

This file contains a depth-first tree traversal of the BNF for the language done at about 27-AUG-1992 11:03:41.64. The specific version of the BNF included here is: ANSI-only, SQL2-only.

```

<SQL_terminal_character> ::=
  <SQL_language_character>
  | <SQL_embedded_language_character>

<SQL_language_character> ::=
  <simple_Latin_letter>
  | <digit>
  | <SQL_special_character>

<simple_Latin_letter> ::=
  <simple_Latin_upper_case_letter>
  | <simple_Latin_lower_case_letter>

<simple_Latin_upper_case_letter> ::=
  A | B | C | D | E | F | G | H | I | J | K | L | M | N | O
  | P | Q | R | S | T | U | V | W | X | Y | Z

<simple_Latin_lower_case_letter> ::=
  a | b | c | d | e | f | g | h | i | j | k | l | m | n | o
  | p | q | r | s | t | u | v | w | x | y | z

<digit> ::=
  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<SQL_special_character> ::=
  <space>
  | <double_quote>
  | <percent>
  | <ampersand>
  | <quote>
  | <left_paren>
  | <right_paren>
  | <asterisk>
  | <plus_sign>
  | <comma>
  | <minus_sign>
  | <period>
  | <solidus>
  | <colon>
  | <semicolon>
  | <less_than_operator>
  | <equals_operator>
  | <greater_than_operator>
  | <question_mark>
  | <underscore>
  | <vertical_bar>

<space> ::= !! <EMPHASIS>(space character in character set in use)

<double_quote> ::= "

<percent> ::= %

<ampersand> ::= &

<quote> ::= '

```

```
<left_paren> ::= (  
<right_paren> ::= )  
<asterisk> ::= *  
<plus_sign> ::= +  
<comma> ::= ,  
<minus_sign> ::= -  
<period> ::= .  
<solidus> ::= /  
<colon> ::= :  
<semicolon> ::= ;  
<less_than_operator> ::= <  
<equals_operator> ::= =  
<greater_than_operator> ::= >  
<question_mark> ::= ?  
<underscore> ::= _  
<vertical_bar> ::= |  
<SQL_embedded_language_character> ::=  
    <left_bracket>  
    | <right_bracket>  
<left_bracket> ::= [  
<right_bracket> ::= ]  
<token> ::=  
    <nondelimiter_token>  
    | <delimiter_token>  
<nondelimiter_token> ::=  
    <regular_identifier>  
    | <key_word>  
    | <unsigned_numeric_literal>  
    | <national_character_string_literal>  
    | <bit_string_literal>  
    | <hex_string_literal>  
<regular_identifier> ::= <identifier_body>  
<identifier_body> ::=  
    <identifier_start> [ { <underscore> | <identifier_part> }... ]  
<identifier_start> ::= <EMPHASIS>(!! See the Syntax Rules)
```

```

<identifier_part> ::=
    <identifier_start>
    | <digit>

<key_word> ::=
    <reserved_word>
    | <non-reserved_word>

<reserved_word> ::=
    ABSOLUTE | ACTION | ADD | ALL
    | ALLOCATE | ALTER | AND
    | ANY | ARE
    | AS | ASC
    | ASSERTION | AT
    | AUTHORIZATION | AVG
    | BEGIN | BETWEEN | BIT | BIT_LENGTH
    | BOTH | BY
    | CASCADE | CASCADED | CASE | CAST
    | CATALOG
    | CHAR | CHARACTER | CHAR_LENGTH
    | CHARACTER_LENGTH | CHECK | CLOSE | COALESCE
    | COLLATE | COLLATION
    | COLUMN | COMMIT
    | CONNECT
    | CONNECTION | CONSTRAINT
    | CONSTRAINTS | CONTINUE
    | CONVERT | CORRESPONDING | COUNT | CREATE | CROSS
    | CURRENT
    | CURRENT_DATE | CURRENT_TIME
    | CURRENT_TIMESTAMP | CURRENT_USER | CURSOR
    | DATE | DAY | DEALLOCATE | DEC
    | DECIMAL | DECLARE | DEFAULT | DEFERRABLE
    | DEFERRED | DELETE | DESC | DESCRIBE | DESCRIPTOR
    | DIAGNOSTICS
    | DISCONNECT | DISTINCT | DOMAIN | DOUBLE | DROP
    | ELSE | END | END-EXEC | ESCAPE
    | EXCEPT | EXCEPTION
    | EXEC | EXECUTE | EXISTS
    | EXTERNAL | EXTRACT
    | FALSE | FETCH | FIRST | FLOAT | FOR
    | FOREIGN | FOUND | FROM | FULL
    | GET | GLOBAL | GO | GOTO
    | GRANT | GROUP
    | HAVING | HOUR
    | IDENTITY | IMMEDIATE | IN | INDICATOR
    | INITIALLY | INNER | INPUT
    | INSENSITIVE | INSERT | INT | INTEGER | INTERSECT
    | INTERVAL | INTO | IS
    | ISOLATION
    | JOIN
    | KEY
    | LANGUAGE | LAST | LEADING | LEFT
    | LEVEL | LIKE | LOCAL | LOWER
    | MATCH | MAX | MIN | MINUTE | MODULE
    | MONTH
    | NAMES | NATIONAL | NATURAL | NCHAR | NEXT | NO
    | NOT | NULL
    | NULLIF | NUMERIC
    | OCTET_LENGTH | OF
    | ON | ONLY | OPEN | OPTION | OR

```

ORDER | OUTER
OUTPUT | OVERLAPS
PAD | PARTIAL | POSITION | PRECISION | PREPARE
PRESERVE | PRIMARY
PRIOR | PRIVILEGES | PROCEDURE | PUBLIC
READ | REAL | REFERENCES | RELATIVE | RESTRICT
REVOKE | RIGHT
ROLLBACK | ROWS
SCHEMA | SCROLL | SECOND | SECTION
SELECT
SESSION | SESSION_USER | SET
SIZE | SMALLINT | SOME | SPACE | SQL | SQLCODE
SQLERROR | SQLSTATE
SUBSTRING | SUM | SYSTEM_USER
TABLE | TEMPORARY
THEN | TIME | TIMESTAMP
TIMEZONE_HOUR | TIMEZONE_MINUTE
TO | TRAILING | TRANSACTION
TRANSLATE | TRANSLATION | TRIM | TRUE
UNION | UNIQUE | UNKNOWN | UPDATE | UPPER | USAGE
USER | USING
VALUE | VALUES | VARCHAR | VARYING | VIEW
WHEN | WHENEVER | WHERE | WITH | WORK | WRITE
YEAR
ZONE

<non-reserved_word> ::=

ADA
C | CATALOG_NAME
CHARACTER_SET_CATALOG | CHARACTER_SET_NAME
CHARACTER_SET_SCHEMA | CLASS_ORIGIN | COBOL | COLLATION_CATALOG
COLLATION_NAME | COLLATION_SCHEMA | COLUMN_NAME | COMMAND_FUNCTION
COMMITTED
CONDITION_NUMBER | CONNECTION_NAME | CONSTRAINT_CATALOG | CONSTRAINT_NAME
CONSTRAINT_SCHEMA | CURSOR_NAME
DATA | DATETIME_INTERVAL_CODE
DATETIME_INTERVAL_PRECISION | DYNAMIC_FUNCTION
FORTRAN
LENGTH
MESSAGE_LENGTH | MESSAGE_OCTET_LENGTH | MESSAGE_TEXT | MORE | MUMPS
NAME | NULLABLE | NUMBER
PASCAL | PLI
REPEATABLE | RETURNED_LENGTH | RETURNED_OCTET_LENGTH | RETURNED_SQLSTATE
ROW_COUNT
SCALE | SCHEMA_NAME | SERIALIZABLE | SERVER_NAME | SUBCLASS_ORIGIN
TABLE_NAME | TYPE
UNCOMMITTED | UNNAMED

<unsigned_numeric_literal> ::=
 <exact_numeric_literal>
 | <approximate_numeric_literal>

<exact_numeric_literal> ::=
 <unsigned_integer> [<period> [<unsigned_integer>]]
 | <period> <unsigned_integer>

<unsigned_integer> ::= <digit>...

<approximate_numeric_literal> ::= <mantissa> E <exponent>

```
<mantissa> ::= <exact_numeric_literal>
<exponent> ::= <signed_integer>
<signed_integer> ::= [ <sign> ] <unsigned_integer>
<sign> ::= <plus_sign> | <minus_sign>
<national_character_string_literal> ::=
  N <quote> [ <character_representation>... ] <quote>
  [ { <separator>... <quote> [ <character_representation>... ] <quote> }... ]
<character_representation> ::=
  <nonquote_character>
  | <quote_symbol>
<nonquote_character> ::= !! <EMPHASIS>(See the Syntax Rules.)
<quote_symbol> ::= <quote>
<separator> ::= { <comment> | <space> | <newline> }...
<comment> ::=
  <comment_introducer> [ <comment_character>... ] <newline>
<comment_introducer> ::= <minus_sign>[...]
<comment_character> ::=
  <nonquote_character>
  | <quote>
<newline> ::= !! <EMPHASIS>(implementation-defined end-of-line indicator)
<bit_string_literal> ::=
  B <quote> [ <bit>... ] <quote>
  [ { <separator>... <quote> [ <bit>... ] <quote> }... ]
<bit> ::= 0 | 1
<hex_string_literal> ::=
  X <quote> [ <hexit>... ] <quote>
  [ { <separator>... <quote> [ <hexit>... ] <quote> }... ]
<hexit> ::= <digit> | A | B | C | D | E | F | a | b | c | d | e | f
<delimiter_token> ::=
  <character_string_literal>
  | <date_string>
  | <time_string>
  | <timestamp_string>
  | <interval_string>
  | <delimited_identifier>
  | <SQL_special_character>
  | <not_equals_operator>
  | <greater_than_or_equals_operator>
  | <less_than_or_equals_operator>
  | <concatenation_operator>
  | <double_period>
  | <left_bracket>
```

```
| <right_bracket>
<character_string_literal> ::=
  [ <introducer> ]
  <quote> [ <character_representation>... ] <quote>
  [ { <separator>... <quote> [ <character_representation>... ] <quote> }... ]
<introducer> ::= <underscore>
<character_set_specification> ::=
  <standard_character_repertoire_name>
  | <implementation-defined_character_repertoire_name>
  | <user-defined_character_repertoire_name>
  | <standard_universal_character_form-of-use_name>
  | <implementation-defined_universal_character_form-of-use_name>
<standard_character_repertoire_name> ::= <character_set_name>
<character_set_name> ::= [ <schema_name> <period> ]
  <SQL_language_identifier>
<schema_name> ::=
  [ <catalog_name> <period> ] <unqualified_schema_name>
<catalog_name> ::= <identifier>
<identifier> ::=
  [ <introducer> ] <actual_identifier>
<actual_identifier> ::=
  <regular_identifier>
  | <delimited_identifier>
<delimited_identifier> ::=
  <double_quote> <delimited_identifier_body> <double_quote>
<delimited_identifier_body> ::= <delimited_identifier_part>...
<delimited_identifier_part> ::=
  <nondoublequote_character>
  | <doublequote_symbol>
<nondoublequote_character> ::= <EMPHASIS>(!! See the Syntax Rules)
<doublequote_symbol> ::= <double_quote>
<unqualified_schema_name> ::= <identifier>
<SQL_language_identifier> ::=
  <SQL_language_identifier_start>
  [ { <underscore> | <SQL_language_identifier_part> }... ]
<SQL_language_identifier_start> ::= <simple_Latin_letter>
<SQL_language_identifier_part> ::=
  <simple_Latin_letter>
  | <digit>
<implementation-defined_character_repertoire_name> ::=
  <character_set_name>
```

```

<user-defined_character_repertoire_name> ::= <character_set_name>

<standard_universal_character_form-of-use_name> ::=
  <character_set_name>

<implementation-defined_universal_character_form-of-use_name> ::=
  <character_set_name>

<date_string> ::=
  <quote> <date_value> <quote>

<date_value> ::=
  <years_value> <minus_sign> <months_value>
  <minus_sign> <days_value>

<years_value> ::= <datetime_value>

<datetime_value> ::= <unsigned_integer>

<months_value> ::= <datetime_value>

<days_value> ::= <datetime_value>

<time_string> ::=
  <quote> <time_value> [ <time_zone_interval> ] <quote>

<time_value> ::=
  <hours_value> <colon> <minutes_value> <colon> <seconds_value>

<hours_value> ::= <datetime_value>

<minutes_value> ::= <datetime_value>

<seconds_value> ::=
  <seconds_integer_value> [ <period> [ <seconds_fraction> ] ]

<seconds_integer_value> ::= <unsigned_integer>

<seconds_fraction> ::= <unsigned_integer>

<time_zone_interval> ::=
  <sign> <hours_value> <colon> <minutes_value>

<timestamp_string> ::=
  <quote> <date_value> <space> <time_value>
  [ <time_zone_interval> ] <quote>

<interval_string> ::=
  <quote> { <year-month_literal> | <day-time_literal> } <quote>

<year-month_literal> ::=
  <years_value>
  | [ <years_value> <minus_sign> ] <months_value>

<day-time_literal> ::=
  <day-time_interval>
  | <time_interval>

<day-time_interval> ::=

```

```
<days_value>
  [ <space> <hours_value> [ <colon> <minutes_value>
    [ <colon> <seconds_value> ] ] ]

<time_interval> ::=
  <hours_value> [ <colon> <minutes_value> [ <colon> <seconds_value> ] ]
  | <minutes_value> [ <colon> <seconds_value> ]
  | <seconds_value>

<not_equals_operator> ::= <>

<greater_than_or_equals_operator> ::= >=

<less_than_or_equals_operator> ::= <=

<concatenation_operator> ::= ||

<double_period> ::= ..

<module> ::=
  <module_name_clause>
  <language_clause>
  <module_authorization_clause>
  [ <temporary_table_declaration>... ]
  <module_contents>...

<module_name_clause> ::=
  MODULE [ <module_name> ]
  [ <module_character_set_specification> ]

<module_name> ::= <identifier>

<module_character_set_specification> ::=
  NAMES ARE <character_set_specification>

<language_clause> ::=
  LANGUAGE <language_name>

<language_name> ::=
  ADA | C | COBOL | FORTRAN | MUMPS | PASCAL | PLI

<module_authorization_clause> ::=
  SCHEMA <schema_name>
  | AUTHORIZATION <module_authorization_identifier>
  | SCHEMA <schema_name>
  AUTHORIZATION <module_authorization_identifier>

<module_authorization_identifier> ::=
  <authorization_identifier>

<authorization_identifier> ::= <identifier>

<temporary_table_declaration> ::=
  DECLARE LOCAL TEMPORARY TABLE
  <qualified_local_table_name>
  <table_element_list>
  [ ON COMMIT { PRESERVE | DELETE } ROWS ]

<qualified_local_table_name> ::=
  MODULE <period> <local_table_name>
```

```

<local_table_name> ::= <qualified_identifier>

<qualified_identifier> ::= <identifier>

<table_element_list> ::=
    <left_paren> <table_element> [ { <comma> <table_element> }... ] <right_paren>

<table_element> ::=
    <column_definition>
  | <table_constraint_definition>

<column_definition> ::=
    <column_name> { <data_type> | <domain_name> }
  [ <default_clause> ]
  [ <column_constraint_definition>... ]
  [ <collate_clause> ]

<column_name> ::= <identifier>

<data_type> ::=
    <character_string_type>
      [ CHARACTER SET <character_set_specification> ]
  | <national_character_string_type>
  | <bit_string_type>
  | <numeric_type>
  | <datetime_type>
  | <interval_type>

<character_string_type> ::=
    CHARACTER [ <left_paren> <length> <right_paren> ]
  | CHAR [ <left_paren> <length> <right_paren> ]
  | CHARACTER VARYING <left_paren> <length> <right_paren>
  | CHAR VARYING <left_paren> <length> <right_paren>
  | VARCHAR <left_paren> <length> <right_paren>

<length> ::= <unsigned_integer>

<national_character_string_type> ::=
    NATIONAL CHARACTER [ <left_paren> <length> <right_paren> ]
  | NATIONAL CHAR [ <left_paren> <length> <right_paren> ]
  | NCHAR [ <left_paren> <length> <right_paren> ]
  | NATIONAL CHARACTER VARYING <left_paren> <length> <right_paren>
  | NATIONAL CHAR VARYING <left_paren> <length> <right_paren>
  | NCHAR VARYING <left_paren> <length> <right_paren>

<bit_string_type> ::=
    BIT [ <left_paren> <length> <right_paren> ]
  | BIT VARYING <left_paren> <length> <right_paren>

<numeric_type> ::=
    <exact_numeric_type>
  | <approximate_numeric_type>

<exact_numeric_type> ::=
    NUMERIC [ <left_paren> <precision> [ <comma> <scale> ] <right_paren> ]
  | DECIMAL [ <left_paren> <precision> [ <comma> <scale> ] <right_paren> ]
  | DEC [ <left_paren> <precision> [ <comma> <scale> ] <right_paren> ]
  | INTEGER
  | INT

```

```

| SMALLINT

<precision> ::= <unsigned_integer>

<scale> ::= <unsigned_integer>

<approximate_numeric_type> ::=
    FLOAT [ <left_paren> <precision> <right_paren> ]
| REAL
| DOUBLE PRECISION

<datetime_type> ::=
    DATE
| TIME [ <left_paren> <time_precision> <right_paren> ]
    [ WITH TIME ZONE ]
| TIMESTAMP [ <left_paren> <timestamp_precision> <right_paren> ]
    [ WITH TIME ZONE ]

<time_precision> ::= <time_fractional_seconds_precision>

<time_fractional_seconds_precision> ::= <unsigned_integer>

<timestamp_precision> ::= <time_fractional_seconds_precision>

<interval_type> ::= INTERVAL <interval_qualifier>

<interval_qualifier> ::=
    <start_field> TO <end_field>
| <single_datetime_field>

<start_field> ::=
    <non-second_datetime_field>
    [ <left_paren> <interval_leading_field_precision> <right_paren> ]

<non-second_datetime_field> ::= YEAR | MONTH | DAY | HOUR
| MINUTE

<interval_leading_field_precision> ::= <unsigned_integer>

<end_field> ::=
    <non-second_datetime_field>
| SECOND [ <left_paren> <interval_fractional_seconds_precision> <right_paren> ]

<interval_fractional_seconds_precision> ::= <unsigned_integer>

<single_datetime_field> ::=
    <non-second_datetime_field>
    [ <left_paren> <interval_leading_field_precision> <right_paren> ]
| SECOND [ <left_paren> <interval_leading_field_precision>
    [ <comma> <interval_fractional_seconds_precision> ] <right_paren> ]

<domain_name> ::= <qualified_name>

<qualified_name> ::=
    [ <schema_name> <period> ] <qualified_identifier>

<default_clause> ::=
    DEFAULT <default_option>

<default_option> ::=

```

```

    <literal>
    | <datetime_value_function>
    | USER
    | CURRENT_USER
    | SESSION_USER
    | SYSTEM_USER
    | NULL

<literal> ::=
    <signed_numeric_literal>
    | <general_literal>

<signed_numeric_literal> ::=
    [ <sign> ] <unsigned_numeric_literal>

<general_literal> ::=
    <character_string_literal>
    | <national_character_string_literal>
    | <bit_string_literal>
    | <hex_string_literal>
    | <datetime_literal>
    | <interval_literal>

<datetime_literal> ::=
    <date_literal>
    | <time_literal>
    | <timestamp_literal>

<date_literal> ::=
    DATE <date_string>

<time_literal> ::=
    TIME <time_string>

<timestamp_literal> ::=
    TIMESTAMP <timestamp_string>

<interval_literal> ::=
    INTERVAL [ <sign> ] <interval_string> <interval_qualifier>

<datetime_value_function> ::=
    <current_date_value_function>
    | <current_time_value_function>
    | <current_timestamp_value_function>

<current_date_value_function> ::= CURRENT_DATE

<current_time_value_function> ::=
    CURRENT_TIME [ <left_paren> <time_precision> <right_paren> ]

<current_timestamp_value_function> ::=
    CURRENT_TIMESTAMP [ <left_paren> <timestamp_precision> <right_paren> ]

<column_constraint_definition> ::=
    [ <constraint_name_definition> ]
    <column_constraint>
    [ <constraint_attributes> ]

<constraint_name_definition> ::= CONSTRAINT <constraint_name>

```

```
<constraint_name> ::= <qualified_name>

<column_constraint> ::=
    NOT NULL
  | <unique_specification>
  | <references_specification>
  | <check_constraint_definition>

<unique_specification> ::=
    UNIQUE | PRIMARY KEY

<references_specification> ::=
    REFERENCES <referenced_table_and_columns>
    [ MATCH <match_type> ]
    [ <referential_triggered_action> ]

<referenced_table_and_columns> ::=
    <table_name> [ <left_paren> <reference_column_list> <right_paren> ]

<table_name> ::=
    <qualified_name>
  | <qualified_local_table_name>

<reference_column_list> ::= <column_name_list>

<column_name_list> ::=
    <column_name> [ { <comma> <column_name> }... ]

<match_type> ::=
    FULL
  | PARTIAL

<referential_triggered_action> ::=
    <update_rule> [ <delete_rule> ]
  | <delete_rule> [ <update_rule> ]

<update_rule> ::= ON UPDATE <referential_action>

<referential_action> ::=
    CASCADE
  | SET NULL
  | SET DEFAULT
  | NO ACTION

<delete_rule> ::= ON DELETE <referential_action>

<check_constraint_definition> ::=
    CHECK
    <left_paren> <search_condition> <right_paren>

<search_condition> ::=
    <boolean_term>
  | <search_condition> OR <boolean_term>

<boolean_term> ::=
    <boolean_factor>
  | <boolean_term> AND <boolean_factor>

<boolean_factor> ::=
    [ NOT ] <boolean_test>
```

```

<boolean_test> ::=
    <boolean_primary> [ IS [ NOT ]
        <truth_value> ]

<boolean_primary> ::=
    <predicate>
    | <left_paren> <search_condition> <right_paren>

<predicate> ::=
    <comparison_predicate>
    | <between_predicate>
    | <in_predicate>
    | <like_predicate>
    | <null_predicate>
    | <quantified_comparison_predicate>
    | <exists_predicate>
    | <unique_predicate>
    | <match_predicate>
    | <overlaps_predicate>

<comparison_predicate> ::=
    <row_value_constructor> <comp_op>
        <row_value_constructor>

<row_value_constructor> ::=
    <row_value_constructor_element>
    | <left_paren> <row_value_constructor_list> <right_paren>
    | <row_subquery>

<row_value_constructor_element> ::=
    <value_expression>
    | <>null_specification>
    | <default_specification>

<value_expression> ::=
    <numeric_value_expression>
    | <string_value_expression>
    | <datetime_value_expression>
    | <interval_value_expression>

<numeric_value_expression> ::=
    <term>
    | <numeric_value_expression> <plus_sign> <term>
    | <numeric_value_expression> <minus_sign> <term>

<term> ::=
    <factor>
    | <term> <asterisk> <factor>
    | <term> <solidus> <factor>

<factor> ::=
    [ <sign> ] <numeric_primary>

<numeric_primary> ::=
    <value_expression_primary>
    | <numeric_value_function>

<value_expression_primary> ::=
    <unsigned_value_specification>

```

```

| <column_reference>
| <set_function_specification>
| <scalar_subquery>
| <case_expression>
| <left_paren> <value_expression> <right_paren>
| <cast_specification>

<unsigned_value_specification> ::=
  <unsigned_literal>
  | <general_value_specification>

<unsigned_literal> ::=
  <unsigned_numeric_literal>
  | <general_literal>

<general_value_specification> ::=
  <parameter_specification>
  | <dynamic_parameter_specification>
  | <variable_specification>
  | USER
  | CURRENT_USER
  | SESSION_USER
  | SYSTEM_USER
  | VALUE

<parameter_specification> ::=
  <parameter_name> [ <indicator_parameter> ]

<parameter_name> ::= <colon> <identifier>

<indicator_parameter> ::=
  [ INDICATOR ] <parameter_name>

<dynamic_parameter_specification> ::= <question_mark>

<variable_specification> ::=
  <embedded_variable_name> [ <indicator_variable> ]

<embedded_variable_name> ::=
  <colon>

<host_identifier> ::=
  <Ada_host_identifier>
  | <C_host_identifier>
  | <COBOL_host_identifier>
  | <Fortran_host_identifier>
  | <MUMPS_host_identifier>
  | <Pascal_host_identifier>
  | <PL/I_host_identifier>

<Ada_host_identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<C_host_identifier> ::=
  !! <EMPHASIS>(See the Syntax Rules.)

<COBOL_host_identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<Fortran_host_identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<MUMPS_host_identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

```

```

<Pascal_host_identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<PL/I_host_identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<indicator_variable> ::=
    [ INDICATOR ] <embedded_variable_name>

<column_reference> ::= [ <qualifier> <period> ] <column_name>

<qualifier> ::=
    <table_name>
    | <correlation_name>

<correlation_name> ::= <identifier>

<set_function_specification> ::=
    COUNT <left_paren> <asterisk> <right_paren>
    | <general_set_function>

<general_set_function> ::=
    <set_function_type>
    <left_paren> [ <set_quantifier> ] <value_expression> <right_paren>

<set_function_type> ::=
    AVG | MAX | MIN | SUM | COUNT

<set_quantifier> ::= DISTINCT | ALL

<scalar_subquery> ::= <subquery>

<subquery> ::= <left_paren> <query_expression> <right_paren>

<query_expression> ::=
    <non-join_query_expression>
    | <joined_table>

<non-join_query_expression> ::=
    <non-join_query_term>
    | <query_expression> UNION [ ALL ]
      [ <corresponding_spec> ] <query_term>
    | <query_expression> EXCEPT [ ALL ]
      [ <corresponding_spec> ] <query_term>

<non-join_query_term> ::=
    <non-join_query_primary>
    | <query_term> INTERSECT [ ALL ]
      [ <corresponding_spec> ] <query_primary>

<non-join_query_primary> ::=
    <simple_table>
    | <left_paren> <non-join_query_expression> <right_paren>

<simple_table> ::=
    <query_specification>
    | <table_value_constructor>
    | <explicit_table>

<query_specification> ::=
    SELECT [ <set_quantifier> ] <select_list> <table_expression>

```

```

<select_list> ::=
    <asterisk>
    | <select_sublist> [ { <comma> <select_sublist> }... ]

<select_sublist> ::=
    <derived_column>
    | <qualifier> <period> <asterisk>

<derived_column> ::= <value_expression> [ <as_clause> ]

<as_clause> ::= [ AS ] <column_name>

<table_expression> ::=
    <from_clause>
    [ <where_clause> ]
    [ <group_by_clause> ]
    [ <having_clause> ]

<from_clause> ::= FROM <table_reference>
    [ { <comma> <table_reference> }... ]

<table_reference> ::=
    <table_name> [ [ AS ] <correlation_name>
        [ <left_paren> <derived_column_list> <right_paren> ] ]
    | <derived_table> [ AS ] <correlation_name>
        [ <left_paren> <derived_column_list> <right_paren> ]
    | <joined_table>

<derived_column_list> ::= <column_name_list>

<derived_table> ::= <table_subquery>

<table_subquery> ::= <subquery>

<joined_table> ::=
    <cross_join>
    | <qualified_join>
    | <left_paren> <joined_table> <right_paren>

<cross_join> ::=
    <table_reference> CROSS JOIN <table_reference>

<qualified_join> ::=
    <table_reference> [ NATURAL ] [ <join_type> ] JOIN
    <table_reference> [ <join_specification> ]

<join_type> ::=
    INNER
    | <outer_join_type> [ OUTER ]
    | UNION

<outer_join_type> ::=
    LEFT
    | RIGHT
    | FULL

<join_specification> ::=
    <join_condition>
    | <named_columns_join>

```

```

<join_condition> ::= ON <search_condition>

<named_columns_join> ::=
    USING <left_paren> <join_column_list> <right_paren>

<join_column_list> ::= <column_name_list>

<where_clause> ::= WHERE <search_condition>

<group_by_clause> ::=
    GROUP BY <grouping_column_reference_list>

<grouping_column_reference_list> ::=
    <grouping_column_reference>
    [ { <comma> <grouping_column_reference> }... ]

<grouping_column_reference> ::=
    <column_reference> [ <collate_clause> ]

<collate_clause> ::= COLLATE <collation_name>

<collation_name> ::= <qualified_name>

<having_clause> ::= HAVING <search_condition>

<table_value_constructor> ::=
    VALUES <table_value_constructor_list>

<table_value_constructor_list> ::=
    <row_value_constructor> [ { <comma> <row_value_constructor> }... ]

<explicit_table> ::= TABLE <table_name>

<query_term> ::=
    <non-join_query_term>
    | <joined_table>

<corresponding_spec> ::=
    CORRESPONDING [ BY <left_paren> <corresponding_column_list> <right_paren> ]

<corresponding_column_list> ::= <column_name_list>

<query_primary> ::=
    <non-join_query_primary>
    | <joined_table>

<case_expression> ::=
    <case_abbreviation>
    | <case_specification>

<case_abbreviation> ::=
    NULLIF <left_paren> <value_expression> <comma>
    <value_expression> <right_paren>
    | COALESCE <left_paren> <value_expression>
    { <comma> <value_expression> }... <right_paren>

<case_specification> ::=
    <simple_case>
    | <searched_case>

```

```

<simple_case> ::=
    CASE <case_operand>
        <simple_when_clause>...
        [ <else_clause> ]
    END

<case_operand> ::= <value_expression>

<simple_when_clause> ::= WHEN <when_operand> THEN <result>

<when_operand> ::= <value_expression>

<result> ::= <result_expression> | NULL

<result_expression> ::= <value_expression>

<else_clause> ::= ELSE <result>

<searched_case> ::=
    CASE
        <searched_when_clause>...
        [ <else_clause> ]
    END

<searched_when_clause> ::= WHEN <search_condition> THEN <result>

<cast_specification> ::=
    CAST <left_paren> <cast_operand> AS
        <cast_target> <right_paren>

<cast_operand> ::=
    <value_expression>
    | NULL

<cast_target> ::=
    <domain_name>
    | <data_type>

<numeric_value_function> ::=
    <position_expression>
    | <extract_expression>
    | <length_expression>

<position_expression> ::=
    POSITION <left_paren> <character_value_expression>
        IN <character_value_expression> <right_paren>

<character_value_expression> ::=
    <concatenation>
    | <character_factor>

<concatenation> ::=
    <character_value_expression> <concatenation_operator>
    <character_factor>

<character_factor> ::=
    <character_primary> [ <collate_clause> ]

<character_primary> ::=

```

```

    <value_expression_primary>
  | <string_value_function>

<string_value_function> ::=
    <character_value_function>
  | <bit_value_function>

<character_value_function> ::=
    <character_substring_function>
  | <fold>
  | <form-of-use_conversion>
  | <character_translation>
  | <trim_function>

<character_substring_function> ::=
    SUBSTRING <left_paren> <character_value_expression> FROM <start_position>
              [ FOR <string_length> ] <right_paren>

<start_position> ::= <numeric_value_expression>

<string_length> ::= <numeric_value_expression>

<fold> ::= { UPPER | LOWER }
          <left_paren> <character_value_expression> <right_paren>

<form-of-use_conversion> ::=
    CONVERT <left_paren> <character_value_expression>
           USING <form-of-use_conversion_name> <right_paren>

<form-of-use_conversion_name> ::= <qualified_name>

<character_translation> ::=
    TRANSLATE <left_paren> <character_value_expression>
             USING <translation_name> <right_paren>

<translation_name> ::= <qualified_name>

<trim_function> ::=
    TRIM <left_paren> <trim_operands> <right_paren>

<trim_operands> ::=
    [ [ <trim_specification> ] [ <trim_character> ] FROM ] <trim_source>

<trim_specification> ::=
    LEADING
  | TRAILING
  | BOTH

<trim_character> ::= <character_value_expression>

<trim_source> ::= <character_value_expression>

<bit_value_function> ::=
    <bit_substring_function>

<bit_substring_function> ::=
    SUBSTRING <left_paren> <bit_value_expression> FROM <start_position>
              [ FOR <string_length> ] <right_paren>

<bit_value_expression> ::=
```

```

    <bit_concatenation>
    | <bit_factor>

<bit_concatenation> ::=
    <bit_value_expression> <concatenation_operator> <bit_factor>

<bit_factor> ::= <bit_primary>

<bit_primary> ::=
    <value_expression_primary>
    | <string_value_function>

<extract_expression> ::=
    EXTRACT <left_paren> <extract_field>
    FROM <extract_source> <right_paren>

<extract_field> ::=
    <datetime_field>
    | <time_zone_field>

<datetime_field> ::=
    <non-second_datetime_field>
    | SECOND

<time_zone_field> ::=
    TIMEZONE_HOUR
    | TIMEZONE_MINUTE

<extract_source> ::=
    <datetime_value_expression>
    | <interval_value_expression>

<datetime_value_expression> ::=
    <datetime_term>
    | <interval_value_expression> <plus_sign> <datetime_term>
    | <datetime_value_expression> <plus_sign> <interval_term>
    | <datetime_value_expression> <minus_sign> <interval_term>

<interval_term> ::=
    <interval_factor>
    | <interval_term_2> <asterisk> <factor>
    | <interval_term_2> <solidus> <factor>
    | <term> <asterisk> <interval_factor>

<interval_factor> ::=
    [ <sign> ] <interval_primary>

<interval_primary> ::=
    <value_expression_primary> [ <interval_qualifier> ]

<interval_term_2> ::= <interval_term>

<interval_value_expression> ::=
    <interval_term>
    | <interval_value_expression_1> <plus_sign> <interval_term_1>
    | <interval_value_expression_1> <minus_sign> <interval_term_1>
    | <left_paren> <datetime_value_expression> <minus_sign>
    <datetime_term> <right_paren> <interval_qualifier>

<interval_value_expression_1> ::= <interval_value_expression>

```

```

<interval_term_1> ::= <interval_term>

<datetime_term> ::=
    <datetime_factor>

<datetime_factor> ::=
    <datetime_primary> [ <time_zone> ]

<datetime_primary> ::=
    <value_expression_primary>
  | <datetime_value_function>

<time_zone> ::=
    AT <time_zone_specifier>

<time_zone_specifier> ::=
    LOCAL
  | TIME_ZONE <interval_value_expression>

<length_expression> ::=
    <char_length_expression>
  | <octet_length_expression>
  | <bit_length_expression>

<char_length_expression> ::=
    { CHAR_LENGTH | CHARACTER_LENGTH }
    <left_paren> <string_value_expression> <right_paren>

<string_value_expression> ::=
    <character_value_expression>
  | <bit_value_expression>

<octet_length_expression> ::=
    OCTET_LENGTH <left_paren> <string_value_expression> <right_paren>

<bit_length_expression> ::=
    BIT_LENGTH <left_paren> <string_value_expression> <right_paren>

<null_specification> ::=
    NULL

<default_specification> ::=
    DEFAULT

<row_value_constructor_list> ::=
    <row_value_constructor_element>
    [ { <comma> <row_value_constructor_element> }... ]

<row_subquery> ::= <subquery>

<comp_op> ::=
    <equals_operator>
  | <not_equals_operator>
  | <less_than_operator>
  | <greater_than_operator>
  | <less_than_or_equals_operator>
  | <greater_than_or_equals_operator>

<between_predicate> ::=

```

```
<row_value_constructor> [ NOT ] BETWEEN
    <row_value_constructor> AND <row_value_constructor>

<in_predicate> ::=
    <row_value_constructor>
    [ NOT ] IN <in_predicate_value>

<in_predicate_value> ::=
    <table_subquery>
    | <left_paren> <in_value_list> <right_paren>

<in_value_list> ::=
    <value_expression> { <comma> <value_expression> }...

<like_predicate> ::=
    <match_value> [ NOT ] LIKE <pattern>
    [ ESCAPE <escape_character> ]

<match_value> ::= <character_value_expression>

<pattern> ::= <character_value_expression>

<escape_character> ::= <character_value_expression>

<null_predicate> ::= <row_value_constructor>
    IS [ NOT ] NULL

<quantified_comparison_predicate> ::=
    <row_value_constructor> <comp_op> <quantifier> <table_subquery>

<quantifier> ::= <all> | <some>

<all> ::= ALL

<some> ::= SOME | ANY

<exists_predicate> ::= EXISTS <table_subquery>

<unique_predicate> ::= UNIQUE <table_subquery>

<match_predicate> ::=
    <row_value_constructor> MATCH [ UNIQUE ]
    [ PARTIAL | FULL ] <table_subquery>

<overlaps_predicate> ::=
    <row_value_constructor_1> OVERLAPS <row_value_constructor_2>

<row_value_constructor_1> ::= <row_value_constructor>

<row_value_constructor_2> ::= <row_value_constructor>

<truth_value> ::=
    TRUE
    | FALSE
    | UNKNOWN

<constraint_attributes> ::=
    <constraint_check_time> [ [ NOT ] DEFERRABLE ]
    | [ NOT ] DEFERRABLE [ <constraint_check_time> ]
```

```

<constraint_check_time> ::=
    INITIALLY DEFERRED
    | INITIALLY IMMEDIATE

<table_constraint_definition> ::=
    [ <constraint_name_definition> ]
    <table_constraint> [ <constraint_attributes> ]

<table_constraint> ::=
    <unique_constraint_definition>
    | <referential_constraint_definition>
    | <check_constraint_definition>

<unique_constraint_definition> ::=
    <unique_specification> even in SQL3)
    <unique_specification>
    <left_paren> <unique_column_list> <right_paren>

<unique_column_list> ::= <column_name_list>

<referential_constraint_definition> ::=
    FOREIGN KEY
        <left_paren> <referencing_columns> <right_paren>
        <references_specification>

<referencing_columns> ::=
    <reference_column_list>

<module_contents> ::=
    <declare_cursor>
    | <dynamic_declare_cursor>
    | <procedure>

<declare_cursor> ::=
    DECLARE <cursor_name> [ INSENSITIVE ] [ SCROLL ] CURSOR
    FOR <cursor_specification>

<cursor_name> ::= <identifier>

<cursor_specification> ::=
    <query_expression> [ <order_by_clause> ]
    [ <updatability_clause> ]

<order_by_clause> ::=
    ORDER BY <sort_specification_list>

<sort_specification_list> ::=
    <sort_specification> [ { <comma> <sort_specification> }... ]

<sort_specification> ::=
    <sort_key> [ <collate_clause> ] [ <ordering_specification> ]

<sort_key> ::=
    <column_name>
    | <unsigned_integer>

<ordering_specification> ::= ASC | DESC

<updatability_clause> ::=
    FOR

```

```

        { READ ONLY |
          UPDATE [ OF <column_name_list> ] }

<dynamic_declare_cursor> ::=
    DECLARE <cursor_name> [ INSENSITIVE ] [ SCROLL ] CURSOR
        FOR <statement_name>

<statement_name> ::= <identifier>

<procedure> ::=
    PROCEDURE <procedure_name>
        <parameter_declaration_list> <semicolon>
        <SQL_procedure_statement> <semicolon>

<procedure_name> ::= <identifier>

<parameter_declaration_list> ::=
    <left_paren> <parameter_declaration>
        [ { <comma> <parameter_declaration> }... ] <right_paren>
    | <parameter_declaration>...

<parameter_declaration> ::=
    <parameter_name> <data_type>
    | <status_parameter>

<status_parameter> ::=
    SQLCODE | SQLSTATE

<SQL_procedure_statement> ::=
    <SQL_schema_statement>
    | <SQL_data_statement>
    | <SQL_transaction_statement>
    | <SQL_connection_statement>
    | <SQL_session_statement>
    | <SQL_dynamic_statement>
    | <SQL_diagnostics_statement>

<SQL_schema_statement> ::=
    <SQL_schema_definition_statement>
    | <SQL_schema_manipulation_statement>

<SQL_schema_definition_statement> ::=
    <schema_definition>
    | <table_definition>
    | <view_definition>
    | <grant_statement>
    | <domain_definition>
    | <character_set_definition>
    | <collation_definition>
    | <translation_definition>
    | <assertion_definition>

<schema_definition> ::=
    CREATE SCHEMA <schema_name_clause>
        [ <schema_character_set_specification> ]
        [ <schema_element>... ]

<schema_name_clause> ::=
    <schema_name>
    | AUTHORIZATION <schema_authorization_identifier>

```

```
| <schema_name> AUTHORIZATION
    <schema_authorization_identifier>

<schema_authorization_identifier> ::=
    <authorization_identifier>

<schema_character_set_specification> ::=
    DEFAULT CHARACTER
        SET <character_set_specification>

<schema_element> ::=
    <domain_definition>
    | <table_definition>
    | <view_definition>
    | <grant_statement>
    | <assertion_definition>
    | <character_set_definition>
    | <collation_definition>
    | <translation_definition>

<domain_definition> ::=
    CREATE DOMAIN <domain_name>
        [ AS ] <data_type>
        [ <default_clause> ]
        [ <domain_constraint>... ]
        [ <collate_clause> ]

<domain_constraint> ::=
    [ <constraint_name_definition> ]
    <check_constraint_definition> [ <constraint_attributes> ]

<table_definition> ::=
    CREATE [ { GLOBAL | LOCAL } TEMPORARY ] TABLE
        <table_name>
        <table_element_list>
        [ ON COMMIT { DELETE | PRESERVE } ROWS ]

<view_definition> ::=
    CREATE VIEW <table_name> [ <left_paren> <view_column_list>
        <right_paren> ]
        AS <query_expression>
        [ WITH [ <levels_clause> ] CHECK OPTION ]

<view_column_list> ::= <column_name_list>

<levels_clause> ::=
    CASCADED | LOCAL

<grant_statement> ::=
    GRANT <privileges> ON <object_name>
        TO <grantee> [ { <comma> <grantee> }... ]
        [ WITH GRANT OPTION ]

<privileges> ::=
    ALL PRIVILEGES
    | <action_list>

<action_list> ::= <action> [ { <comma> <action> }... ]

<action> ::=
```

```

SELECT
| DELETE
| INSERT [ <left_paren> <privilege_column_list> <right_paren> ]
| UPDATE [ <left_paren> <privilege_column_list> <right_paren> ]
| REFERENCES [ <left_paren> <privilege_column_list> <right_paren> ]
| USAGE

```

<privilege_column_list> ::= <column_name_list>

```

<object_name> ::=
  [ TABLE ] <table_name>
| DOMAIN <domain_name>
| COLLATION <collation_name>
| CHARACTER SET <character_set_name>
| TRANSLATION <translation_name>

```

```

<grantee> ::=
  PUBLIC
| <authorization_identifier>

```

```

<assertion_definition> ::=
  CREATE ASSERTION <constraint_name> <assertion_check>
  [ <constraint_attributes> ]

```

```

<assertion_check> ::=
  CHECK
    <left_paren> <search_condition> <right_paren>

```

```

<character_set_definition> ::=
  CREATE CHARACTER SET <character_set_name>
  [ AS ]
  <character_set_source>
  [ <collate_clause> | <limited_collation_definition> ]

```

```

<character_set_source> ::=
  GET <existing_character_set_name>

```

```

<existing_character_set_name> ::=
  <standard_character_repertoire_name>
| <implementation-defined_character_repertoire_name>
| <schema_character_set_name>

```

<schema_character_set_name> ::= <character_set_name>

```

<limited_collation_definition> ::=
  COLLATION FROM <collation_source>

```

```

<collation_source> ::=
  <collating_sequence_definition>
| <translation_collation>

```

```

<collating_sequence_definition> ::=
  <external_collation>
| <schema_collation_name>
| DESC <left_paren> <collation_name> <right_paren>
| DEFAULT

```

```

<external_collation> ::=
  EXTERNAL <left_paren> <quote> <external_collation_name> <quote> <right_paren>

```

```

<external_collation_name> ::=
    <standard_collation_name>
    | <implementation-defined_collation_name>

<standard_collation_name> ::= <collation_name>

<implementation-defined_collation_name> ::= <collation_name>

<schema_collation_name> ::= <collation_name>

<translation_collation> ::=
    TRANSLATION <translation_name>
    [ THEN COLLATION <collation_name> ]

<collation_definition> ::=
    CREATE COLLATION <collation_name> FOR
    <character_set_specification>
    FROM <collation_source>
    [ <pad_attribute> ]

<pad_attribute> ::=
    NO PAD
    | PAD SPACE

<translation_definition> ::=
    CREATE TRANSLATION <translation_name>
    FOR <source_character_set_specification>
    TO <target_character_set_specification>
    FROM <translation_source>

<source_character_set_specification> ::= <character_set_specification>

<target_character_set_specification> ::= <character_set_specification>

<translation_source> ::=
    <translation_specification>

<translation_specification> ::=
    <external_translation>
    | IDENTITY
    | <schema_translation_name>

<external_translation> ::=
    EXTERNAL <left_paren> <quote> <external_translation_name> <quote> <right_paren>

<external_translation_name> ::=
    <standard_translation_name>
    | <implementation-defined_translation_name>

<standard_translation_name> ::= <translation_name>

<implementation-defined_translation_name> ::= <translation_name>

<schema_translation_name> ::= <translation_name>

<SQL_schema_manipulation_statement> ::=
    <drop_schema_statement>
    | <alter_table_statement>
    | <drop_table_statement>
    | <drop_view_statement>

```

```

| <revoke_statement>
| <alter_domain_statement>
| <drop_domain_statement>
| <drop_character_set_statement>
| <drop_collation_statement>
| <drop_translation_statement>
| <drop_assertion_statement>

<drop_schema_statement> ::=
    DROP SCHEMA <schema_name> <drop_behavior>

<drop_behavior> ::= CASCADE | RESTRICT

<alter_table_statement> ::=
    ALTER TABLE <table_name> <alter_table_action>

<alter_table_action> ::=
    <add_column_definition>
  | <alter_column_definition>
  | <drop_column_definition>
  | <add_table_constraint_definition>
  | <drop_table_constraint_definition>

<add_column_definition> ::=
    ADD [ COLUMN ] <column_definition>

<alter_column_definition> ::=
    ALTER [ COLUMN ] <column_name> <alter_column_action>

<alter_column_action> ::=
    <set_column_default_clause>
  | <drop_column_default_clause>

<set_column_default_clause> ::=
    SET <default_clause>

<drop_column_default_clause> ::=
    DROP DEFAULT

<drop_column_definition> ::=
    DROP [ COLUMN ] <column_name> <drop_behavior>

<add_table_constraint_definition> ::=
    ADD <table_constraint_definition>

<drop_table_constraint_definition> ::=
    DROP CONSTRAINT <constraint_name> <drop_behavior>

<drop_table_statement> ::=
    DROP TABLE <table_name> <drop_behavior>

<drop_view_statement> ::=
    DROP VIEW <table_name> <drop_behavior>

<revoke_statement> ::=
    REVOKE [ GRANT OPTION FOR ]
        <privileges>
    ON <object_name>
    FROM <grantee> [ { <comma> <grantee> }... ] <drop_behavior>

```

```

<alter_domain_statement> ::=
    ALTER DOMAIN <domain_name> <alter_domain_action>

<alter_domain_action> ::=
    <set_domain_default_clause>
  | <drop_domain_default_clause>
  | <add_domain_constraint_definition>
  | <drop_domain_constraint_definition>

<set_domain_default_clause> ::= SET <default_clause>

<drop_domain_default_clause> ::= DROP DEFAULT

<add_domain_constraint_definition> ::=
    ADD <domain_constraint>

<drop_domain_constraint_definition> ::=
    DROP CONSTRAINT <constraint_name>

<drop_domain_statement> ::=
    DROP DOMAIN <domain_name> <drop_behavior>

<drop_character_set_statement> ::=
    DROP CHARACTER SET <character_set_name>

<drop_collation_statement> ::=
    DROP COLLATION <collation_name>

<drop_translation_statement> ::=
    DROP TRANSLATION <translation_name>

<drop_assertion_statement> ::=
    DROP ASSERTION <constraint_name>

<SQL_data_statement> ::=
    <open_statement>
  | <fetch_statement>
  | <close_statement>
  | <select_statement: single_row>
  | <SQL_data_change_statement>

<open_statement> ::=
    OPEN <cursor_name>

<fetch_statement> ::=
    FETCH [ [ <fetch_orientation> ] FROM ]
    <cursor_name> INTO <fetch_target_list>

<fetch_orientation> ::=
    NEXT
  | PRIOR
  | FIRST
  | LAST
  | { ABSOLUTE | RELATIVE } <simple_value_specification>

<simple_value_specification> ::=
    <parameter_name>
  | <embedded_variable_name>
  | <literal>

```

```

<fetch_target_list> ::=
    <target_specification> [ { <comma> <target_specification> }... ]

<target_specification> ::=
    <parameter_specification>
  | <variable_specification>

<close_statement> ::=
    CLOSE <cursor_name>

<select_statement:_single_row> ::=
    SELECT [ <set_quantifier> ] <select_list>
    INTO <select_target_list>
    <table_expression>

<select_target_list> ::=
    <target_specification> [ { <comma> <target_specification> }... ]

<SQL_data_change_statement> ::=
    <delete_statement:_positioned>
  | <delete_statement:_searched>
  | <insert_statement>
  | <update_statement:_positioned>
  | <update_statement:_searched>

<delete_statement:_positioned> ::=
    DELETE FROM <table_name>
    WHERE CURRENT OF <cursor_name>

<delete_statement:_searched> ::=
    DELETE FROM <table_name>
    [ WHERE <search_condition> ]

<insert_statement> ::=
    INSERT INTO <table_name>
    <insert_columns_and_source>

<insert_columns_and_source> ::=
    [ <left_paren> <insert_column_list> <right_paren> ]
    <query_expression>
  | DEFAULT VALUES

<insert_column_list> ::= <column_name_list>

<update_statement:_positioned> ::=
    UPDATE <table_name>
    SET <set_clause_list>
    WHERE CURRENT OF <cursor_name>

<set_clause_list> ::=
    <set_clause> [ { <comma> <set_clause> }... ]

<set_clause> ::=
    <object_column> <equals_operator> <update_source>

<object_column> ::= <column_name>

<update_source> ::=
    <value_expression>
  | <null_specification>

```

```

| DEFAULT

<update_statement:_searched> ::=
    UPDATE <table_name>
    SET <set_clause_list>
    [ WHERE <search_condition> ]

<SQL_transaction_statement> ::=
    <set_transaction_statement>
    | <set_constraints_mode_statement>
    | <commit_statement>
    | <rollback_statement>

<set_transaction_statement> ::=
    SET TRANSACTION <transaction_mode>
    [ { <comma> <transaction_mode> }... ]

<transaction_mode> ::=
    <isolation_level>
    | <transaction_access_mode>
    | <diagnostics_size>

<isolation_level> ::=
    ISOLATION LEVEL <level_of_isolation>

<level_of_isolation> ::=
    READ UNCOMMITTED
    | READ COMMITTED
    | REPEATABLE READ
    | SERIALIZABLE

<transaction_access_mode> ::=
    READ ONLY
    | READ WRITE

<diagnostics_size> ::=
    DIAGNOSTICS SIZE <number_of_conditions>

<number_of_conditions> ::= <simple_value_specification>

<set_constraints_mode_statement> ::=
    SET CONSTRAINTS <constraint_name_list>
    { DEFERRED | IMMEDIATE }

<constraint_name_list> ::=
    ALL
    | <constraint_name> [ { <comma> <constraint_name> }... ]

<commit_statement> ::=
    COMMIT [ WORK ]

<rollback_statement> ::=
    ROLLBACK [ WORK ]

<SQL_connection_statement> ::=
    <connect_statement>
    | <set_connection_statement>
    | <disconnect_statement>

<connect_statement> ::=

```

```
CONNECT TO <connection_target>

<connection_target> ::=
    <SQL-server_name>
    [ AS <connection_name> ]
    correspondence with Tony Gordon)
    [ USER <user_name> ]
    | DEFAULT

<SQL-server_name> ::= <simple_value_specification>

<connection_name> ::= <simple_value_specification>

<user_name> ::= <simple_value_specification>

<set_connection_statement> ::=
    SET CONNECTION <connection_object>

<connection_object> ::=
    DEFAULT
    | <connection_name>

<disconnect_statement> ::=
    DISCONNECT <disconnect_object>

<disconnect_object> ::=
    <connection_object>
    | ALL
    | CURRENT

<SQL_session_statement> ::=
    <set_catalog_statement>
    | <set_schema_statement>
    | <set_names_statement>
    | <set_session_authorization_identifier_statement>
    | <set_local_time_zone_statement>

<set_catalog_statement> ::=
    SET CATALOG <value_specification>

<value_specification> ::=
    <literal>
    | <general_value_specification>

<set_schema_statement> ::=
    SET SCHEMA <value_specification>

<set_names_statement> ::=
    SET NAMES <value_specification>

<set_session_authorization_identifier_statement> ::=
    SET SESSION AUTHORIZATION
    <value_specification>

<set_local_time_zone_statement> ::=
    SET TIME ZONE
    <set_time_zone_value>

<set_time_zone_value> ::=
    <interval_value_expression>
```

| LOCAL

```

<SQL_dynamic_statement> ::=
  <system_descriptor_statement>
  | <prepare_statement>
  | <deallocate_prepared_statement>
  | <describe_statement>
  | <execute_statement>
  | <execute_immediate_statement>
  | <SQL_dynamic_data_statement>

<system_descriptor_statement> ::=
  <allocate_descriptor_statement>
  | <deallocate_descriptor_statement>
  | <set_descriptor_statement>
  | <get_descriptor_statement>

<allocate_descriptor_statement> ::=
  ALLOCATE DESCRIPTOR <descriptor_name>
  [ WITH MAX <occurrences> ]

<descriptor_name> ::=
  [ <scope_option> ] <simple_value_specification>

<scope_option> ::=
  GLOBAL
  | LOCAL

<occurrences> ::= <simple_value_specification>

<deallocate_descriptor_statement> ::=
  DEALLOCATE DESCRIPTOR <descriptor_name>

<set_descriptor_statement> ::=
  SET DESCRIPTOR <descriptor_name>
  <set_descriptor_information>

<set_descriptor_information> ::=
  <set_count>
  | VALUE <item_number>
  <set_item_information> [ { <comma> <set_item_information> }... ]

<set_count> ::=
  COUNT <equals_operator> <simple_value_specification_1>

<simple_value_specification_1> ::= <simple_value_specification>

<item_number> ::= <simple_value_specification>

<set_item_information> ::=
  <descriptor_item_name> <equals_operator> <simple_value_specification_2>

<descriptor_item_name> ::=
  TYPE
  | LENGTH
  | OCTET_LENGTH
  | RETURNED_LENGTH
  | RETURNED_OCTET_LENGTH
  | PRECISION
  | SCALE

```

```

| DATETIME_INTERVAL_CODE
| DATETIME_INTERVAL_PRECISION
| NULLABLE
| INDICATOR
| DATA
| NAME
| UNNAMED
| COLLATION_CATALOG
| COLLATION_SCHEMA
| COLLATION_NAME
| CHARACTER_SET_CATALOG
| CHARACTER_SET_SCHEMA
| CHARACTER_SET_NAME

```

```
<simple_value_specification_2> ::= <simple_value_specification>
```

```
<item_number> ::= <simple_value_specification>
```

```
<get_descriptor_statement> ::=
  GET DESCRIPTOR <descriptor_name> <get_descriptor_information>
```

```
<get_descriptor_information> ::=
  <get_count>
  | VALUE <item_number>
    <get_item_information> [ { <comma> <get_item_information> }... ]
```

```
<get_count> ::=
  <simple_target_specification_1> <equals_operator>
  COUNT
```

```
<simple_target_specification_1> ::= <simple_target_specification>
```

```
<simple_target_specification> ::=
  <parameter_name>
  | <embedded_variable_name>
```

```
<get_item_information> ::=
  <simple_target_specification_2> <equals_operator> <descriptor_item_name>>
```

```
<simple_target_specification_2> ::= <simple_target_specification>
```

```
<prepare_statement> ::=
  PREPARE <SQL_statement_name> FROM <SQL_statement_variable>
```

```
<SQL_statement_name> ::=
  <statement_name>
  | <extended_statement_name>
```

```
<extended_statement_name> ::=
  [ <scope_option> ] <simple_value_specification>
```

```
<SQL_statement_variable> ::= <simple_value_specification>
```

```
<deallocate_prepared_statement> ::=
  DEALLOCATE PREPARE <SQL_statement_name>
```

```
<describe_statement> ::=
  <describe_input_statement>
  | <describe_output_statement>
```

```

<describe_input_statement> ::=
    DESCRIBE INPUT <SQL_statement_name> <using_descriptor>

<using_descriptor> ::=
    { USING | INTO } SQL DESCRIPTOR <descriptor_name>

<describe_output_statement> ::=
    DESCRIBE [ OUTPUT ] <SQL_statement_name> <using_descriptor>

<execute_statement> ::=
    EXECUTE <SQL_statement_name>
    [ <result_using_clause> ]
    [ <parameter_using_clause> ]

<result_using_clause> ::= <using_clause>

<using_clause> ::=
    <using_arguments>
    | <using_descriptor>

<using_arguments> ::=
    { USING | INTO } <argument> [ { <comma> <argument> }... ]

<argument> ::= <target_specification>

<parameter_using_clause> ::= <using_clause>

<execute_immediate_statement> ::=
    EXECUTE IMMEDIATE <SQL_statement_variable>

<SQL_dynamic_data_statement> ::=
    <allocate_cursor_statement>
    | <dynamic_open_statement>
    | <dynamic_fetch_statement>
    | <dynamic_close_statement>
    | <dynamic_delete_statement:positioned>
    | <dynamic_update_statement:positioned>

<allocate_cursor_statement> ::=
    ALLOCATE <extended_cursor_name> [ INSENSITIVE ]
    [ SCROLL ] CURSOR
    FOR <extended_statement_name>

<extended_cursor_name> ::=
    [ <scope_option> ] <simple_value_specification>

<dynamic_open_statement> ::=
    OPEN <dynamic_cursor_name> [ <using_clause> ]

<dynamic_cursor_name> ::=
    <cursor_name>
    | <extended_cursor_name>

<dynamic_fetch_statement> ::=
    FETCH [ [ <fetch_orientation> ] FROM ] <dynamic_cursor_name>
    <using_clause>

<dynamic_close_statement> ::=
    CLOSE <dynamic_cursor_name>

```

```

<dynamic_delete_statement:_positioned> ::=
    DELETE FROM <table_name>
        WHERE CURRENT OF
            <dynamic_cursor_name>

<dynamic_update_statement:_positioned> ::=
    UPDATE <table_name>
        SET <set_clause>
            [ { <comma> <set_clause> }... ]
        WHERE CURRENT OF
            <dynamic_cursor_name>

<SQL_diagnostics_statement> ::=
    <get_diagnostics_statement>

<get_diagnostics_statement> ::=
    GET DIAGNOSTICS <sql_diagnostics_information>

<sql_diagnostics_information> ::=
    <statement_information>
    | <condition_information>

<statement_information> ::=
    <statement_information_item> [ { <comma> <statement_information_item> }... ]

<statement_information_item> ::=
    <simple_target_specification> <equals_operator> <statement_information_item_name>

<statement_information_item_name> ::=
    NUMBER
    | MORE
    | COMMAND_FUNCTION
    | DYNAMIC_FUNCTION
    | ROW_COUNT

<condition_information> ::=
    EXCEPTION <condition_number>
        <condition_information_item> [ { <comma> <condition_information_item> }... ]

<condition_number> ::= <simple_value_specification>

<condition_information_item> ::=
    <simple_target_specification> <equals_operator> <condition_information_item_name>

<condition_information_item_name> ::=
    CONDITION_NUMBER
    | RETURNED_SQLSTATE
    | CLASS_ORIGIN
    | SUBCLASS_ORIGIN
    | SERVER_NAME
    | CONNECTION_NAME
    | CONSTRAINT_CATALOG
    | CONSTRAINT_SCHEMA
    | CONSTRAINT_NAME
    | CATALOG_NAME
    | SCHEMA_NAME
    | TABLE_NAME
    | COLUMN_NAME
    | CURSOR_NAME
    | MESSAGE_TEXT

```

```

| MESSAGE_LENGTH
| MESSAGE_OCTET_LENGTH

<embedded_SQL_host_program> ::=
  <embedded_SQL_Ada_program>
| <embedded_SQL_C_program>
| <embedded_SQL_COBOL_program>
| <embedded_SQL_Fortran_program>
| <embedded_SQL_MUMPS_program>
| <embedded_SQL_Pascal_program>
| <embedded_SQL_PL/I_program>

<embedded_SQL_Ada_program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded_SQL_C_program> ::=
  !! <EMPHASIS>(See the Syntax Rules.)

<embedded_SQL_COBOL_program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded_SQL_Fortran_program> ::=
  !! <EMPHASIS>(See the Syntax Rules.)

<embedded_SQL_MUMPS_program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded_SQL_Pascal_program> ::=
  !! <EMPHASIS>(See the Syntax Rules.)

<embedded_SQL_PL/I_program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded_SQL_declare_section> ::=
  <embedded_SQL_begin_declare>
  [ <embedded_character_set_declaration> ]
  [ <host_variable_definition>... ]
  <embedded_SQL_end_declare>
| <embedded_SQL_MUMPS_declare>

<embedded_SQL_begin_declare> ::=
  <SQL_prefix> BEGIN DECLARE SECTION
  [ <SQL_terminator> ]

<SQL_prefix> ::=
  EXEC SQL
| <ampersand>SQL

<SQL_terminator> ::=
  END-EXEC
| <semicolon>
| <right_paren>

<embedded_character_set_declaration> ::=
  SQL NAMES ARE <character_set_specification>

<host_variable_definition> ::=
  <Ada_variable_definition>
| <C_variable_definition>
| <COBOL_variable_definition>
| <Fortran_variable_definition>
| <MUMPS_variable_definition>
| <Pascal_variable_definition>
| <PL/I_variable_definition>

```

```

<Ada_variable_definition> ::=
  <Ada_host_identifier> [ { <comma> <Ada_host_identifier> }... ] :
  <Ada_type_specification> [ <Ada_initial_value> ]

<Ada_type_specification> ::=
  <Ada_qualified_type_specification>
  | <Ada_unqualified_type_specification>

<Ada_qualified_type_specification> ::=
  SQL_STANDARD.CHAR [ CHARACTER SET
    [ IS ] <character_set_specification> ]
  <left_paren> 1 <double_period> <length> <right_paren>
  | SQL_STANDARD.BIT
    <left_paren> 1 <double_period> <length> <right_paren>
  | SQL_STANDARD.SMALLINT
  | SQL_STANDARD.INT
  | SQL_STANDARD.REAL
  | SQL_STANDARD.DOUBLE_PRECISION
  | SQL_STANDARD.SQLCODE_TYPE
  | SQL_STANDARD.SQLSTATE_TYPE
  | SQL_STANDARD.INDICATOR_TYPE

<Ada_unqualified_type_specification> ::=
  CHAR
    <left_paren> 1 <double_period> <length> <right_paren>
  | BIT
    <left_paren> 1 <double_period> <length> <right_paren>
  | SMALLINT
  | INT
  | REAL
  | DOUBLE_PRECISION
  | SQLCODE_TYPE
  | SQLSTATE_TYPE
  | INDICATOR_TYPE

<Ada_initial_value> ::=
  <Ada_assignment_operator> <character_representation>...

<Ada_assignment_operator> ::= <colon>

<C_variable_definition> ::=
  [ <C_storage_class> ]
  [ <C_class_modifier> ]
  <C_variable_specification>
  <semicolon>

<C_storage_class> ::=
  auto
  | extern
  | static

<C_class_modifier> ::= const | volatile

<C_variable_specification> ::=
  <C_numeric_variable>
  | <C_character_variable>
  | <C_derived_variable>

<C_numeric_variable> ::=

```

```

    { long | short | float | double }
    <C_host_identifier> [ <C_initial_value> ]
      [ { <comma> <C_host_identifier> [ <C_initial_value> ] }... ]

<C_initial_value> ::=
    <equals_operator> <character_representation>...

<C_character_variable> ::=
    char [ CHARACTER SET
      [ IS ] <character_set_specification> ]
    <C_host_identifier>
    <C_array_specification> [ <C_initial_value> ]
    [ { <comma> <C_host_identifier>
      <C_array_specification>
      [ <C_initial_value> ] }... ]

<C_array_specification> ::=
    <left_bracket> <length> <right_bracket>

<C_derived_variable> ::=
    <C_VARCHAR_variable>
    | <C_bit_variable>

<C_VARCHAR_variable> ::=
    VARCHAR [ CHARACTER SET [ IS ]
      <character_set_specification> ]
    <C_host_identifier>
    <C_array_specification> [ <C_initial_value> ]
    [ { <comma> <C_host_identifier>
      <C_array_specification>
      [ <C_initial_value> ] }... ]

<C_bit_variable> ::=
    BIT <C_host_identifier>
    <C_array_specification> [ <C_initial_value> ]
    [ { <comma> <C_host_identifier>
      <C_array_specification>
      [ <C_initial_value> ] }... ]

<COBOL_variable_definition> ::=
    {01|77} <COBOL_host_identifier> <COBOL_type_specification>
    [ <character_representation>... ] <period>

<COBOL_type_specification> ::=
    <COBOL_character_type>
    | <COBOL_bit_type>
    | <COBOL_numeric_type>
    | <COBOL_integer_type>

<COBOL_character_type> ::=
    [ CHARACTER SET [ IS ]
      <character_set_specification> ]
    { PIC | PICTURE } [ IS ] { X [ <left_paren> <length> <right_paren> ] }...

<COBOL_bit_type> ::=
    { PIC | PICTURE } [ IS ]
      { B [ <left_paren> <length> <right_paren> ] }...

<COBOL_numeric_type> ::=
    { PIC | PICTURE } [ IS ]

```

```

    S <COBOL_nines_specification>
    [ USAGE [ IS ] ] DISPLAY SIGN LEADING SEPARATE

<COBOL_nines_specification> ::=
    <COBOL_nines> [ V [ <COBOL_nines> ] ]
    | V <COBOL_nines>

<COBOL_nines> ::= { 9 [ <left_paren> <length> <right_paren> ] }...

<COBOL_integer_type> ::=
    <COBOL_computational_integer>
    | <COBOL_binary_integer>

<COBOL_computational_integer> ::=
    { PIC | PICTURE } [ IS ] S
    [ USAGE [ IS ] ] { COMP | COMPUTATIONAL }

<COBOL_binary_integer> ::=
    { PIC | PICTURE } [ IS ] S
    [ USAGE [ IS ] ] BINARY

<Fortran_variable_definition> ::=
    <Fortran_type_specification>
    <Fortran_host_identifier>
    [ { <comma> <Fortran_host_identifier> }... ]

<Fortran_type_specification> ::=
    CHARACTER [ <asterisk> <length> ]
    [ CHARACTER SET [ IS ]
      <character_set_specification> ]
    | BIT [ <asterisk> <length> ]
    | INTEGER
    | REAL
    | DOUBLE PRECISION

<MUMPS_variable_definition> ::=
    { <MUMPS_numeric_variable> | <MUMPS_character_variable> }
    <semicolon>

<MUMPS_numeric_variable> ::=
    <MUMPS_type_specification>
    <MUMPS_host_identifier> [ { <comma> <MUMPS_host_identifier> }... ]

<MUMPS_type_specification> ::=
    INT
    | DEC
    [ <left_paren> <precision> [ <comma> <scale> ] <right_paren> ]
    | REAL

<MUMPS_character_variable> ::=
    VARCHAR <MUMPS_host_identifier> <MUMPS_length_specification>
    [ { <comma> <MUMPS_host_identifier> <MUMPS_length_specification> }... ]

<MUMPS_length_specification> ::=
    <left_paren> <length> <right_paren>

<Pascal_variable_definition> ::=
    <Pascal_host_identifier> [ { <comma> <Pascal_host_identifier> }... ] <colon>
    <Pascal_type_specification> <semicolon>

```

```

<Pascal_type_specification> ::=
    PACKED ARRAY
        <left_bracket> 1 <double_period> <length> <right_bracket>
    OF CHAR
        [ CHARACTER SET [ IS ]
            <character_set_specification> ]
    | PACKED ARRAY
        <left_bracket> 1 <double_period> <length> <right_bracket>
    OF BIT
    | INTEGER
    | REAL
    | CHAR [ CHARACTER SET
                                                [ IS ] <character_set_specification> ]
    | BIT

<PL/I_variable_definition> ::=
    {DCL | DECLARE}
        {
            <PL/I_host_identifier>
            | <left_paren> <PL/I_host_identifier>
                [ { <comma> <PL/I_host_identifier> }... ] <right_paren> }
    <PL/I_type_specification>
    [ <character_representation>... ] <semicolon>

<PL/I_type_specification> ::=
    { CHAR | CHARACTER } [ VARYING ]
        <left_paren>
        [ CHARACTER SET
            [ IS ] <character_set_specification> ]
    | BIT [ VARYING ] <left_paren>
    | <PL/I_type_fixed_decimal> <left_paren> <precision>
        [ <comma> <scale> ] <right_paren>
    | <PL/I_type_fixed_binary> [ <left_paren> <precision> <right_paren> ]
    | <PL/I_type_float_binary> <left_paren> <precision> <right_paren>

<PL/I_type_fixed_decimal> ::=
    { DEC | DECIMAL } FIXED
    | FIXED { DEC | DECIMAL }

<PL/I_type_fixed_binary> ::=
    { BIN | BINARY } FIXED
    | FIXED { BIN | BINARY }

<PL/I_type_float_binary> ::=
    { BIN | BINARY } FLOAT
    | FLOAT { BIN | BINARY }

<embedded_SQL_end_declare> ::=
    <SQL_prefix> END DECLARE SECTION
        [ <SQL_terminator> ]

<embedded_SQL_MUMPS_declare> ::=
    <SQL_prefix>
    BEGIN DECLARE SECTION
        [ <embedded_character_set_declaration> ]
        [ <host_variable_definition>... ]
    END DECLARE SECTION
    <SQL_terminator>

<embedded_SQL_statement> ::=
    <SQL_prefix>

```

```

    <statement_or_declaration>
  [ <SQL_terminator> ]

<statement_or_declaration> ::=
  <declare_cursor>
  | <dynamic_declare_cursor>
  | <temporary_table_declaration>
  | <embedded_exception_declaration>
  | <SQL_procedure_statement>

<embedded_exception_declaration> ::=
  WHENEVER <condition> <condition_action>

<condition> ::=
  SQLERROR | NOT FOUND

<condition_action> ::=
  CONTINUE | <go_to>

<go_to> ::=
  { GOTO | GO TO } <goto_target>

<goto_target> ::=
  <host_label_identifier>
  | <unsigned_integer>
  | <host_PL/I_label_variable>

<host_label_identifier> ::= !(See the Syntax Rules.)

<host_PL/I_label_variable> ::= !(See the Syntax Rules.)

<preparable_statement> ::=
  <preparable_SQL_data_statement>
  | <preparable_SQL_schema_statement>
  | <preparable_SQL_transaction_statement>
  | <preparable_SQL_session_statement>
  | <preparable_implementation-defined_statement>

<preparable_SQL_data_statement> ::=
  <delete_statement:searched>
  | <dynamic_single_row_select_statement>
  | <insert_statement>
  | <dynamic_select_statement>
  | <update_statement:searched>
  | <preparable_dynamic_delete_statement:positioned>
  | <preparable_dynamic_update_statement:positioned>

<dynamic_single_row_select_statement> ::= <query_specification>

<dynamic_select_statement> ::= <cursor_specification>

<preparable_dynamic_delete_statement:positioned> ::=
  DELETE [ FROM <table_name> ]
  WHERE CURRENT OF <cursor_name>

<preparable_dynamic_update_statement:positioned> ::=
  UPDATE [ <table_name> ]
  SET <set_clause_list>
  WHERE CURRENT OF <cursor_name>

```

```
<preparable_SQL_schema_statement> ::=
    <SQL_schema_statement>

<preparable_SQL_transaction_statement> ::=
    <SQL_transaction_statement>

<preparable_SQL_session_statement> ::=
    <SQL_session_statement>

<preparable_implementation-defined_statement> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<direct_SQL_statement> ::=
    <directly_executable_statement> <semicolon>

<directly_executable_statement> ::=
    <direct_SQL_data_statement>
    | <SQL_schema_statement>
    | <SQL_transaction_statement>
    | <SQL_connection_statement>
    | <SQL_session_statement>
    | <direct_implementation-defined_statement>

<direct_SQL_data_statement> ::=
    <delete_statement: searched>
    | <direct_select_statement: multiple_rows>
    | <insert_statement>
    | <update_statement: searched>
    | <temporary_table_declaration>

<direct_select_statement: multiple_rows> ::=
    <query_expression> [ <order_by_clause> ]

<direct_implementation-defined_statement> ::=
    !!(See the Syntax Rules)

<SQL_object_identifier> ::=
    <SQL_provenance> <SQL_variant>

<SQL_provenance> ::= <arc1> <arc2> <arc3>

<arc1> ::= iso | 1 | iso <left_paren> 1 <right_paren>

<arc2> ::= standard | 0 | standard <left_paren> 0 <right_paren>

<arc3> ::= 9075

<SQL_variant> ::= <SQL_edition> <SQL_conformance>

<SQL_edition> ::= <1987> | <1989> | <1992>

<1987> ::= 0 | edition1987 <left_paren> 0 <right_paren>

<1989> ::= <1989_base> <1989_package>

<1989_base> ::= 1 | edition1989 <left_paren> 1 <right_paren>

<1989_package> ::= <integrity_no> | <integrity_yes>

<integrity_no> ::= 0 | IntegrityNo <left_paren> 0 <right_paren>
```

```
<integrity_yes> ::= 1 | IntegrityYes <left_paren> 1 <right_paren>  
<1992> ::= 2 | edition1992 <left_paren> 2 <right_paren>  
<SQL_conformance> ::= <low> | <intermediate> | <high>  
<low> ::= 0 | Low <left_paren> 0 <right_paren>  
<intermediate> ::= 1 | Intermediate <left_paren> 1 <right_paren>  
<high> ::= 2 | High <left_paren> 2 <right_paren>
```