<https://www.sqlservercentral.com/articles/isolation-levels-in-sql-server>

SQL Server isolation levels are used to define the degree to which one transaction must be isolated from resource or data modifications made by other concurrent transactions. The different Isolation Levels are:

1. Read Uncommitted
2. Read Committed
3. Repeatable Read
4. Serializable
5. Snapshot

Read Committed is the default isolation level. However, it can be changed from Query Window as well as Management Studio Tools.

The syntax is:

SET TRANSACTION ISOLATION LEVEL

{READ UNCOMMITTED

| READ COMMITTED

| REPEATABLE READ

| SNAPSHOT

| SERIALIZABLE

}

|  |  |  |  |
| --- | --- | --- | --- |
| solation Level | Dirty read | Non-repeatable read | Phantom Read |
| READ UNCOMMITTED | Possible | Possible | Possible |
| READ COMMITTED | Not Possible | Possible | Possible |
| REPEATABLE READ | Not Possible | Not Possible | Possible |
| SERIALIZABLE | Not Possible | Not Possible | Not Possible |

Read Uncommitted

Transactions running at this level do not issue shared locks to prevent other transactions from modifying data read by the current transaction. Also, transactions are not blocked by exclusive locks at the time of data modification, thus allowing other transactions to read the modified data which is not yet committed.

## **The Significance of Different Isolation Levels**

Only one of the isolation level options can be set at a time, and it remains set for that connection until it is explicitly changed. A lower isolation level increases the ability of many users to access data at the same time, but increases the number of concurrency effects, such as dirty reads or lost updates etc. Conversely, a higher isolation level reduces the types of concurrency effects that users might encounter, but requires more system resources and increases the chances that one transaction will block another.

The lowest isolation level, read uncommitted, can retrieve data that has been modified but not committed by other transactions. All concurrency side effects can happen in read uncommitted, but there's no read locking or versioning, so overhead is minimized.

READ COMMITTED is the default isolation level for SQL Server. It prevents dirty reads by specifying that statements cannot read data values that have been modified but not yet committed by other transactions. If the READ\_COMMITTED\_SNAPSHOT option is set as ON, the Read transactions need not wait and can access the last committed records. Other transactions can modify, insert, or delete data between executions of individual SELECT statements within the current transaction, resulting in non-repeatable reads or phantom rows.

REPEATABLE READ is a more restrictive isolation level than READ COMMITTED. It encompasses READ COMMITTED and additionally specifies that no other transactions can modify or delete data that has been read by the current transaction until the current transaction commits. Concurrency is lower than for READ COMMITTED because shared locks on read data are held for the duration of the transaction instead of being released at the end of each statement. But other transactions can insert data between executions of individual SELECT statements within the current transaction, resulting in phantom rows.

The highest isolation level, serializable, guarantees that a transaction will retrieve exactly the same data every time it repeats a read operation, but it does this by performing a level of locking that is likely to impact other users in multi-user systems.

SNAPSHOT isolation specifies that data read within a transaction will never reflect changes made by other simultaneous transactions. The transaction uses the data row versions that exist when the transaction begins. No locks are placed on the data when it is read, so SNAPSHOT transactions do not block other transactions from writing data. Transactions that write data do not block snapshot transactions from reading data. If no waiting is acceptable for the SELECT operation but the last committed data is enough to be displayed, this isolation level may be appropriate.