Locking Hints

SQL Server 2000

A range of table-level locking hints can be specified using the SELECT, INSERT, UPDATE, and DELETE statements to direct Microsoft® SQL Server™ 2000 to the type of locks to be used. Table-level locking hints can be used when a finer control of the types of locks acquired on an object is required. These locking hints override the current transaction isolation level for the session.

Note The SQL Server query optimizer automatically makes the correct determination. It is recommended that table-level locking hints be used to change the default locking behavior only when necessary. Disallowing a locking level can affect concurrency adversely.

Locking hint	**Description**
HOLDLOCK	Hold a shared lock until completion of the transaction instead of releasing the lock as soon as the required table, row, or data page is no longer required. HOLDLOCK is equivalent to SERIALIZABLE.
NOLOCK	Do not issue shared locks and do not honor exclusive locks. When this option is in effect, it is possible to read an uncommitted transaction or a set of pages that are rolled back in the middle of a read. Dirty reads are possible. Only applies to the SELECT statement.
PAGLOCK	Use page locks where a single table lock would usually be taken.
READCOMMITTED	Perform a scan with the same locking semantics as a transaction running at the READ COMMITTED isolation level. By default, SQL Server 2000 operates at this isolation level.
READPAST	Skip locked rows. This option causes a transaction to skip rows locked by other transactions that would ordinarily appear in the result set, rather than block the transaction waiting for the other transactions to release their locks on these rows. The READPAST lock hint applies only to transactions operating at READ COMMITTED isolation and will read only past row-level locks. Applies only to the SELECT statement.
READUNCOMMITTED	Equivalent to NOLOCK.
REPEATABLEREAD	Perform a scan with the same locking semantics as a transaction running at the REPEATABLE READ isolation level.
ROWLOCK	Use row-level locks instead of the coarser-grained page- and table-level locks.
SERIALIZABLE	Perform a scan with the same locking semantics as a transaction running at the SERIALIZABLE isolation level. Equivalent to HOLDLOCK.
TABLOCK	Use a table lock instead of the finer-grained row- or page-level locks. SQL Server holds this lock until the end of the statement. However, if you also specify HOLDLOCK, the lock is held until the end of the transaction.
TABLOCKX	Use an exclusive lock on a table. This lock prevents others from reading or updating the table and is held until the end of the statement or transaction.
UPDLOCK	Use update locks instead of shared locks while reading a table, and hold locks until the end of the statement or

	transaction. UPDLOCK has the advantage of allowing you to read data (without blocking other readers) and update it later with the assurance that the data has not changed since you last read it.
XLOCK	Use an exclusive lock that will be held until the end of the transaction on all data processed by the statement. This lock can be specified with either PAGLOCK or TABLOCK, in which case the exclusive lock applies to the appropriate level of granularity.

For example, if the transaction isolation level is set to SERIALIZABLE, and the table-level locking hint NOLOCK is used with the SELECT statement, key-range locks typically used to maintain serializable transactions are not taken.

```
USE pubs
GO
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
GO
BEGIN TRANSACTION
SELECT au_lname FROM authors WITH (NOLOCK)
GO
```

The locks generated are:

```
EXEC sp_lock
GO
```

spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	1	0	0	DB		S	GRANT
6	1	0	0	DB		S	GRANT
7	1	0	0	DB		S	GRANT
8	4	0	0	DB		S	GRANT
8	4	0	0	DB		S	GRANT
8	4	117575457	0	TAB		Sch-S	GRANT
9	4	0	0	DB		S	GRANT
9	1	21575115	0	TAB		IS	GRANT

```
SELECT object_name(117575457)
GO
----------------------------
authors
```

The only lock taken that references **authors** is a schema stability (Sch-S) lock. In this case, serializability is no longer guaranteed.

See Also

DELETE[1]

INSERT[2]

SELECT[3]

SET TRANSACTION ISOLATION LEVEL[4]

UPDATE[5]

Links Table

[1] *http://msdn.microsoft.com/en-us/library/aa258847(v=sql.80).aspx*

[2] *http://msdn.microsoft.com/en-us/library/aa933206(v=sql.80).aspx*

[3] *http://msdn.microsoft.com/en-us/library/aa259187(v=sql.80).aspx*

[4] *http://msdn.microsoft.com/en-us/library/aa259216(v=sql.80).aspx*

[5] *http://msdn.microsoft.com/en-us/library/aa260662(v=sql.80).aspx*

Community Content