Small. Fast. Reliable.
Choose any three.

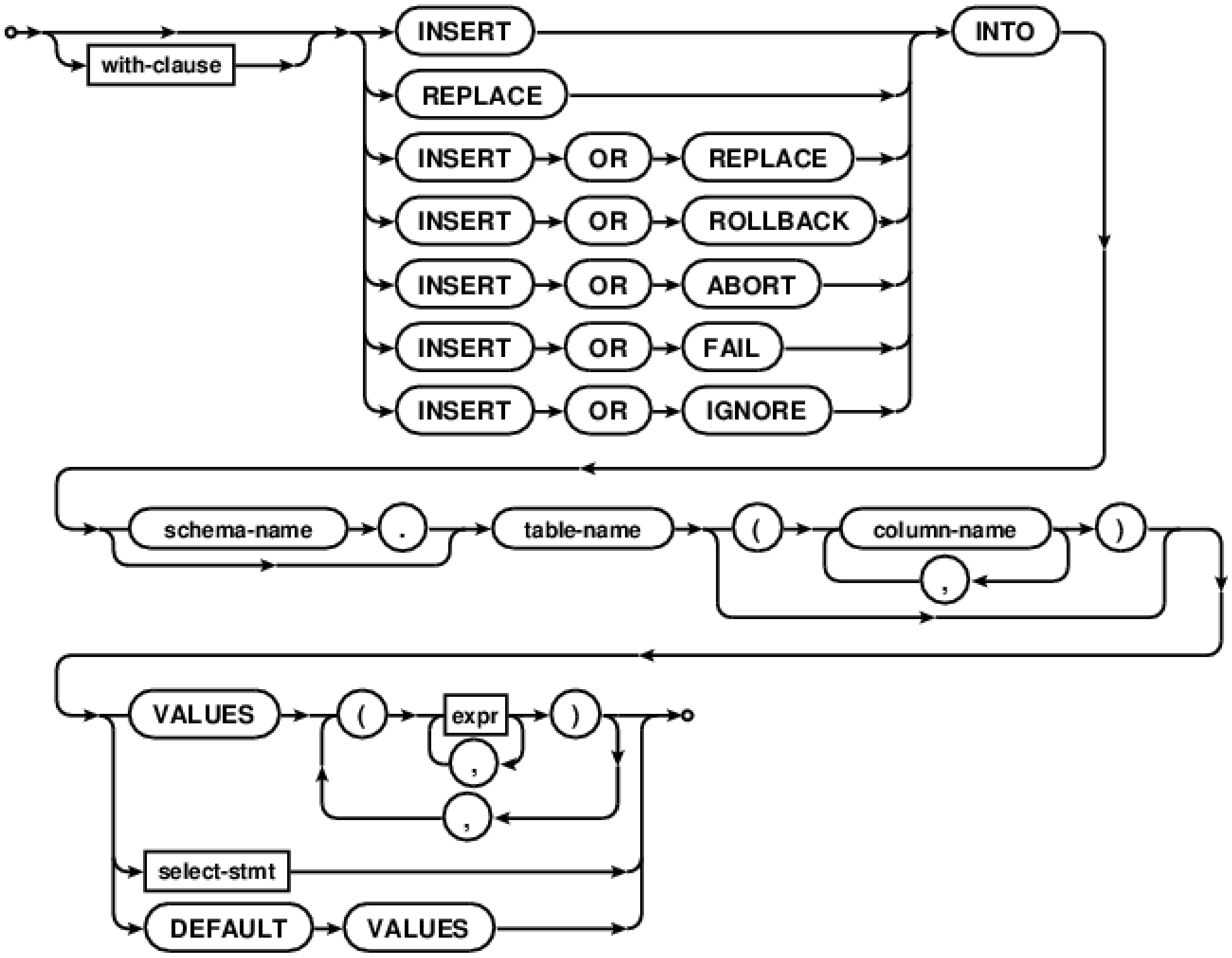
Home Menu About Documentation Download License Support
Purchase Search

SQL As Understood By SQLite

[Top]

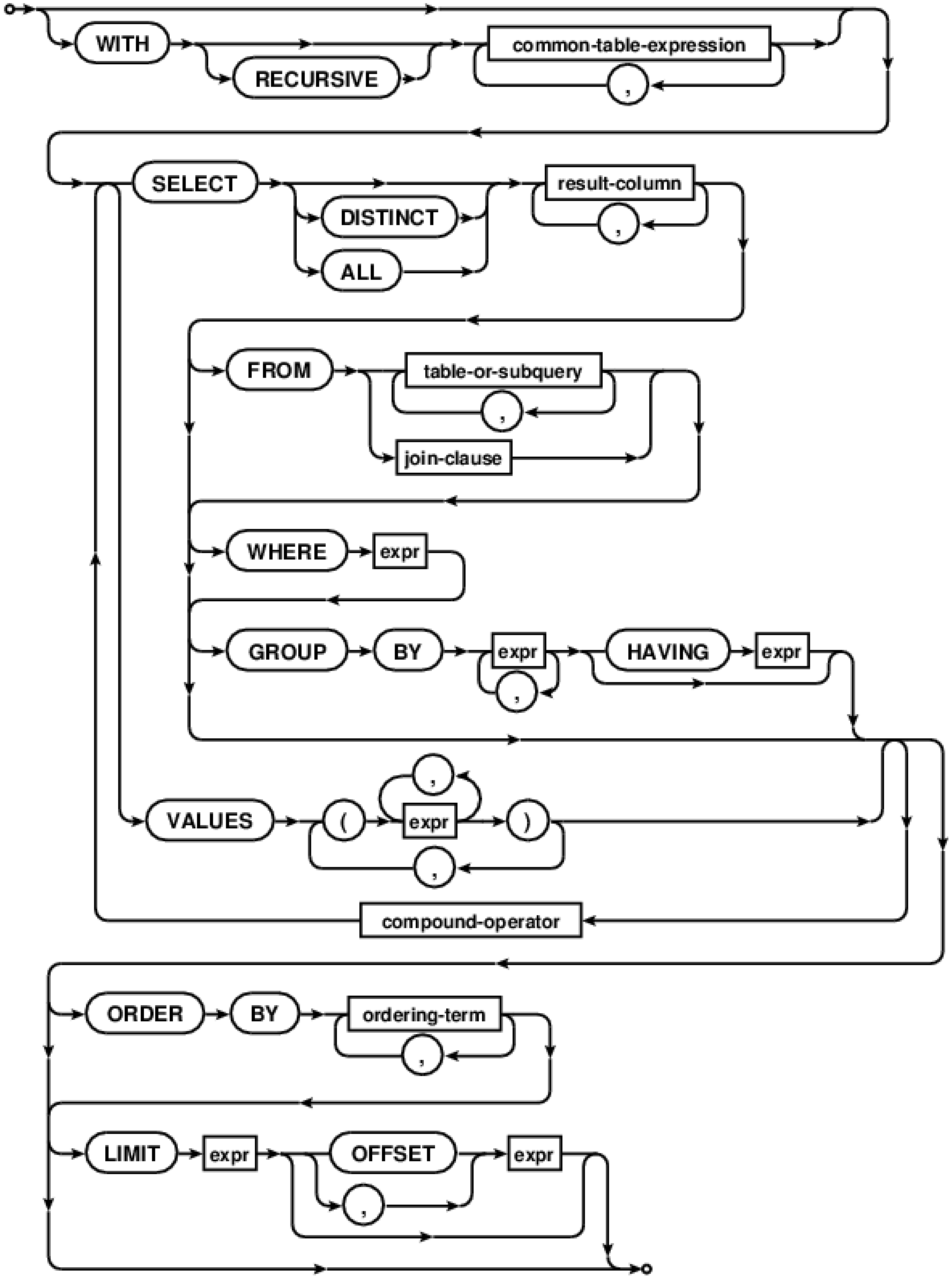
INSERT

insert-stmt: hide



expr: show

select-stmt: hide



common-table-expression: show

compound-operator: show

join-clause: show

ordering-term: show

result-column: show

table-or-subquery: show

with-clause: show

The INSERT statement comes in three basic forms.

1. **INSERT INTO *table* VALUES(...);**

   The first form (with the "VALUES" keyword) creates one or more new rows in an existing table. If the column-name list after table-name is omitted then the number of values inserted into each row must be the same as the number of columns in the table. In this case the result of evaluating the left-most expression from each term of the VALUES list is inserted into the left-most column of each new row, and so forth for each subsequent expression. If a column-name list is specified, then the number of values in each term of the VALUE list must match the number of specified columns. Each of the named columns of the new row is populated with the results of evaluating the corresponding VALUES expression. Table columns that do not appear in the column list are populated with the default column value (specified as part of the CREATE TABLE statement), or with NULL if no default value is specified.

2. **INSERT INTO *table* SELECT ...;**

   The second form of the INSERT statement contains a SELECT statement instead of a VALUES clause. A new entry is inserted into the table for each row of data returned by executing the SELECT statement. If a column-list is specified, the number of columns in the result of the SELECT must be the same as the number of items in the column-list. Otherwise, if no column-list is specified, the number of columns in the result of the SELECT must be the same as the number of columns in the table. Any SELECT statement, including compound SELECTs and SELECT statements with ORDER BY and/or LIMIT clauses, may be used in an INSERT statement of this form.

3. **INSERT INTO *table* DEFAULT VALUES;**

   The third form of an INSERT statement is with DEFAULT VALUES. The INSERT ... DEFAULT VALUES statement inserts a single new row into the named table. Each column of the new row is populated with its default value, or with a NULL if no default value is specified as part of the column definition in the CREATE TABLE statement.

The initial "INSERT" keyword can be replaced by "REPLACE" or "INSERT OR *action*" to specify an alternative constraint conflict resolution algorithm to use during that one INSERT command. For compatibility with MySQL, the parser allows the use of the single keyword REPLACE as an alias for "INSERT OR REPLACE".

The optional "*schema-name***.**" prefix on the table-name is supported for top-level INSERT statements only. The table name must be unqualified for INSERT statements that occur within CREATE TRIGGER statements. Similarly, the "DEFAULT VALUES" form of the INSERT statement is supported for top-level INSERT statements only and not for INSERT statements within triggers.