**SET TRANSACTION ISOLATION LEVEL (Transact-SQL)**

SET TRANSACTION ISOLATION LEVEL

{ READ UNCOMMITTED

| READ COMMITTED

| REPEATABLE READ

| SNAPSHOT

| SERIALIZABLE

}

**READ UNCOMMITTED**  
Specifies that statements can read rows that have been modified by other transactions but not yet committed.

Transactions running at the READ UNCOMMITTED level do not issue shared locks to prevent other transactions from modifying data read by the current transaction. READ UNCOMMITTED transactions are also not blocked by exclusive locks that would prevent the current transaction from reading rows that have been modified but not committed by other transactions. When this option is set, it is possible to read uncommitted modifications, which are called dirty reads. Values in the data can be changed and rows can appear or disappear in the data set before the end of the transaction. This option has the same effect as setting NOLOCK on all tables in all SELECT statements in a transaction. This is the least restrictive of the isolation levels.

**READ COMMITTED**  
Specifies that statements cannot read data that has been modified but not committed by other transactions. This prevents dirty reads. Data can be changed by other transactions between individual statements within the current transaction, resulting in nonrepeatable reads or phantom data. This option is the SQL Server default.

**REPEATABLE READ**  
Specifies that statements cannot read data that has been modified but not yet committed by other transactions and that no other transactions can modify data that has been read by the current transaction until the current transaction completes.

**SNAPSHOT**  
Specifies that data read by any statement in a transaction will be the transactionally consistent version of the data that existed at the start of the transaction. The transaction can only recognize data modifications that were committed before the start of the transaction. Data modifications made by other transactions after the start of the current transaction are not visible to statements executing in the current transaction. The effect is as if the statements in a transaction get a snapshot of the committed data as it existed at the start of the transaction.

Except when a database is being recovered, SNAPSHOT transactions do not request locks when reading data. SNAPSHOT transactions reading data do not block other transactions from writing data. Transactions writing data do not block SNAPSHOT transactions from reading data.

During the roll-back phase of a database recovery, SNAPSHOT transactions will request a lock if an attempt is made to read data that is locked by another transaction that is being rolled back. The SNAPSHOT transaction is blocked until that transaction has been rolled back. The lock is released immediately after it has been granted.

**SERIALIZABLE**  
Specifies the following:

* Statements cannot read data that has been modified but not yet committed by other transactions.
* No other transactions can modify data that has been read by the current transaction until the current transaction completes.
* Other transactions cannot insert new rows with key values that would fall in the range of keys read by any statements in the current transaction until the current transaction completes.