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

Generally, a transaction represents any change within a DBMS against a database. The changes can consist of a single read, write, delete, or update operations or a combination of those. ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee the database transactions are processed reliably. Atomicity means all operations in a transaction succeed or every operation is rolled back. Consistency stands for the database is structurally sound on the completion of a transaction. Isolation implies that transactions do not contend with one another. Durability exemplifies the results of applying a transaction are permanent, even in the presence of failures.

SQL Server can operate three different transactions modes: Autocommit Transaction mode, Implicit transaction mode, and Explicit transaction mode. Autocommit transaction is a default mode, and each T-SQL statement is evaluated as a transaction, and they are committed or rolled back according to their results in this mode. If we run a query without mentioning the BEGIN TRAN, it will be considered an implicit transition. On the other hand, if we run a query that starts with BEGIN TRAN and ends with COMMIT or ROLLBACK, it will be considered an explicit transaction. We can also name a transaction and save it use the SAVE TRANSACTION syntax. It can be a save point, and we can use it to roll back any particular part of the transaction rather than the entire transaction.

Concurrency is a situation that arises in a database due to the transaction process. DBMS concurrency is a problem because accessing data simultaneously by two different users can lead to inconsistent results or invalid behavior. SQL Server provides five different levels of transaction isolation to overcome these Concurrency problems. These five isolation levels work on two major concurrency models: the pessimistic model and the optimistic model. In the pessimistic model of managing concurrent data access, the readers can block writers, and the writers can block readers. On the other hand, in the optimistic model of controlling concurrent data access, the readers cannot block writers, and the writers cannot block readers, but the writer can block another writer.

Pessimistic concurrency control utilizes a system of locks to prevent users from modifying data in a way that affects other users. When the database engine processes a statement, the query processor decides which resources need to be accessed and how they will be used. Based on this information, it then determines what types of locks are required to protect each resource. The acquired locks also depend upon the transaction isolation level setting. In short, locks are there to protect resources. Lock modes include Shared, Update, Exclusive, etc.

Shared locksallow concurrent transactions to read a resource, but no other transaction can modify the resource while the lock is held. When transactions might need to update the resource, we use Update locks. Only one transaction can gain an update lock at a time. If data is to be modified, then that lock is converted to an exclusive lock. To ensure that only one transaction can update data at one time, we use Exclusive locks. And nothing else can access that data unless the read uncommitted isolation level is set.

When we connect to a SQL server database, the application can submit queries with one of five different isolation levels. Read Uncommitted, Read Committed, Repeatable Read, and Serializable come under the pessimistic concurrency model. Snapshot comes under the optimistic concurrency model.

Read Uncommitted is the first level of isolation. One transaction is allowed to read the data that is about to be changed by the commit of another process. Read Committed is the second level of isolation and it is the default level. In this level, if you are reading data then the concurrent transactions that can delete or write data, some work is blocked until the other work is complete. Repeatable Read is the third level of isolation. At this level, the transaction has to wait till another transaction's update or read query is complete. But if there is an insert transaction, it does not wait for anyone. Serializable is the highest level of isolation. In this level of isolation, we can ask any transaction to wait until the current transaction completes.

Snapshot takes a snapshot of the current data and uses it as a copy for the different transactions. Each transaction has its copy of data here. If a user tries to perform a transaction like an update or insert, Snapshot will ask him to re-verify all the operations before the process gets started executing.

29. Write a short essay, plus screenshots talking about performance tuning in SQL Server. Must include Tuning Advisor, Extended Events, DMV, Logs and Execution Plan.

30. Write a short essay talking about a scenario: Good news everyone! We (Wide World Importers) just brought out a small company called “Adventure works”! Now that bike shop is our sub-company. The first thing of all works pending would be to merge the user logon information, person information (including emails, phone numbers) and products (of course, add category, colors) to WWI database. Include screenshot, mapping and query.

31. Database Design: OLTP db design request for EMS business: when people call 911 for medical emergency, 911 will dispatch UNITs to the given address. A UNIT means a crew on an apparatus (Fire Engine, Ambulance, Medic Ambulance, Helicopter, EMS supervisor). A crew member would have a medical level (EMR, EMT, A-EMT, Medic). All the treatments provided on scene are free. If the patient needs to be transported, that’s where the bill comes in. A bill consists of Units dispatched (Fire Engine and EMS Supervisor are free), crew members provided care (EMRs and EMTs are free), Transported miles from the scene to the hospital (Helicopters have a much higher rate, as you can image) and tax (Tax rate is 6%). Bill should be sent to the patient insurance company first. If there is a deductible, we send the unpaid bill to the patient only. Don’t forget about patient information, medical nature and bill paying status.