

Re: Transaction and locks

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.clients/2007-01/msg00006.html>

- *From:* "Dejan Sarka" <dejan_please_reply_to_newsgroups.sarka@xxxxxxxxxx>
 - *Date:* Fri, 12 Jan 2007 08:03:01 +0100
-

Huh... You should read a bit more about transactions in Books OnLine. You have some misconcepts here.

This may be a naive question. Since DB Transactions with higher isolation levels internally use locks,

All transactions use locks, even in read uncommitted level you get exclusive locks for updates.

does it make sense to use Transactions for concurrency management , in place of using some kind of pessimistic locks ?

In SQL Server, all updates are automatically transactions. You should specify your own transactions when you need higher grain of atomicity, i.e. if you want to have more than one update joined in a single transaction, so all updates are committed or all are rolled back. Locks are the mechanism for concurrency management. Some isolation levels (read uncommitted, read committed, repeatable read, serializable) are pessimistic, some (read committed snapshot, snapshot) are optimistic. It is not locks that define whether the locking is optimistic or pessimistic; it is the fact whether you have more than one copy of the same data or not.

—
Dejan Sarka

<http://www.solidqualitylearning.com/blogs/>