

RE: Xlocking with a select statement

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-04/4508.html>

From: vishal subramaniam (*vishalsu_at_microsoft.com*)

Date: 04/26/04

Date: Mon, 26 Apr 2004 10:22:52 GMT

vishalsu@online.microsoft.com

RESOLUTION/LINKS:

=====

SELECT Statement (named_select_statement)

A SELECT statement (named_select_statement) defines and creates a result table with the name result_table_name (see named/unnamed result table).

Syntax

```
<named_select_statement> ::= <named_query_expression>  
[<order_clause>] [<update_clause>] [<lock_option>] [FOR REUSE]
```

named query expression, order clause, update clause, lock option

Explanation

An OPEN CURSOR statement is not permitted for result tables created with this SELECT statement.

The SELECT statement (named select statement) is subject to the rules that were specified for the DECLARE CURSOR statement and those that were specified for the OPEN CURSOR statement.

Depending on the search strategy, either all the rows in the result table are searched when the SELECT statement (named select statement) is executed and the result table is physically generated, or each next result table row is searched when a FETCH statement is executed, without being physically stored. This must be taken into account for the time behavior of FETCH statements.

Updateable result table

A result table or the underlying base tables are updateable if the query

statement satisfies the following conditions:

The QUERY expression (named_query_expression) must only comprise one QUERY specification (named_query_spec).

Only one base table or one updateable view table may be specified in the FROM clause of the QUERY specification (named query spec).

The DISTINCT keyword (see DISTINCT specification), a GROUP clause, or HAVING clause must not be specified.

Expressions must not contain a set function (set_function_spec).

See also the section entitled "Updateable result table" under QUERY statement.

ORDER clause

The ORDER clause specifies a sort sequence for a result table.

UPDATE clause

An UPDATE clause can only be specified for updateable result tables. For updateable result tables, a position within a particular result table always corresponds to a position in the underlying tables and thus, ultimately, to a position in one or more base tables.

If an UPDATE clause was specified, the base tables can be updated using the position in the result table (CURRENT OF <result table name>) by means of an UPDATE statement or a DELETE statement. A lock can be requested for the affected lines of each of the affected base tables using a LOCK statement.

LOCK option

The LOCK option determines which locks are to be set on the read rows.

FOR REUSE

If the result table is to be specified in the from clause of a subsequent QUERY statement, the table should be specified with FOR REUSE keywords. If FOR REUSE is not specified, the reusability of the result table depends on internal system strategies.

Since specifying FOR REUSE increases the response times of some query statements, it should only be specified if it is required to reuse the result table.

SELECT Statement (select_statement)

A SELECT statement (select_statement) defines and creates an unnamed result table (see named/unnamed result table).

Syntax

```
<select_statement> ::= <query_expression> [<order_clause>]  
[<update_clause>] [<lock_option>] [FOR REUSE]
```

query expression, order clause, update clause, lock option

Explanation

An OPEN CURSOR statement is not permitted for result tables created with this SELECT statement.

The SELECT statement (select_statement) is subject to the rules that were specified for the DECLARE CURSOR statement and those that were specified for the OPEN CURSOR statement.

Depending on the search strategy, either all the rows in the result table are searched when the SELECT statement (select_statement) is executed and the result table is physically generated, or each next result table row is searched when a FETCH statement is executed, without being physically stored. This must be taken into account for the time behavior of FETCH statements.

Updateable result table

A result table or the underlying base tables are updateable if the query statement satisfies the following conditions:

The QUERY statement comprises a DECLARE CURSOR statement.

The QUERY expression (query_expression) must only comprise one QUERY specification (query_spec).

Only one base table or one updateable view table may be specified in the FROM clause of the QUERY specification (query_spec).

The DISTINCT keyword (see DISTINCT specification), a GROUP clause, or HAVING clause must not be specified.

Expressions must not contain a set function (set_function_spec).

See also the section entitled "Updateable result table" under QUERY statement.

ORDER clause

The ORDER clause specifies a sort sequence for a result table.

UPDATE clause

An UPDATE clause can only be specified for updateable result tables. For updateable result tables, a position within a particular result table always corresponds to a position in the underlying tables and thus, ultimately, to a position in one or more base tables.

If an UPDATE clause was specified, the base tables can be updated using the position in the result table (CURRENT OF <result table name>) by means of

an UPDATE statement or a DELETE statement. A lock can be requested for the affected lines of each of the affected base tables using a LOCK statement.

LOCK option

The LOCK option determines which locks are to be set on the read rows.

FOR REUSE

If the result table is to be specified in the from clause of a subsequent QUERY statement, the table should be specified with FOR REUSE keywords. If FOR REUSE is not specified, the reusability of the result table depends on internal system strategies.

Since specifying FOR REUSE increases the response times of some query statements, it should only be specified if it is required to reuse the result table.

=====
===

LOCK Option (lock_option)

The LOCK option requests a lock for each selected row.

Syntax

```
<lock_option> ::=  
  WITH LOCK [(IGNORE)|(NOWAIT)] [EXCLUSIVE|OPTIMISTIC] [ISOLATION LEVEL  
<unsigned_integer>]
```

unsigned_integer may only have the values 0, 1, 2, 3, 10, 15, 20, or 30

Explanation

IGNORE

If (IGNORE) is not specified and a lock collision occurs, the system waits for a locked row to be released (but only as long as is specified by the database parameter REQUEST_TIMEOUT).

If (IGNORE) is specified, the system does not wait for a locked row to be released by another transaction. Instead, it ignores this row if a lock collision occurs. If there is no collision, the requested lock is set. (IGNORE) can only be specified in isolation level 1.

NOWAIT

If (NOWAIT) is not specified and a lock collision occurs, the system waits for the locked data object to be released (but only as long as is specified by the database parameter REQUEST_TIMEOUT).

If (NOWAIT) is specified, the database system does not wait until another user has released a data object. Instead, it returns a message if a collision occurs. If there is no collision, the requested lock is set.

EXCLUSIVE

An exclusive lock is defined. As long as the locked row has not been changed or deleted, the exclusive lock can be released using the UNLOCK statement.

OPTIMISTIC

An optimistic lock is defined on rows. This is only meaningful in connection with isolation levels 0, 1, 10, and 15.

Share Lock

If neither EXCLUSIVE nor OPTIMISTIC is specified, a share lock is set on the corresponding rows.

ISOLATION LEVEL

The locks are set independently of the ISOLATION specification (isolation_spec) of the CONNECT statement. The isolation level of the LOCK option can have a higher or lower value than that in the CONNECT statement.

If an isolation level is specified by the lock option, it is only valid for the duration of the SQL statement which contains the LOCK option specification. Afterwards, the isolation level that was specified in the CONNECT statement is applicable again. In the case of a SELECT statement (named_select_statement), SELECT statement (select_statement), or an OPEN CURSOR statement, for which the result table is not actually physically generated, the specified isolation level is valid for this SQL statement and all FETCH statements that refer to the result table. The isolation level that was specified in the CONNECT statement is applicable for other SQL statements that were executed in the meantime.

This posting is provided "AS IS" with no warranties, and confers no rights.