5.4 Names and identifiers

Function

Specify names.

Format

```
<identifier> ::= <actual identifier>

<actual identifier> ::=
    <regular identifier>
  | <delimited identifier>
  | <Unicode delimited identifier>

<SQL language identifier> ::=
    <SQL language identifier start> [ <SQL language identifier part>... ]

<SQL language identifier start> ::= <simple Latin letter>

<SQL language identifier part> ::=
    <simple Latin letter>
  | <digit>
  | <underscore>

<authorization identifier> ::=
    <role name>
  | <user identifier>

<table name> ::= <local or schema qualified name>

<domain name> ::= <schema qualified name>

<schema name> ::= [ <catalog name> <period> ] <unqualified schema name>

<unqualified schema name> ::= <identifier>

<catalog name> ::= <identifier>

<schema qualified name> ::= [ <schema name> <period> ] <qualified identifier>

<local or schema qualified name> ::=
    [ <local or schema qualifier> <period> ] <qualified identifier>

<local or schema qualifier> ::=
    <schema name>
  | <local qualifier>

<qualified identifier> ::= <identifier>

<column name> ::= <identifier>

<correlation name> ::= <identifier>

<query name> ::= <identifier>
```

```
<SQL-client module name> ::= <identifier>

<procedure name> ::= <identifier>

<schema qualified routine name> ::= <schema qualified name>

<method name> ::= <identifier>

<specific name> ::= <schema qualified name>

<cursor name> ::= <local qualified name>

<local qualified name> ::= [ <local qualifier> <period> ] <qualified identifier>

<local qualifier> ::= MODULE

<host parameter name> ::= <colon> <identifier>

<SQL parameter name> ::= <identifier>

<constraint name> ::= <schema qualified name>

<external routine name> ::=
    <identifier>
  | <character string literal>

<trigger name> ::= <schema qualified name>

<collation name> ::= <schema qualified name>

<character set name> ::= [ <schema name> <period> ] <SQL language identifier>

<transliteration name> ::= <schema qualified name>

<transcoding name> ::= <schema qualified name>

<schema-resolved user-defined type name> ::= <user-defined type name>

<user-defined type name> ::= [ <schema name> <period> ] <qualified identifier>

<attribute name> ::= <identifier>

<field name> ::= <identifier>

<savepoint name> ::= <identifier>

<sequence generator name> ::= <schema qualified name>

<role name> ::= <identifier>

<user identifier> ::= <identifier>

<connection name> ::= <simple value specification>

<SQL-server name> ::= <simple value specification>

<connection user name> ::= <simple value specification>

<SQL statement name> ::=
    <statement name>
```



```
  | <extended statement name>

<statement name> ::= <identifier>

<extended statement name> ::= [ <scope option> ] <simple value specification>

<dynamic cursor name> ::=
    <cursor name>
  | <extended cursor name>

<extended cursor name> ::= [ <scope option> ] <simple value specification>

<descriptor name> ::= [ <scope option> ] <simple value specification>

<scope option> ::=
    GLOBAL
  | LOCAL

<window name> ::= <identifier>
```

Syntax Rules

1) In an <SQL language identifier>, the number of <SQL language identifier part>s shall be less than 128.

2) An <SQL language identifier> is equivalent to an <SQL language identifier> in which every letter that is a lower-case letter is replaced by the corresponding upper-case letter or letters. This treatment includes determination of equivalence, representation in the Information and Definition Schemas, representation in diagnostics areas, and similar uses.

   NOTE 76 — The Information Schema and Definition Schema are defined in ISO/IEC 9075-11.

3) An <SQL language identifier> (with every letter that is a lower-case letter replaced by the corresponding upper-case letter or letters), treated as the repetition of a <character string literal> that specifies a <character set specification> of SQL_IDENTIFIER, shall not be equal, according to the comparison rules in Subclause 8.2, "<comparison predicate>", to any <reserved word> (with every letter that is a lower-case letter replaced by the corresponding upper-case letter or letters), treated as the repetition of a <character string literal> that specifies a <character set specification> of SQL_IDENTIFIER.

   NOTE 77 — It is the intention that no <key word> specified in ISO/IEC 9075 or revisions thereto shall end with an <underscore>.

4) If a <local or schema qualified name> does not contain a <local or schema qualifier>, then

   Case:

   a) If the <local or schema qualified name> is contained, without an intervening <schema definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the default <unqualified schema name> for the SQL-session is implicit.

   b) If the <local or schema qualified name> is contained in a <schema definition>, then the <schema name> that is specified or implicit in the <schema definition> is implicit.

   c) Otherwise, the <schema name> that is specified or implicit for the SQL-client module is implicit.

5) Let *TN* be a <table name> with a <qualified identifier> *QI* and a <local or schema qualifier> *LSQ*.

Case:

a) If *LSQ* is "MODULE", then *TN* shall be contained in an <SQL-client module definition> *M* and the <module contents> of *M* shall contain a <temporary table declaration> *TT* whose <table name> has a <qualified identifier> equivalent to *QI*.

b) Otherwise, *LSQ* shall be a <schema name> that identifies a schema that contains a <table definition> or <view definition> whose <table name> has a <qualified identifier> equivalent to *QI*.

6) If a <cursor name> *CN* with a <qualified identifier> *QI* does not contain a <local qualifier>, then the <local qualifier> MODULE is implicit.

7) Let *CN* be a <cursor name> with a <qualified identifier> *QI* and a <local qualifier> *LQ*. *LQ* shall be "MODULE" and *CN* shall be contained in an <SQL-client module definition> whose <module contents> contain a <declare cursor> whose <cursor name> is *CN*.

8) If <user-defined type name> *UDTN* with a <qualified identifier> *QI* is specified, then

Case:

a) If *UDTN* is simply contained in <path-resolved user-defined type name>, then

Case:

i) If *UDTN* contains a <schema name> *SN*, then the schema identified by *SN* shall contain the descriptor of a user-defined type *UDT* such that the <qualified identifier> of *UDT* is equivalent to *QI*. *UDT* is the user-defined type identified by *UDTN*.

ii) Otherwise,

1) Case:

A) If *UDTN* is contained, without an intervening <schema definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then let *DP* be the SQL-path of the current SQL-session.

B) If *UDTN* is contained in a <schema definition>, then let *DP* be the SQL-path of that <schema definition>.

C) Otherwise, let *DP* be the SQL-path of the <SQL-client module definition> that contains *UDTN*.

2) Let *N* be the number of <schema name>s in *DP*. Let S_i, 1 (one) $\le i \le N$, be the *i*-th <schema name> in *DP*.

3) Let the *set of subject types* be the set containing every user-defined type *T* in the schema identified by some S_i, 1 (one) $\le i \le N$, such that the <qualified identifier> of *T* is equivalent to *QI*. There shall be at least one type in the set of subject types.

4) Let *UDT* be the user-defined type contained in the set of subject types such that there is no other type *UDT2* for which the <schema name> of the schema that includes the user-defined type descriptor of *UDT2* precedes in *DP* the <schema name> identifying the schema that includes the user-defined type descriptor of *UDT*. *UDTN* identifies *UDT*.

5) The implicit <schema name> of *UDTN* is the <schema name> of the schema that includes the user-defined type descriptor of *UDT*.

b) If *UDTN* is simply contained in <schema-resolved user-defined type name> and *UDTN* does not contain a <schema name>, then

Case:

i) If *UDTN* is contained, without an intervening <schema definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the implicit <schema name> of *UDTN* is the default <unqualified schema name> for the SQL-session.

ii) If *UDTN* is contained in a <schema definition>, then the implicit <schema name> of *UDTN* is the <schema name> that is specified or implicit in <schema definition>.

iii) Otherwise, the implicit <schema name> of *UDTN* is the <schema name> that is specified or implicit in <SQL-client module definition>.

9) Two <user-defined type name>s are equivalent if any only if they have equivalent <qualified identifier>s and equivalent <schema name>s, regardless of whether the <schema name>s are implicit or explicit.

10) No <unqualified schema name> shall specify DEFINITION_SCHEMA.

11) If a <transcoding name> does not specify a <schema name>, then INFORMATION_SCHEMA is implicit; otherwise, INFORMATION_SCHEMA shall be specified.

12) If a <character set name> does not specify a <schema name>, then

Case:

a) If <character set name> is not immediately contained in:

i) A <character set definition>.

ii) A <drop character set statement>.

then <schema name> INFORMATION_SCHEMA is implicit.

b) Otherwise,

Case:

i) If the <character set name> is contained, without an intervening <schema definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the default <unqualified schema name> for the SQL-session is implicit.

ii) If the <character set name> is contained in a <schema definition>, then the <schema name> that is specified or implicit in the <schema definition> is implicit.

iii) Otherwise, the <character set name> that is specified or implicit for the <SQL-client module definition> is implicit.

13) If a <schema qualified name> *SQN* other than a <transcoding name> does not contain a <schema name>, then

Case:

a) If any of the following is true:

i) *SQN* is is immediately contained in a <collation name> that is not immediately contained in a <collation definition> or in a <drop collation statement>.

ii) *SQN* is immediately contained in a <transliteration name> that is not immediately contained in a <transliteration definition> or in a <drop transliteration statement>.

then<schema name> INFORMATION_SCHEMA is implicit.

b) Otherwise,

Case:

i) If the <schema qualified name> is contained, without an intervening <schema definition>, in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the default <unqualified schema name> for the SQL-session is implicit.

ii) If the <schema qualified name> is contained in a <schema definition>, then the <schema name> that is specified or implicit in the <schema definition> is implicit.

iii) Otherwise, the <schema name> that is specified or implicit for the <SQL-client module definition> is implicit.

14) If a <schema name> does not contain a <catalog name>, then

Case:

a) If the <unqualified schema name> is contained in a <preparable statement> that is prepared in the current SQL-session by an <execute immediate statement> or a <prepare statement> or in a <direct SQL statement> that is invoked directly, then the default catalog name for the SQL-session is implicit.

b) If the <unqualified schema name> is contained in a <module authorization clause>, then an implementation-defined <catalog name> is implicit.

c) If the <unqualified schema name> is contained in a <schema definition> other than in a <schema name clause>, then the <catalog name> that is specified or implicit in the <schema name clause> is implicit.

d) If the <unqualified schema name> is contained in a <schema name clause>, then

Case:

i) If the <schema name clause> is contained in an <SQL-client module definition>, then the explicit or implicit <catalog name> contained in the <module authorization clause> is implicit.

ii) Otherwise, an implementation-defined <catalog name> is implicit.

e) Otherwise, the explicit or implicit <catalog name> contained in the <module authorization clause> is implicit.

15) Two <schema qualified name>s are equivalent if and only if their <qualified identifier>s are equivalent and their <schema name>s are equivalent, regardless of whether the <schema name>s are implicit or explicit.

16) Two <local or schema qualified name>s are equivalent if and only if their <qualified identifier>s are equivalent and either they both specify MODULE or they both specify or imply <schema name>s that are equivalent.

17) Two <character set name>s are equivalent if and only if their <SQL language identifier>s are equivalent and their <schema name>s are equivalent, regardless of whether the <schema name>s are implicit or explicit.

18) Two <schema name>s are equivalent if and only if their <unqualified schema name>s are equivalent and their <catalog name>s are equivalent, regardless of whether the <catalog name>s are implicit or explicit.

19) An <identifier> that is a <correlation name> is associated with a table within a particular scope. The scope of a <correlation name> is either a <select statement: single row>, <subquery>, or <query specification> (see Subclause 7.6, "<table reference>"), or is a <trigger definition> (see Subclause 11.39, "<trigger definition>"). Scopes may be nested. In different scopes, the same <correlation name> may be associated with different tables or with the same table.

20) No <authorization identifier> shall specify "PUBLIC".

21) Those <identifier>s that are valid <authorization identifier>s are implementation-defined.

22) Those <identifier>s that are valid <catalog name>s are implementation-defined.

23) The <data type> of <SQL-server name>, <connection name>, and <connection user name> shall be character string with an implementation-defined character set and shall have an octet length of 128 characters or less.

24) The <simple value specification> of <extended statement name> or <extended cursor name> shall not be a <literal>.

25) The declared type of the <simple value specification> of <extended statement name> shall be character string with an implementation-defined character set and shall have an octet length of 128 octets or less.

26) The declared type of the <simple value specification> of <extended cursor name> shall be character string with an implementation-defined character set and shall have an octet length of 128 octets or less.

27) The declared type of the <simple value specification> of <descriptor name> shall be character string with an implementation-defined character set and shall have an octet length of 128 octets or less.

28) In a <descriptor name>, <extended statement name>, or <extended cursor name>, if a <scope option> is not specified, then a <scope option> of LOCAL is implicit.

Access Rules

None.

General Rules

1) A <table name> identifies a table.

2) Within its scope, a <correlation name> identifies a table.

3) Within its scope, a <query name> identifies the table defined or returned by some associated <query expression body>.

4) A <column name> identifies a column.

5) A <domain name> identifies a domain.

6) An <authorization identifier> identifies a set of privileges.

7) An <SQL-client module name> identifies an SQL-client module.

8) A <schema qualified routine name> identifies an SQL-invoked routine.

9) A <method name> identifies an SQL-invoked method *M* whose descriptor is included in the schema that includes the descriptor of the user-defined type that is the type of *M*.

10) A <specific name> identifies an SQL-invoked routine.

11) A <cursor name> identifies a cursor.

12) A <host parameter name> identifies a host parameter.

13) An <SQL parameter name> identifies an SQL parameter.

14) An <external routine name> identifies an external routine.

15) A <trigger name> identifies a trigger.

16) A <constraint name> identifies a table constraint, a domain constraint, or an assertion.

17) A <catalog name> identifies a catalog.

18) A <schema name> identifies a schema.

19) A <collation name> identifies a collation.

20) A <character set name> identifies a character set.

21) A <transliteration name> identifies a character transliteration.

22) A <transcoding name> identifies a transcoding. All <transcoding name>s are implementation-defined.

23) A <connection name> identifies an SQL-connection.

24) A <user-defined type name> identifies a user-defined type.

25) An <attribute name> identifies an attribute of a structured type.

26) A <savepoint name> identifies a savepoint. The scope of a <savepoint name> is the SQL-transaction in which it was defined.

27) A <sequence generator name> identifies a sequence generator.

28) A <field name> identifies a field.

29) A <role name> identifies a role.

30) A <user identifier> identifies a user.

31) The value *ESN* of an <extended statement name> identifies a statement prepared by the execution of a <prepare statement>. If a <scope option> of GLOBAL is specified, then *ESN* is a global extended name; otherwise, it is a local extended name.

NOTE 78 — The scope of an extended name is defined in Subclause 4.24.2, "Dynamic SQL statements and descriptor areas".

32) A <dynamic cursor name> is a non-extended name that identifies a cursor in an <SQL dynamic statement>.

NOTE 79 — The scope of a non-extended name is defined in Subclause 4.24.2, "Dynamic SQL statements and descriptor areas".

33) A <statement name> is a non-extended name that identifies a prepared statement created by the execution of a <prepare statement>.

34) The value *ECN* of an <extended cursor name> identifies a cursor created by the execution of an <allocate cursor statement>. If a <scope option> of GLOBAL is specified, then *ECN* is a global extended name; otherwise, it is a local extended name.

35) The value *DN* of a <descriptor name> identifies an SQL descriptor area created by the execution of an <allocate descriptor statement>. If a <scope option> of GLOBAL is specified, then *DN* is a global extended name; otherwise, it is a local extended name.

36) A <window name> identifies a window.

Conformance Rules

1) Without Feature T271, "Savepoints", conforming SQL language shall not contain a <savepoint name>.

2) Without Feature T331, "Basic roles", conforming SQL language shall not contain a <role name>.

3) Without Feature T121, "WITH (excluding RECURSIVE) in query expression", conforming SQL language shall not contain a <query name>.

4) Without Feature S023, "Basic structured types", conforming SQL language shall not contain a <attribute name>.

5) Without Feature T051, "Row types", conforming SQL language shall not contain a <field name>.

6) Without Feature F651, "Catalog name qualifiers", conforming SQL language shall not contain a <catalog name>.

7) Without Feature F771, "Connection management", conforming SQL language shall not contain an explicit <connection name>.

8) Without Feature F690, "Collation support", conforming SQL language shall not contain a <collation name>.

9) Without Feature F695, "Translation support", conforming SQL language shall not contain a <transliteration name>.

10) Without Feature F695, "Translation support", conforming SQL language shall not contain a <transcoding name>.

11) Without Feature F821, "Local table references", conforming SQL language shall not contain a <local or schema qualifier> that contains a <local qualifier>.

12) Without Feature F251, "Domain support", conforming SQL language shall not contain a <domain name>.

13) Without Feature F491, "Constraint management", conforming SQL language shall not contain a <constraint name>.

14) Without Feature F461, “Named character sets”, conforming SQL language shall not contain a <character set name>.

15) Without Feature T601, “Local cursor references”, a <cursor name> shall not contain a <local qualifier>.

16) Without Feature B032, “Extended dynamic SQL”, conforming SQL language shall not contain a <extended statement name> or <extended cursor name>.

17) Without Feature B031, “Basic dynamic SQL”, conforming SQL language shall not contain an <SQL statement name>.

18) Without Feature B031, “Basic dynamic SQL”, conforming SQL language shall not contain <dynamic cursor name>.

19) Without Feature B031, “Basic dynamic SQL”, conforming SQL language shall not contain a <descriptor name>.

20) Without Feature T612, “Advanced OLAP operations”, conforming SQL language shall not contain a <window name>.

21) Without Feature T176, “Sequence generator support”, conforming SQL language shall not contain a <sequence generator name>.

22) Without Feature B032, “Extended dynamic SQL”, conforming SQL language shall not contain a <descriptor name> that is not a <literal>.

