

28.

SQL transaction is a single unit of work, aimed to solve concurrently-executing issues. Transaction started with the BEGIN TRANSACTION statement or when the prior transaction completes. Transaction has two outcomes: commit or rollback. All transactions explicitly ended with a COMMIT or ROLLBACK statement.

Every SQL Server transaction to pass the ACID test, Atomicity ,Consistency, Isolation, and Durability.

Atomicity and consistency are related to each other. A transaction cannot be divided, all operations in the transaction are either all done or not done. A transaction must create a valid state of new data that cannot violate the integrity of the database, otherwise the transaction needs to be rolled back.

Isolation means when a transaction that is still running, no other transactions take place and affect the running transaction.

Durability means committed data must be stored using a method that will preserve all data in the correct state and available to a user, even System failures or restarts do not affect committed transactions.

Lock is the essential part of the isolation requirement and is designed to ensure the integrity of the data making it impossible for two or more users to update the row at the same time.

There are three main types of lock, Shared, Update, and Exclusive Lock:

Exclusive Lock: Other transactions cannot read and modify that data

Shared lock: Allows concurrent transactions to read the same data, but it prevents modification.

Update lock: places a Shared lock on a resource that already has another shared lock on it. When the transaction is ready to make its changes, the update lock converts to an exclusive lock.







Transaction isolation levels control what type of locks are requested and how long the read locks are held. A lower isolation level increases the ability of many users to access data at the same time. It also increases the number of concurrency effects SQL server provides 5 types of isolation levels. From low level to high level are: Read uncommitted, Read committed, Repeatable read, Snapshot, Serializable.

29.

Performance tuning in SQL Server is to make the SQL statements run in the fastest possible time. It will identify which queries are experiencing slowdowns and optimizing them for maximum efficiency

Database Engine Tuning Advisor is a tool for creating, modifying, and deleting partitions and indexes. It will recommend how users can improve query processing performance by modifying database structures such as indexes, indexed views, and partitioning.

Extended Events is a performance monitoring feature in the SSMS. It allows creating sessions to monitor and collect different events, such as `sql_transaction` and `cursor execute`.

Connect      

DESKTOP-FPRLB3 (SQL Server 15.0)

- Databases
 - System Databases
 - Database Snapshots
 - AdventureWorks2019
 - WideWorldImporters
 - WideWorldImportersDW
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
 - Policy Management
 - Data Collection
 - Resource Governor
 - Extended Events
 - Sessions
 - AlwaysOn_health
 - ses
 - package0.event_file
 - system_health
 - telemetry_events
 - Maintenance Plans
 - SQL Server Logs
 - Database Mail
 - Distributed Transaction Coordinator
 - Integration Services Catalogs
 - SQL Server Agent (Agent XPs disabled)
 - XEvent Profiler

Displaying 102720 Events

| | name | timestamp |
|---|-----------------|-----------------------------|
| ▶ | sql_transaction | 2021-09-21 11:45:54.8621372 |
| | sql_transaction | 2021-09-21 11:45:54.8622437 |
| | sql_transaction | 2021-09-21 11:45:54.8622760 |
| | sql_transaction | 2021-09-21 11:45:54.8623369 |
| | sql_transaction | 2021-09-21 11:45:59.8754164 |
| | sql_transaction | 2021-09-21 11:45:59.8756492 |
| | sql_transaction | 2021-09-21 11:45:59.8756973 |
| | sql_transaction | 2021-09-21 11:45:59.8758319 |
| | sql_transaction | 2021-09-21 11:46:01.9828369 |
| | sql_transaction | 2021-09-21 11:46:01.9828713 |
| | sql_transaction | 2021-09-21 11:46:01.9829537 |
| | sql_transaction | 2021-09-21 11:46:01.9829713 |
| | sql_transaction | 2021-09-21 11:46:04.3571379 |
| | sql_transaction | 2021-09-21 11:46:04.3571496 |

Event: sql_transaction (2021-09-21 11:45:54.8621372)

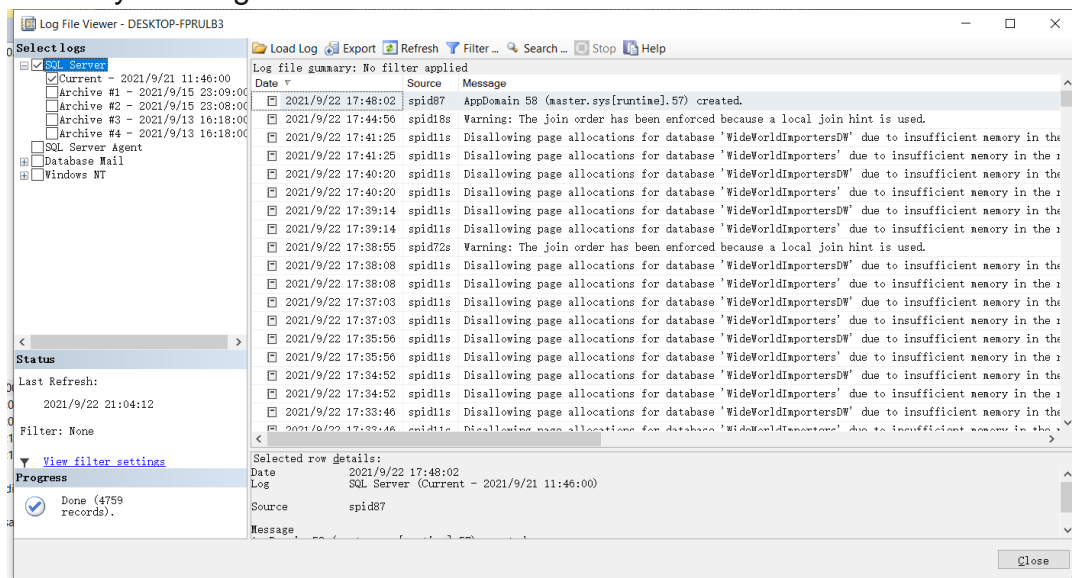
Details

| Field | Value |
|--------------|------------------|
| duration | 0 |
| last_error | 0 |
| object_name | GarbageCollector |
| savepoint... | |
| session_id | 64 |
| transacti... | 968205 |
| transacti... | Begin |
| transacti... | System |

DMV represents “Dynamic Management Views”, it can be implemented by using the query like “select * from sys.dm_db_log_stats(1)” to see the system views and system functions that return metadata of the system state.

| | database_id | recovery_model | log_min_lsn | log_end_lsn | current_vlf_sequence_number | current_vlf_size_mb | total_vlf_count | total_log_size_mb | active_vlf_count | active_log_size_mb |
|---|-------------|----------------|------------------------|------------------------|-----------------------------|---------------------|-----------------|-------------------|------------------|--------------------|
| 1 | 1 | SIMPLE | 000000f2:00000088:0001 | 000000f2:000000c0:0001 | 242 | 0.242187 | 8 | 1.992187 | 1 | 0.085937 |

SQL Server error log contains user-defined events and certain system events used for troubleshooting. It can be viewed under the Log File Viewer. Log File Viewer contains helpful info to detect any current or potential problem areas, including automatic recovery messages



Execution plan is a simple graphical representation of the operations that the query optimizer generates to calculate the most efficient way to return a set of results. It will tell various steps involved in fetching results from the database tables, and also display the cost in percentage of each step, such as hash match, clustered index scan and nested loop.

