

## Annex 1 - Minimal SQL

The International Standard for Database Language SQL, ISO/ISO 9075, specifies three levels of conformance for SQL language and SQL implementations:

Entry SQL,  
Intermediate SQL, and  
Full SQL.

Each of these existing SQL conformance levels require the facilities of a full-function SQL processor, i.e. schema definition, data manipulation, transaction management, and access control. New conformance alternatives are needed for non-SQL processors that wish to claim conformance to only a portion of the SQL language. Such processors may be able to provide very sophisticated data retrieval capabilities, but may not be able to allow update of data instances or creation of new schema objects. Since existing SQL levels cut across both the schema definition and data manipulation facilities in the SQL standard, it is necessary to consider each SQL level separately as applied to schema definition or data manipulation.

Consider the SQL leveling rules separately for schema definition and data manipulation. Use the term Schema Definition Language (SDL) to identify SQL language features defined in Clause 11, "Schema definition and manipulation", in the SQL'92 standard, and use the term Data Manipulation Language (DML) to identify SQL language features defined in Clause 13, "Data Manipulation". One is then able to discuss the following alternatives for partial support of the SQL language:

Entry DML	Entry SDL
Intermediate DML	Intermediate SDL
Full DML	Full SDL

There is an additional requirement to specify new Minimal DML and Minimal SDL levels to be used exclusively in the definition of application interfaces that do not pretend to support all of the functionality of a full-function SQL processor. These Minimal definitions are intended for use only by non-SQL processors and cannot be used to claim conformance to the SQL standard as an SQL processor.

Minimal DML will support SQL operations on a single table, with no joins and no subqueries, and with severe limitations on derived columns and set functions. Minimal SDL will support specification of only the simplest views and the simplest SQL tables, using only character string, integer, decimal, and real data types, with no table constraints, with only very limited column constraints, and possibly no support for null values.

Levels of conformance in the SQL standard are specified by Leveling Rules in each clause of the specification. Using the style of the SQL standard, the following subsections specify restrictions that apply for Minimal SDL and Minimal DML in addition to any restrictions for Entry SQL. All Clause and Subclause references, and all syntactic terms delimited by angle brackets (i.e. <...>) are from ISO/IEC 9075:1992.

## Minimal Schema Definition Language

1. A <schema element> contained in a <schema definition> shall be a <table definition> or a <view definition>.
2. A <table element> contained in a <table definition> shall be a <column definition>.
3. A <column constraint> shall not be a <unique specification>, a <references specification>, or a <check constraint definition>; thus a <column constraint> may only specify NOT NULL.
4. In some cases, an SQL/ERI Server implementation at the Minimal SDL level or below may choose not to provide support for SQL null values; if every column of every accessible table is constrained to be NOT NULL, then the implementation may require that every <column definition> in a new <table definition> have an explicit or implicit NOT NULL constraint.
5. The <data type> of a <column definition> shall not specify NUMERIC, FLOAT, or DOUBLE PRECISION; thus a <column definition> may only specify DECIMAL, REAL, INTEGER, SMALLINT, and fixed length CHARACTER string <data type>s.
6. A <view definition> shall not specify WITH CHECK OPTION.
7. The <query expression> contained in a <view definition> shall satisfy the restrictions specified by the Minimal Data Manipulation Language leveling rules below.

## Minimal Data Manipulation Language

1. A <query expression> shall be a <query specification>.
2. A <derived column> in the <select list> of a <query specification> shall be a <value expression primary> that is either a <column reference> or a <set function specification>, and the <derived column> shall not contain an <as clause>.
3. A <set function specification> that is a <derived column> in the <select list> of a <query specification> shall be either COUNT(\*) or a <general set function> whose directly contained <value expression> is a <column reference>.
4. A <table expression> shall not contain a <group by clause> or a <having clause>.
5. The <from clause> contained in a <table expression> shall contain exactly one <table reference>, and that <table reference> shall be a single <table name> without an associated <correlation name>. A <table name> may be qualified to include a <schema name>.
6. A <search condition> contained in an <SQL data statement> shall not contain any <subquery>. Any <predicate> contained in a <search condition> shall be a <comparison predicate> without subqueries, a <between predicate>, a <like predicate>, a <null

predicate>, or an <in predicate> whose <in predicate value> is a parenthesized list of <value specification>s.

7. A <row value constructor> contained in any <predicate> shall have exactly one <row value constructor element> that is a <value expression>.
8. A <value expression> in a <search condition> shall be either a <numeric value expression> or a <string value expression> that is a <character primary>.
9. A <value expression primary> in a <search condition> shall be either a <column reference> or an <unsigned value specification>; thus it may not be a <set function specification> or a <scalar subquery>.
10. A <numeric primary> shall not be a <numeric value function>.
11. A <character primary> shall not be a <character value function>.
12. A <sort key> in a <declare cursor> shall be a <column name>; thus it may not be an <unsigned integer>.

Note: Leveling Rule 2a of Subclause 13.8, "<insert statement>", is incorrect in that it should also allow a <null specification>. This is corrected in SQL Technical Corrigendum 1 [20].