

-----inl_test.sql

This test creates 2 tables: table1 and table2.

It selects * the tables to see what is in them.

It then creates an index on table1.id.

It then performs an equijoin to make sure INL is the algorithm used.

It then creates an index on table2.id2.

It then performs the same equijoin to make sure INL is the algorithm used.

It then drops the index on table1.id so that table2.id2 is the only index.

It then performs the same equijoin to make sure INL is the algorithm used.

The tables both have duplicates so the test case makes sure that the duplicates are not ignored.

----join_test.sql

This test case tests which algorithms join.cpp calls with different conditions.

We create 2 table: table1 and table2.

We then perform an equijoin (with no indices on the relations) to see if SMJ is used.

We then perform 2 non-equijoins (with no indices on the relations) to see if SNL is used.

We then create an index on table2.id2.

We then perform 2 non-equijoins(with an index) to see if SNL is still called.

We then perform an equijoin (with an index) to see if INL is used.

---select_test.report

This test case tests the select function vigorously.

It starts by creating 2 tables: table1 and table2

It then selects * on both tables to see if all of the tuples output.

It then performs a simple select query to see if it will correctly print out 1 attribute.

It then prforms a query that prints out 2 attributes that are in the opposite order of

which they are stored in the database to see if they are outputted in the correct order.

It then performs a query with an equality condition to make sure ScanSelect is used.

It then creates an index on table2.id2 and performs the same query to see if IndexScan is now used.

It then performs a query with a non-equality condition.

It then proceeds to test the different operators to make sure they work.

---snl_test.sql

The test creates 2 tables: table1 and table2.

It then selects * from the tables to show what they contain.

It then attempts to do perform a non-equijoin to make sure SNL is the algorithm used.

It then creates an index table1(id) and then attempts to do perform the same non-equijoin to make sure SNL is the algorithm used still.

It then performs non-equijoins using all of the different operators to make sure the records in the 2 tables are matched correctly.

The tables both have duplicates so the test case makes sure that the duplicates are not ignored.

---insert_smj_join.sql

The test case creates two tables: soaps and stars

It first tests regular insert query

Then tests insert without order

Then tests insert with null value

Then tests insert with value exceeding length limit

Then test SMJ join algorithm based on non-duplicate value and duplicate value