

NOLOCK Hint

Database Health Reports considers the NOLOCK hint to be SQL Technical Debt by default. This can be overridden in the settings page.



Details

Below is an example of a query selecting an account balance, and a last transaction date from a table called AccountBalance and the WITH (NOLOCK) was used.

```
SELECT SUM(bal.[Balance])  
FROM [banking].[AccountBalance] AS bal WITH (NOLOCK)  
WHERE bal.AccountID = 12345;
```

Assume this was your bank account, and you were attempting to withdraw cash at an ATM. If the NOLOCK hint is used, the account balance may not be accurate, and if your actual account balance was \$500 and you were attempting to withdraw \$400, you should be able to do it. Here is the NOLOCK problem. If other transactions are occurring on the AccountBalance table that might cause a page split to occur, or an clustered index to be rearranged, then with the NOLOCK command is is possible that when you run your query you get one result, then running the query a moment later you get a different result, then run it a third time and you get the original amount.

Imagine an account balance of \$500 that was made up of a \$1000 deposit, and a check for \$500 with the remaining balance of \$500. Imagine that the AccountBalance table has a clustered index on AccountID, and that the data page containing your deposit of \$1000 is full, and based on someone elses transaction that a page split is required. The page split takes the contents of one page in memory and splits it into 2. If this happens at the time of your transaction, then your deposit of \$1000 may get missed, or it may get counted twice, and with your check for \$500 it may get counted twice or missed all together.

The query above should be returning for the balance an amount of \$500, but instead it may return any of the following values when NOLOCK is used.

- \$2000 in the case that the deposit is counted twice and the check is missed due to NOLOCK.
- \$1500 in the case that the deposit is counted twice and the check counted once due to NOLOCK.
- \$1000 in the case that the deposit is counted twice and the check counted twice due to NOLOCK.
- \$0 in the case that the deposit is counted once, and the check is counted twice due to NOLOCK.
- \$-500 in the case that the deposit is missed, and the check is counted once due to NOLOCK.
- \$500 in the case that we get lucky and no page splits occur and we get the right result NOLOCK.

If this was my bank account and I was trying to withdrawal money at an ATM and was rejected because my balance was -\$500 I would be very upset with the bank.

If you would like a consistent read then your query would look like the following and to not use the NOLOCK option:

```
SELECT SUM(bal.[Balance])  
FROM [banking].[AccountBalance] AS bal  
WHERE bal.AccountID = 12345;
```

Technically when you use the WITH (NOLOCK) option the query may run a little bit faster than without it, but the negative impact of not getting consistent results is generally enough of a reason to not use it.

NOLOCK is not the RUN FASTER or TURBO button for a query.

Do you agree?

If you don't agree that that the NOLOCK hint in a query is generally a bad thing, that is just fine, in the Database Health Reports settings dialog you can turn the check. As long as you have coding standards that state one opinion or another then you are doing better than most.

A good coding standard can over-ride many of the items that may be considered as technical debt.

See Also:

[SQL Server Technical Debt](#)