Small. Fast. Reliable.
Choose any three.

SQL As Understood By SQLite

[Top]

SQLite Keywords

The SQL standard specifies a large number of keywords which may not be used as the names of tables, indices, columns, databases, user-defined functions, collations, virtual table modules, or any other named object. The list of keywords is so long that few people can remember them all. For most SQL code, your safest bet is to never use any English language word as the name of a user-defined object.

If you want to use a keyword as a name, you need to quote it. There are four ways of quoting keywords in SQLite:

'keyword'	A keyword in single quotes is a string literal.
"keyword"	A keyword in double-quotes is an identifier.
[keyword]	A keyword enclosed in square brackets is an identifier. This is not standard SQL. This quoting mechanism is used by MS Access and SQL Server and is included in SQLite for compatibility.
`keyword`	A keyword enclosed in grave accents (ASCII code 96) is an identifier. This is not standard SQL. This quoting mechanism is used by MySQL and is included in SQLite for compatibility.

For resilience when confronted with historical SQL statements, SQLite will sometimes bend the quoting rules above:

- If a keyword in single quotes (ex: **'key'** or **'glob'**) is used in a context where an identifier is allowed but where a string literal is not allowed, then the token is understood to be an identifier instead of a string literal.
- If a keyword in double quotes (ex: **"key"** or **"glob"**) is used in a context where it cannot be resolved to an identifier but where a string literal is allowed, then the token is understood to be a string literal instead of an identifier.

Programmers are cautioned not to use the two exceptions described in the previous bullets. We emphasize that they exist only so that old and ill-formed SQL statements will

run correctly. Future versions of SQLite might raise errors instead of accepting the malformed statements covered by the exceptions above.

SQLite adds new keywords from time to time when it takes on new features. So to prevent your code from being broken by future enhancements, you should normally quote any identifier that is an English language word, even if you do not have to.

The list below shows all possible keywords used by any build of SQLite regardless of compile-time options. Most reasonable configurations use most or all of these keywords, but some keywords may be omitted when SQL language features are disabled. Applications can use the sqlite3_keyword_count(), sqlite3_keyword_name(), and sqlite3_keyword_check() interfaces to determine the keywords recognized by SQLite at run-time. Regardless of the compile-time configuration, any identifier that is not on the following 145 element list is not a keyword to the SQL parser in SQLite:

1. ABORT
2. ACTION
3. ADD
4. AFTER
5. ALL
6. ALTER
7. ALWAYS
8. ANALYZE
9. AND
10. AS
11. ASC
12. ATTACH
13. AUTOINCREMENT
14. BEFORE
15. BEGIN
16. BETWEEN
17. BY
18. CASCADE
19. CASE
20. CAST
21. CHECK
22. COLLATE
23. COLUMN
24. COMMIT
25. CONFLICT
26. CONSTRAINT
27. CREATE
28. CROSS
29. CURRENT
30. CURRENT_DATE
31. CURRENT_TIME
32. CURRENT_TIMESTAMP
33. DATABASE
34. DEFAULT
35. DEFERRABLE
36. DEFERRED
37. DELETE

38. DESC
39. DETACH
40. DISTINCT
41. DO
42. DROP
43. EACH
44. ELSE
45. END
46. ESCAPE
47. EXCEPT
48. EXCLUDE
49. EXCLUSIVE
50. EXISTS
51. EXPLAIN
52. FAIL
53. FILTER
54. FIRST
55. FOLLOWING
56. FOR
57. FOREIGN
58. FROM
59. FULL
60. GENERATED
61. GLOB
62. GROUP
63. GROUPS
64. HAVING
65. IF
66. IGNORE
67. IMMEDIATE
68. IN
69. INDEX
70. INDEXED
71. INITIALLY
72. INNER
73. INSERT
74. INSTEAD
75. INTERSECT
76. INTO
77. IS
78. ISNULL
79. JOIN
80. KEY
81. LAST
82. LEFT
83. LIKE
84. LIMIT
85. MATCH
86. NATURAL
87. NO
88. NOT

89. NOTHING
90. NOTNULL
91. NULL
92. NULLS
93. OF
94. OFFSET
95. ON
96. OR
97. ORDER
98. OTHERS
99. OUTER
100. OVER
101. PARTITION
102. PLAN
103. PRAGMA
104. PRECEDING
105. PRIMARY
106. QUERY
107. RAISE
108. RANGE
109. RECURSIVE
110. REFERENCES
111. REGEXP
112. REINDEX
113. RELEASE
114. RENAME
115. REPLACE
116. RESTRICT
117. RIGHT
118. ROLLBACK
119. ROW
120. ROWS
121. SAVEPOINT
122. SELECT
123. SET
124. TABLE
125. TEMP
126. TEMPORARY
127. THEN
128. TIES
129. TO
130. TRANSACTION
131. TRIGGER
132. UNBOUNDED
133. UNION
134. UNIQUE
135. UPDATE
136. USING
137. VACUUM
138. VALUES
139. VIEW

140. VIRTUAL
141. WHEN
142. WHERE
143. WINDOW
144. WITH
145. WITHOUT