined.

2. *Rationale: Clarify assignable and comparable.*

Insert the following paragraph after the 7th paragraph:

Datalinks are not comparable. A datalink is assignable only to sites of type DATALINK.

4.9 Type conversions and mixing of data types

1. *Rationale: Clarify assignable and comparable.*

Delete the subclause.

4.18.1 Handles

1. *Rationale: Editorial.*

In the 2nd paragraph, replace the 4th bullet with:

- **Request handle:** This handle is allocated by the SQL-server to reference an SQL-statement that is to be executed by a foreign server. A request handle may reference a simple statement, such as `SELECT * FROM T`, or it may reference a complex statement that includes predicates, joins, ordering, *etc.* A request handle is used by the foreign-data wrapper to retrieve (for example) the names of foreign tables referenced in the from clause, the names of column references in the select list, *etc.*, using foreign-data wrapper interface SQL-server routines. This handle is allocated implicitly.

4.18.4 Return codes

<https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-db80553af42e/iso-iec-9075-9-2001-cor-1-2003>

1. *Rationale: Add missing text for Success with Information condition.*

Insert the following into the 1st paragraph as the second bullet:

- A value of 1 (one) indicates **Success with information**. The foreign-data wrapper interface routine executed successfully but a completion condition was raised: *warning*.

4.18.7 Foreign-data wrapper descriptor areas

1. *Rationale: Editorial - misspelled acronym.*

Modify the 2nd bullet of the 5th paragraph with:

- **Wrapper Row Descriptor (WRD):** This descriptor is allocated by the SQL-server if a foreign-data wrapper requests its allocation. It is used to describe the result of a statement to be executed by that foreign-data wrapper in pass-through mode, and is associated with an ExecutionHandle. The foreign-data wrapper uses the `SetDescriptor()` routine to set information in the WRD. The SQL-server can obtain the handle to a WRD by invoking the `GetWRDHandle()` routine. It can then retrieve the information in that WRD by invoking the `GetDescriptor()` routine.

5.2 <token> and <separator>

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

In the Format, replace the productions for <non-reserved word> and <reserved word> with:

```

<non-reserved word> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !!All alternatives from ISO/IEC 9075-5
    | BLOCKED
    | CONTROL
    | DB
    | FILE | FS
    | INTEGRITY
    | LIBRARY | LIMIT | LINK
    | MAPPING
    | PASSTHROUGH | PERMISSION
    | RECOVERY | RESTORE
    | SELECTIVE | SERVER
    | UNLINK
    | VERSION
    | WRAPPER
    | YES

<reserved word> ::=
    !! All alternatives from ISO/IEC 9075-2
    | !!All alternatives from ISO/IEC 9075-5
    | DATALINK | DLURLCOMPLETE | DLURLPATH | DLURLPATHONLY | DLURLSCHEME
    | DLURLSERVER | DLVALUE
    | IMPORT

```

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 9075-9:2001/Cor.1:2003
<https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-db80553af42e/iso-iec-9075-9-2001-cor-1-2003>

6.2 <column reference>

1. *Rationale: there is no <row value expression> immediately contained in a <set clause>.*

Replace Access Rule 1) with:

- 1) Replace AR 1) If CR is a <column reference> whose qualifying table is a base table, a foreign table or a viewed table and that is contained in any of:
- A <query expression> simply contained in a <cursor specification>, a <view definition> or an <insert statement>.
 - A <sort specification list> contained in a <cursor specification>.
 - A <table expression> immediately contained in a <select statement: single row>.
 - A <search condition> immediately contained in a <trigger definition>, a <delete statement: searched> or an <update statement: searched>.
 - A <select list> immediately contained in a <select statement: single row>.
 - A <value expression> simply contained in an <update source>.

then let *C* be the column referenced by *CR*.

Case:

- a) If <column reference> is contained in an <SQL schema statement>, then the applicable privileges of the <authorization identifier> that owns the containing schema shall include SELECT for *C*.
- b) Otherwise, the current privileges shall include SELECT on *C*.

NOTE 24 – “applicable privileges” and “current privileges” are defined in Subclause 11.1, “<privileges>”.

6.5 <datalink value>

Replace Syntax Rule 4) with:

- 4) The character set name, collating sequence, and coercibility characteristic of the File Reference of the result of valuating a the <datalink value constructor> are the character set name, collation, and coercibility characteristic, respectively, of the <data location>.

7.1 <table reference>

ITeh STANDARD PREVIEW
(standards.iteh.ai)

1. *Rationale:* There is no <row value expression> immediately contained in a <set clause>.

Replace Access Rule 1) b) with: [ISO/IEC 9075-9:2001/Cor 1:2003](https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-d80553af42a/iso-iec-9075-9-2001-cor-1-2003)
[https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-](https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-d80553af42a/iso-iec-9075-9-2001-cor-1-2003)

- 1) b) If *T* is a base table, a foreign table or a viewed table and the <table reference> is contained in any of:
 - A <query expression> simply contained in a <cursor specification>, a <view definition>, or an <insert statement>.
 - A <table expression> or <select list> immediately contained in a <select statement: single row>.
 - A <search condition> immediately contained in a <delete statement: searched> or an <update statement: searched>.
 - A <value expression> simply contained in an <update source>.

then

Case:

- i) If <table reference> is contained in an <SQL schema statement> then, the applicable privileges of the <authorization identifier> that owns the containing schema shall include SELECT on at least one column of *T*.

ii) Otherwise, the current privileges shall include SELECT on at least one column of *T*.

NOTE 29 — “applicable privileges” and “current privileges” are defined in Subclause 11.1, “<privileges>”.

2. *Rationale: Delete incorrect implication that there will only every be 1 TRH.*

Replace General Rule 1) b) ix) with:

- 1) b) ix) Let *TRDH* be the TableReferenceDescriptorHandle allocated for *TRD*.

3. *Rationale: Clarify when a descriptor handle is used instead of the descriptor itself.*

Replace General Rules 1) b) xvii) and 1) b) xviii) with:

- 1) b) xvii) Let *NC* be the value of the COUNT descriptor field that would be returned by invocation of the `GetDescriptor()` routine with *TRDH* as the DescriptorHandle parameter, 0 (zero) as the RecordNumber parameter, and the code for COUNT from Table 32, “Codes used for foreign-data wrapper descriptor fields”, as the FieldIdentifier parameter.

- 1) b) xviii) Let DT_j be the effective data type of the j -th column, for $1 \text{ (one)} \leq j \leq NC$, as represented by the values of the TYPE, LENGTH, OCTET_LENGTH, PRECISION, SCALE, DATETIME_INTERVAL_CODE, DATETIME_INTERVAL_PRECISION, CHARACTER_SET_CATALOG, CHARACTER_SET_SCHEMA, CHARACTER_SET_NAME, USER_DEFINED_TYPE_CATALOG, USER_DEFINED_TYPE_SCHEMA, USER_DEFINED_TYPE_NAME, SCOPE_CATALOG, SCOPE_SCHEMA, and SCOPE_NAME fields that would be returned by separate invocations of the `GetDescriptor()` routine with *TRDH* as the DescriptorHandle parameter, j as the RecordNumber parameter, and the code for the fields TYPE, LENGTH, OCTET_LENGTH, PRECISION, SCALE, DATETIME_INTERVAL_CODE, DATETIME_INTERVAL_PRECISION, CHARACTER_SET_CATALOG, CHARACTER_SET_SCHEMA, CHARACTER_SET_NAME, USER_DEFINED_TYPE_CATALOG, USER_DEFINED_TYPE_SCHEMA, USER_DEFINED_TYPE_NAME, SCOPE_CATALOG, SCOPE_SCHEMA, and SCOPE_NAME from Table 32 “Codes used for foreign-data wrapper descriptor fields”, as the FieldIdentifier parameter.

4. *Rationale: Correct undefined tag.*

Replace General Rule 1) b) xxvi) with:

- 1) b) xxvi) The `<table reference>` references the table that consists of every row returned by the repeated invocation of the `Iterate()` routine in the library identified by *WRLN* with *EXH* as the argument until the return code indicates **No data found**.

13.1 <foreign server definition>

1. *Rationale: Delete the optional AUTHORIZATION-clause from <foreign server definition>.*

In the Format replace the production for <foreign server definition> with:

```
<foreign server definition> ::=
  CREATE SERVER <foreign server name>
  [ TYPE <server type> ]
  [ VERSION <server version> ]
  FOREIGN DATA WRAPPER <foreign-data wrapper name>
  [ <generic options> ]
```

Delete Syntax Rule 3).

Replace General Rule 1) e) with:

- 1) e) The current authorization identifier.

Replace General Rule 2) with:

- 2) A privilege descriptor is created that defines the USAGE privilege on this foreign server to the current authorization identifier. The grantor of the privilege descriptor is set to the special grantor value “_SYSTEM”. This privilege is grantable.

13.4 <foreign-data wrapper definition>

1. *Rationale: Delete the optional AUTHORIZATION-clause from <foreign-data wrapper definition>.*

In the Format replace the production for <foreign-data wrapper definition> with:

```
<foreign-data wrapper definition> ::=  
CREATE FOREIGN DATA WRAPPER <foreign-data wrapper name>  
  [ <library name specification> ]  
  <language clause>  
  [ <generic options> ]
```

Delete Syntax Rule 3).

Replace General Rule 1) b) with:

- 1) b) The current authorization identifier.

Replace General Rule 2) with:

- 2) A privilege descriptor is created that defines the USAGE privilege on this foreign-data wrapper to the current authorization identifier. The grantor of the privilege descriptor is set to the special grantor value “_SYSTEM”. This privilege is grantable.

14.1 <revoke statement>

1. *Rationale: Cater properly for abandoned foreign table descriptors.*

Replace Syntax Rule 1) with:

- 1) Insert after SR 21) Let *T* be any foreign table descriptor included in *SI*. *T* is said to be *abandoned* if the revoke destruction action would result in *AI* no longer having USAGE privilege on the foreign server associated with the foreign table described by *T*.

Replace Syntax Rule 3) with:

- 3) Augment SR 36) Add abandoned server descriptor, and abandoned foreign table descriptor to the list of objects that shall not exist.

Insert the following General Rule:

- 2) Insert this GR For every abandoned foreign table descriptor *FT*, let *FTN* be the <table name> of *FT*. The following <drop foreign table statement> is effectively executed without further Access Rule checking:

```
DROP FOREIGN TABLE S1.FTN CASCADE
```

14.3 <alter user mapping>

1. *Rationale: Correct symbol.*

Replace General Rule 1) with:

- 1) The General Rules of Subclause 11.3, “<alter generic options>”, are applied to *AGO* with the generic options descriptor included in *UMD* as the applicable generic options descriptor.

18.4 <describe statement>

1. *Rationale: Clarify that wrapper row descriptor is meant by WRD.*

Replace General Rule 1) a) with:

- 1) a) Let *EXH* be the ExecutionHandle associated with <SQL statement name>. Let *WPD* and *WRD* be the wrapper parameter descriptor and wrapper row descriptor, respectively, associated with the *WPDHandle* and *WRDHandle*, respectively, that would be returned by the invocation of the *GetWPDHandle()* and *GetWRDHandle()* routines with *EXH* as the ExecutionHandle parameter.

18.6 <output using clause>

1. *Rationale: Clarify that wrapper row descriptor is meant by WRD, and server row descriptor by SRD.*

Replace General Rule 1) c) with:

- 1) c) Let *EXH* be the ExecutionHandle associated with *SN*. Let *WRD* and *SRD* be the wrapper row descriptor and server row descriptor, respectively, associated with the *WRDHandle* and *SRDHandle*, respectively, that would be returned by the invocation of the *GetWRDHandle()* and *GetSRDHandle()* routines with *EXH* as the ExecutionHandle parameter.

22.9 Tables used with SQL/MED

1. *Rationale: Remove a row that had been forgotten to be removed when a new data retrieval architecture was introduced.*

Delete the following row from Table 33 — Codes used for foreign-data wrapper handle types

Handle type	Code

DataHandle	5
------------	---

23.2 <foreign-data wrapper interface routine> invocation

1. *Rationale: Add missing text for Success with Information condition.*

Insert the following General Rule:

- 4) a) ii) 1.1) If a completion condition is raised: *warning*, then *RC* is set to indicate **Success with information**.

2. *Rationale: Add missing text for Success with Information condition.*

Insert the following General Rule:

- 4) b) iii.1) If *RN* is a foreign-data wrapper interface wrapper routine, then the actions of the invoking SQL-server in response to the failed execution of *RN* are implementation-dependent.

23.3.1 AllocWrapperEnv

1. *Rationale: Make implementation-defined explicit.*

Replace General Rule 3) with:

- 3) If the implementation-defined maximum number of foreign-data wrapper environments that can be allocated at one time has already been reached, then an exception condition is raised: *FDW-specific condition — limit on number of handles exceeded*. A skeleton *FDW*-environment is allocated and is assigned a unique value that is returned in *WrapperEnvHandle*.

23.3.3 ConnectServer

1. *Rationale: Make implementation-defined explicit.*

Replace General Rule 13) with:

- 13) If the implementation-defined maximum number of *FS*-connections that can be allocated at one time has already been reached, then *FSConnectionHandle* is set to zero and an exception condition is raised: *FDWspecific condition — limit on number of handles exceeded*.

23.3.18 InitRequest

1. *Rationale: Make implementation-defined explicit.*

Replace General Rule 7) with:

- 7) If the implementation-defined maximum number of *FDW*-replies that can be allocated at one time has already been reached, then *ReplyHandle* is set to zero and an exception condition is raised: *FDW-specific condition — limit on number of handles exceeded*.

iTech STANDARD PREVIEW
(standards.itech.ai)

ISO/IEC 9075-9:2001/Cor.1:2003
<https://standards.itech.ai/catalog/standard/item?id=9075-9:2001/Cor.1:2003>
#90557#22#res-ix-9075-9:2001/Cor.1:2003

Replace General Rule 10) with:

- 10) If the implementation-defined maximum number of FDW-executions that can be allocated at one time has already been reached, then ExecutionHandle is set to zero and an exception condition is raised:
FDW-specific condition — limit on number of handles exceeded.

2. *Rationale: Clarify when a descriptor handle is used instead of the descriptor itself.*

Replace General Rule 14) with:

- 14) Let *NIDA* be the number of item descriptor areas that must be set up for the server row descriptor. Let *SRDHandle* be the DescriptorHandle that is returned by an invocation of the `AllocDescriptor()` routine with *NIDA* as the `MaxDetailAreas` parameter. Let *SRD* be the server row descriptor identified by *SRDHandle*. *SRD* is associated with the allocated FDW-execution. For this descriptor area, fields with non-blank entries in Table 36, “Foreign-data wrapper descriptor field default values”, are set to the specified default values by the invocation of the `SetDescriptor()` routine with *SRDHandle* as the DescriptorHandle parameter and *r* as the Record-Number parameter, 1 (one) $r \leq NIDA$, and the code for the fields with non-blank entries in Table 36, “Foreign-data wrapper descriptor field default values”, from Table 32, “Codes used for foreign-data wrapper descriptor fields”, as the FieldIdentifier parameter. All other fields in the item descriptor areas of *SRD* are initially undefined.

23.3.20 Open

iTeh STANDARD PREVIEW
(standards.iteh.ai)

1. *Rationale: Define previously undefined symbols.*

Insert the following General Rule: [ISO/IEC 9075-9:2001/Cor 1:2003](https://standards.iteh.ai/catalog/standards/sist/cb404ce8-0135-4d26-8490-488055a42c30/iso-9075-9-2001-cor-1-2003)

- 5) a.0) Let *SRD* be the SRDHandle that would be returned by an invocation of the `GetSRDHandle()` routine with *EH* as the ExecutionHandle parameter. Let *SPD* be the SPDHandle that would be returned by an invocation of the `GetSPDHandle()` routine with *EH* as the ExecutionHandle parameter. Let *WRD* be the WRDHandle that would be returned by an invocation of the `GetWRDHandle()` routine with *EH* as the ExecutionHandle parameter. Let *WPD* be the WPDHandle that would be returned by an invocation of the `GetWPDHandle()` routine with *EH* as the ExecutionHandle parameter.

2. *Rationale: Define previously undefined symbols.*

Replace General Rule 5) with:

- 5) c) Let TD_j be the effective data type of the *j*-th <target specification>, for 1 (one) $j \leq NCR$, as represented by the values of the TYPE, LENGTH, OCTET_LENGTH, PRECISION, SCALE, DATETIME_INTERVAL_CODE, DATETIME_INTERVAL_PRECISION, CHARACTER_SET_CATALOG, CHARACTER_SET_SCHEMA, CHARACTER_SET_NAME, USER_DEFINED_TYPE_CATALOG, USER_DEFINED_TYPE_SCHEMA, USER_DEFINED_TYPE_NAME, SCOPE_CATALOG, SCOPE_SCHEMA, and SCOPE_NAME fields that would be set by separate invocations of the `GetDescriptor()` routine with *SRD* as the DescriptorHandle parameter, *j* as the RecordNumber parameter, and the code for the fields TYPE, LENGTH, OCTET_LENGTH, PRECISION, SCALE, DATETIME_INTERVAL_CODE, DATETIME_INTERVAL_PRECISION, CHARACTER_SET_CATALOG, CHARACTER_SET_SCHEMA, CHARACTER_SET_NAME, USER_DEFINED_TYPE_CATALOG, USER_DEFINED_TYPE_SCHEMA,