**Write a short essay talking about your understanding of transactions, locks and isolation levels.**

Transaction is basically a sequence of operations that change data stored in database. It is treated as one single unit. Therefore, the transaction basically ensures that either all of the command are executed successfully or none of them is. It has four properties to control and maintain accurate and consistent which includes atomicity, consistency, isolation and durability. Atomicity means that the entire transaction takes place at once or doesn’t happen at all. Consistency means that the database must be consistent before and after the transaction. Isolation means multiple transactions happen independently without interfering with one another. Durability means once a transaction has happened, the changes stay there even if a system failure occurs afterwards. And there only two possible outcomes of it, commit and rollback. The former is used to save the changes and the latter is used to undo the changes before saving. But we can use savepoint to rollback to the savepoint instead of the start of the transaction.

Locks are used to enforce the isolation property of transactions to do the pessimistic concurrency control. Shared lock is used for reading data which ensures that a record is not in process of being updated during a read-only request and prevents any kind of updates of record. Exclusive locks prevent any other locker from obtaining any sort of a lock on the object. Update lock is used to avoid deadlock. Locks in a situation of multiple transaction may cause deadlocks when a transaction is trying to modify data that is being modified by another transaction.

Isolation levels determine the level of consistency user gets when user interacts with data. Isolation level can be defined based on locking model or row versioning. As for base on locks, SQL Server supports four isolation levels that are based on pure locking model: read uncommitted, read committed, repeatable read and serializable. Read uncommitted is the lowest available isolation level. In this level, a writer can change data while a reader that is running under the READ UNCOMMITTED isolation level reads data. The reading in this situation also called dirty read. The lowest isolation level that prevents dirty reads is READ COMMITTED, which is also the default isolation level in SQL Server. It allows readers to read only committed changes and prevents uncommitted reads by requiring a reader to obtain a shared lock. In REPEATABLE READ level, not only does a reader need a shared lock to be able to read, but it also holds the lock until the end of the transaction. As soon as the reader acquires a shared lock on a data resource to read it, no one can obtain an exclusive lock to modify that resource until the reader ends the transaction. But it locks only resources that query found the first time it ran, not rows that weren’t there when the query ran. Therefore, a second read in the same transaction might return new rows as well. Those new rows are called phantom. Phantom read happens if, in between the reads, another transaction inserts new rows that satisfy the reader’s query filter. To prevent phantom read, we have SERIALIZABLE level. It causes a reader to lock the whole range of keys that qualify for the query’s filter. So the reader locks not only the existing rows that qualify for the query’s filter, but also future ones.