SQL File Evaluation (SQLFE) Tool
SQL Test Documentation

1. SQLFE uses an assignment properties file to specify an answer query for each question, and also to list a set of tests for evaluation of the submitted query. Below is an organized list and description of the tests that are currently implemented.
    a. Each test returns an integer value from 0 to 10 indicating the awarded score. Most tests only return 0 or 10 (that is, not correct or correct), though any test could return any partial credit value, and one test (TestResultSetEqualContent) uses a weighting algorithm comparing rows and columns that allows it to assign any value in the 0 to 10 range.
    b. There are two types of SQL tests:
        i. Condition Tests – names begin with the string *Cond,* and compare the submitted query using a condition built into the test (e.g. counting the number of SELECT keywords) and using an expression (e.g. " >= 2) to evaluate that test by the criteria specified in the expression. For certain COND tests (e.g. CondCompiles), the expression is empty (""), as the entire test condition is coded into the test method itself.
        ii. Test Tests – names begin with the string *Test*, and compare the submitted query to the instructor-specified query or queries in the assignment properties file. Such tests might compare the query text (TestQueryStringEqual), the result set generated by the query (e.g. TestResultSetEqualContent for equality regardless of column order), or other aspects of the query.
    c. The test groups and individual tests are described below.

2. **Correctness Tests – tests that evaluate the overall correctness of the query and/or its results**
    a. TestResultSetEqualContent – compares the result set of the submitted query to the result set of the answer query, regardless of column or row order.
        i. Can return any value between 0 and 10.
    b. TestResultSetEqualExactly – compares the result set of the submitted query to the result set of the answer query, but expects the exact same column order for the columns, and exactly the same row order for the rows.
        i. Returns either 0 or 10.
    c. TestQueryStringEqual – compares the submitted query string to the answer query string. Because of the possibility of different query strings giving the same

answer and the possibility of different spacing between keywords, this test should be used cautiously.

i. Returns either 0 or 10.

d. TestColumnCount – compares the column count of the submitted query's result set to the column count of the answer query's result set.

i. Returns either 0 or 10.

e. TestRowCount – compares the row count of the submitted query's result set to the row count of the answer query's result set.

i. Returns either 0 or 10.

f. TestSameColumnSet – compares the set of column names of the submitted query's result set to the set of column names of the answer query's result set.

i. Returns either 0 or 10.

g. TestTableListEqual – compares the set of table names used in the submitted query to the set of table names found in the answer query.

i. Returns either 0 or 10.

3. **Partial Credit Tests – Cond tests that evaluate basic functionality of the submitted query**

a. CondCompiles – tests if a submitted query correctly compiles and generates a result set.

i. Returns either 0 or 10.

ii. Expression is empty (""), as test is absolute

b. CondBasicContent – tests if a submitted query contains at least one SELECT clause and one FROM clause.

i. Returns either 0 or 10.

ii. Expression is empty (""), as test is absolute

4. **Structure Tests – tests that evaluate other information about structure of query**

a. CondRowCount – tests if a submitted query's result set contains the appropriate number of rows as compared by the expression.

i. Returns either 0 or 10.

ii. Expression is comparative; e.g. " >= 10"

b. CondSubselectDepthCount – tests if a submitted query contains the appropriate level of nested subselect queries as compared by the expression.

i. Returns either 0 or 10.

ii. Expression is comparative; e.g. " >= 2"

c. CondTableCount – tests if a submitted query contains the appropriate number of table references (FROM, explicit JOIN, or comma-separated table join) as compared by the expression.

i. Returns either 0 or 10.

        ii. Expression is comparative; e.g. “ >= 2”

5. **Style Tests – tests for query style and format**
    a. CondNumLinesCount – test if a submitted query contains the number of lines as compared by the expression (basic testing to make sure query isn’t just on one/a few long lines)
        i. Returns either 0 or 10
        ii. Expression is comparative; e.g. “ >= 1”
6. **Keyword Count Tests – Cond tests that count the number of occurrences of a certain keyword and evaluate that within the context of a numerical expression**
    a. CondAndCount – tests if a submitted query contains the appropriate number of AND keywords as compared by the expression.
        i. Returns either 0 or 10.
        ii. Expression is comparative; e.g. “ >= 1”
    b. CondAvgCount – tests if a submitted query contains the appropriate number of AVG keywords as compared by the expression.
        i. Returns either 0 or 10.
        ii. Expression is comparative; e.g. “ >= 1”
    c. CondCountCount – tests if a submitted query contains the appropriate number of COUNT keywords as compared by the expression.
        i. Returns either 0 or 10.
        ii. Expression is comparative; e.g. “ >= 1”
    d. CondDescCount – tests if a submitted query contains the appropriate number of DESC or DESCENDING keywords as compared by the expression.
        i. Returns either 0 or 10.
        ii. Expression is comparative; e.g. “ >= 1”
    e. CondDistinctCount – tests if a submitted query contains the appropriate number of DISTINCT keywords as compared by the expression.
        i. Returns either 0 or 10.
        ii. Expression is comparative; e.g. “ >= 1”
    f. CondFromCount – tests if a submitted query contains the appropriate number of FROM keywords as compared by the expression.
        i. Returns either 0 or 10.
        ii. Expression is comparative; e.g. “ >= 1”
    g. CondGroupByCount – tests if a submitted query contains the appropriate number of GROUP BY keywords as compared by the expression.
        i. Returns either 0 or 10.
        ii. Expression is comparative; e.g. “ >= 1”

h. CondHavingCount – tests if a submitted query contains the appropriate number of HAVING keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
i. CondIntersectCount – tests if a submitted query contains the appropriate number of INTERSECT keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
j. CondJoinCount – tests if a submitted query contains the appropriate number of JOIN keywords (of any variation) as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
k. CondJoinCrossCount – tests if a submitted query contains the appropriate number of CROSS JOIN keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
l. CondJoinExplicitCount – tests if a submitted query contains the appropriate number of AND keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
m. CondJoinImplicitCount – tests if a submitted query contains the appropriate number of comma-separated table joins (implicit, or SQL-92 join style) as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
n. CondJoinOuterCount – tests if a submitted query contains the appropriate number of OUTER JOIN keywords (of any variant – LEFT, RIGHT, FULL) as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
o. CondLikeCount – tests if a submitted query contains the appropriate number of LIKE keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”
p. CondMaxCount – tests if a submitted query contains the appropriate number of MAX keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. “ >= 1”

q. CondMinCount – tests if a submitted query contains the appropriate number of MIN keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

r. CondMinusCount – tests if a submitted query contains the appropriate number of MINUS keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

s. CondNotCount – tests if a submitted query contains the appropriate number of NOT keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

t. CondOrCount – tests if a submitted query contains the appropriate number of OR keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

u. CondOrderByCount – tests if a submitted query contains the appropriate number of ORDER BY keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

v. CondSelectCount – tests if a submitted query contains the appropriate number of SELECT keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

w. CondSumCount – tests if a submitted query contains the appropriate number of SUM keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

x. CondUnionCount – tests if a submitted query's result set contains the appropriate number of UNION keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"

y. CondWhereCount – tests if a submitted query's result set contains the appropriate number of WHERE keywords as compared by the expression.
    i. Returns either 0 or 10.
    ii. Expression is comparative; e.g. " >= 1"