**Q28**

Write a short essay (400-750 words) talking about your understanding of transactions, locks and isolation levels.

A transaction is a logical unit of work that performs a single activity or multiple activities on the database. A transaction can consist of a single read, write, delete, or update operation or a combination of these operations. Suppose when we want to withdraw money from ATM, the ATM application will implement this operation in three steps. In the first step, the application will verify the account balance and then deduct the money from the source account. With these two processes, you will keep a record of this withdrawal activity. The following figure basically illustrates the operating principle of transactions in a relational database system.

The object of the transaction is that when each statement returns an error, the entire modification is rolled back to provide data integrity. On the other hand, if all statements are successfully completed, the data modification will persist on the database. Therefore, if we encounter any power outages or other problems during withdrawal from ATM, the transaction can guarantee the consistency of our balance. This would be the best way to perform all these steps through a transaction, because the four main attributes of a transaction make all operations more accurate and consistent. All these attributes are called ACID (atomicity, consistency, isolation, durability) in relational database systems, and are represented by the initials of their names.

Atomicity means all operations included in a successfully executed transaction. Otherwise, cancel all operations at the point of failure and roll back all previous operations. Consistency conduct with attribute ensures that all data will remain consistent after the transaction is completed according to defined rules, constraints, cascades, and triggers. Isolation indicates transactions are isolated from other transactions. Persistence indicates modifications of committed transactions are persisted in the database

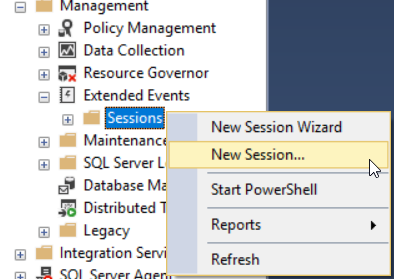
The transaction must run at an isolation level of at least repeatable reads to prevent loss of updates that may occur when two transactions each retrieve the same row, and then update the row based on the originally retrieved value. If two transactions use a single UPDATE statement to update rows and do not update based on previously retrieved values, no missing updates will occur at the default isolation level of committed reads.

Locking is critical to successful SQL Server transaction processing and is designed to allow SQL Server to run smoothly in a multi-user environment. Locking is how SQL Server manages transaction concurrency. Basically, a lock is a structure in memory with the owner, type, and hash of the resource to be protected. The size of the lock-like structure in the memory is 96 bytes.

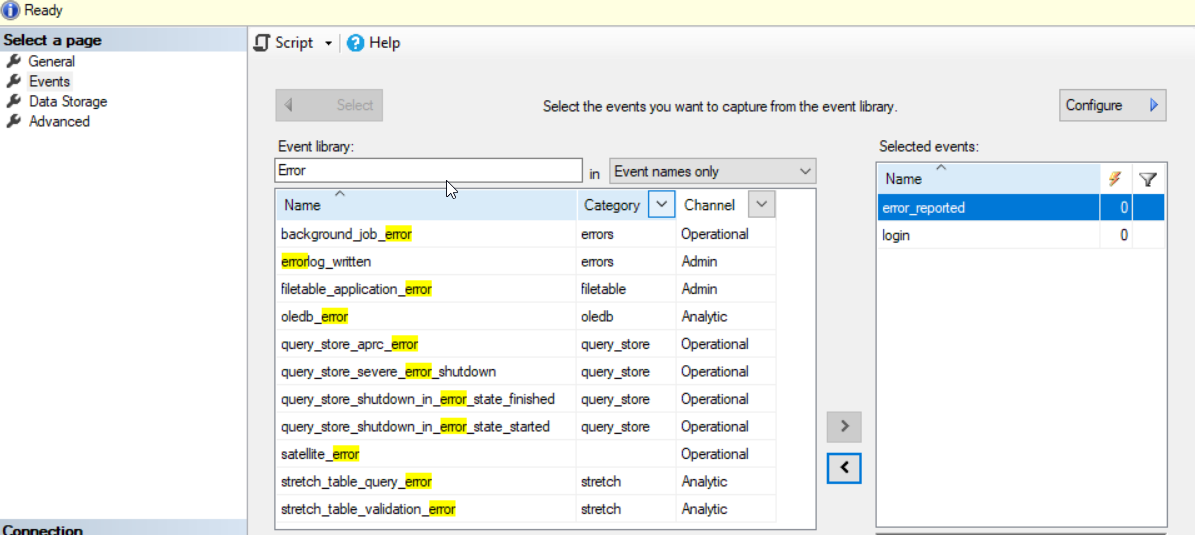
**Q29**

Write a short essay (300+ words, plus screenshots) talking about a scenario: It is Monday morning and your boss just told you that “he got feedbacks from a random average user that the ‘website’ was very slow and even threw db related errors during the weekend”. What should you do?

I would like to check the performance monitor such as extended events to categorize wait time type by events to see what’s reason that our query was very slow and threw error. To explore whether the error occur from deadlock or blocking, I would like to use Extend events, since SQL Server Extended Events are a highly scalable and configurable events framework that helps in collecting as much useful information as possible from the wide range of available actions, with the least possible SQL Server resources consumption, for troubleshooting and performance tuning purposes. SQL Server Extended Events are a highly scalable and configurable events framework that helps in collecting as much useful information as possible from the wide range of available actions, with the least possible SQL Server resources consumption, for troubleshooting and performance tuning purposes. Firstly, I would create a SQL Server Extended Events session that audits both the succeeded and failed login processes. To do that, expand the Extended Events option under the Management node, right-click on the Sessions option and choose New Session.



on the Events tab, I will customize session and choose the events that I are managed to monitor. I will choose the Login event to track the successful login processes and the Error Reported event to collect the failed logins as follows:



After performing successful and failed login processes, the events will be collected and displayed by the SQL Server Extended Events session. the successful login process properties, including the user name, the host name, the application used for the login and other useful information will be displayed as shown picture

