INSENSITIVE (SQL-92 Syntax)

The insensitive cursor executes the query and stores the entire set of data that is used for the cursor in tempdb. Thus, the query is only run once and the data is static. The result of this is good performance, since the cursor query is only run once, but no modifications to the cursor can be made, nor are changes made to the underlying tables reflected in the cursor. With an insensitive cursor, only FETCH NEXT is supported.

This makes sense for batch applications, since throughput is going to be better since the cursor query is only going to be executed only once. However, the data set returned is static on a volatile file, the cusor results may not reflect the current data in the table.

SCROLL (SQL-92 Syntax)

Unlike the insensitive cursor, the scroll cursor does support modifications as well as all cursor directional options such as FETCH NEXT, FETCH FIRST, FETCH LAST, FETCH PRIOR, FETCH RELATIVE and FETCH ABSOLUTE. In addition, changes to the underlying tables are reflected in the cursor. The downside is that the cursor is re-evaluated on each call.

This could be a performace hit, if the cursor query is executed repetitively. However, for interactive applications this type cursor definition makes sense when presenting list data for evaluation, changes to the table will be reflected in the cursor and different fetch options, FETCH FIRST, FETCH PRIOR, etc. allow more flexibility in how the information may be presented.

Performance considerations for SQL cursors Edward Whalen, Performance Tuning Corporation

ASENSITIVE (DB2 UDB)

This is the default setting on all cursor definitions. The database manager can implement the cursor as either SENSITIVE or INSENSITIVE depending on optimization of the SQL request. Updateable cursors do not default to ASENSITIVE instead they are always implemented as SENSITIVE cursors.

SENSITIVE (DB2 UDB)

The cursor has some level of sensitivity to any inserts, updates, or deletes made to the associated tables after the cursor has been opened. If the database manager cannot make changes visible to the cursor, then an error is returned on the open request.

Performance and query optimization impacts

The default setting of ASENSITIVE is the best performing option because it allows the query optimizer to use its complete set of algorithms when deciding on the best method for implementing a query. Utilization of an algorithm that makes a copy of the data (e.g., hashing) can drastically improve performance and other queries where performance is better when copies of the data are avoided. The ASENSITIVE setting gives the query optimizer the freedom to choose the best performing method. The SENSITIVE and INSENSITVE cursor settings can force the optimizer into a plan that is sub-optimal from a performance point of view. The SENSITIVE setting eliminates the usage of temporary data copies by the optimizer; this also prevents parallel processing since the DB2 Symmetric MultiProcessing (DB2 SMP) feature makes copies of the table data. The INSENSITIVE setting forces a copy of the table data whether it it is good for performance or not. Thus, the INSENSITIVE and SENSITIVE cursor settings should be used **only** when the associated cursor behavior is absolutely required by the application.

DB2 UDB Cursor technical article by Ken Milligan, DB2 UDB Technology Specialist, SDI Corp.