

SQL Server Replication

Interview Questions and Answers

Introduction: This chapter takes you through the SQL Server Replication related interview questions and answers. These questions are helpful for range of database administrators starting from a junior to an expert level DBA while preparing for a technical interview. These questions and answers are deals with the below topics:

- SQL Server Replication
- Replication Scenarios

SQL Server Replication

1. What are the new features added in SQL Server 2008 Replication?

Ans:

- More improvements in Peer-To-Peer Transactional Replication:
 - ✓ The ability to detect conflicts during synchronization
 - ✓ The ability to add nodes to a replication topology without quiescing\disabling the topology
- Improved Replication Monitor:
 - ✓ On Monitor grid now we can apply “Selection,” “Sort” and “Filter” on grid columns
 - ✓ Common Jobs Tab on Publisher node has been renamed as Agents and it showcases all agents and

- jobs associated with the selected publisher
- 2. What are the new features added in SQL Server 2008 R2 Replication?

Ans:

In 2008 R2 there are no changes in Replication features.

- 3. What are the new features added in SQL Server 2012 Replication?

Ans:

- Replication supports ALWAYSON Availability Groups
- There are 4 new system stored procedures introduced for replication support on Availability groups
- Replication supports extended events
- Replication supports up to 15000 partitions for tables and

indexes

4. What are the limitations – Replication on Availability Groups on SQL Server 2012?

Ans:

- Publication database can be part of an availability group. The publication instances must share the common distributor.
- Supports Transaction, Merge and snapshot replication
- ALWAYSON Secondary can't be a publisher
- Republishing is not supported when replication is combined with ALWAYSON.
- Doesn't supports Peer-To-Peer (P2P), bi-directional, reciprocal transactional publications, and Oracle Publishing.

- A database that is enabled for Change Data Capture (CDC) and Change Tracking (CT) can be part of an availability group.
5. What are the new features added in SQL Server 2014 Replication?

Ans:

- There are no significant features added in SQL Server 2014 Replication.
 - SQL Server 2014 and above does not support replication to or from SQL Server 2005 or SQL Server Compact.
6. What are the new features added in SQL Server 2016 Replication?

Ans:

- Replication supports memory-optimized tables.
- Replication is now supported to Azure SQL Database.

7. Can you define replication?

Ans:

Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency. Using replication, you can distribute data to different locations and to remote or mobile users over local and wide area networks, dial-up connections, wireless connections, and the Internet.

8. What are the areas where the replication can be useful?

Ans:

Load balancing: Replication allows you to disseminate your

data to a number of servers and then distribute the query load among those servers.

Offline processing: You may wish to manipulate data from your database on a machine that is not always connected to the network.

Redundancy: Replication allows you to build a fail-over database server which is ready to pick up the processing load at a moment's notice.

Replicating data in a server to server environment:

- Improving scalability and availability
- Data warehousing and reporting
- Integrating data from multiple sites
- Integrating heterogeneous data
- Offloading batch processing

Replicating data between a server and clients:

- Exchanging data with mobile users
- Retail point of sale (POS) applications
- Integrating data from multiple sites

9. What are the various components involved in replication?

Ans:

Publisher:

The Publisher is a database instance that makes data available to other locations through replication. The Publisher can have one or more publications, each defining a logically related set of objects and data to replicate.

Distributor:

- The “Distributor” is a database instance that acts as a store

- The “Distributor” is a database instance that acts as a store for replication specific data associated with one or more Publishers.
- Each Publisher is associated with a single database (known as a distribution database) at the Distributor.
- The distribution database stores replication status data, metadata about the publication and in some cases acts as a queue for data moving from the Publisher to the Subscribers.
- **Local Distributer:** A single database server instance acts as both the Publisher and the Distributor.
- **Remote Distributer:** Publisher and the Distributor are configured on separate database server instances

Subscribers:

- A Subscriber is a database instance that receives replicated data.
- A Subscriber can receive data from multiple Publishers and publications.
- Depending on the type of replication chosen, the Subscriber can also pass data changes back to the Publisher or republish the data to other Subscribers.

Article:

- An article identifies a database object that is included in a publication.
- A publication can contain different types of articles, including tables, views, stored procedures, and other objects.
- When tables are published as articles, filters can be used to

restrict the columns and rows of the data sent to Subscribers.

Publication:

A publication is a collection of one or more articles from one database. The grouping of multiple articles into a publication makes it easier to specify a logically related set of database objects and data that are replicated as a unit.

Subscription:

A subscription is a request for a copy of a publication to be delivered to a Subscriber. The subscription defines what publication will be received, where, and when. There are two types of subscriptions: push and pull.

- **Push:** With a push subscription, the Publisher propagates changes to a Subscriber without a request from the

Subscriber. Changes can be pushed to Subscribers on demand, continuously, or on a scheduled basis. The Distribution Agent or Merge Agent runs at the Distributor.

- **Pull:** With a pull subscription, the Subscriber requests changes made at the Publisher. Pull subscriptions allow the user at the Subscriber to determine when the data changes are synchronized. The Distribution Agent or the Merge Agent runs at the Subscriber.

10. What are the agents involved in replication?

Ans:

Replication uses a number of standalone programs, called agents, to carry out the tasks associated with tracking changes and distributing data.

SQL Server Agent:

SQL Server Agent hosts and schedules the agents used in

replication and provides an easy way to run replication agents.

Snapshot Agent:

- The Snapshot Agent is typically used with all types of replication.
- It prepares schema and initial data files of published tables and other objects, stores the snapshot files, and records information about synchronization in the distribution database.
- The Snapshot Agent runs at the Distributor.

Log Reader Agent:

- The Log Reader Agent is used with transactional replication.
- It moves transactions marked for replication from the

transaction log on the Publisher to the distribution database.

- Each database published using transactional replication has its own Log Reader Agent that runs on the Distributor and connects to the Publisher

Distribution Agent:

- The Distribution Agent is used with snapshot replication and transactional replication.
- It applies the initial snapshot to the Subscriber and moves transactions held in the distribution database to Subscribers.
- The Distribution Agent runs at either the Distributor for push subscriptions or at the Subscriber for pull subscriptions.

- The Queue Reader Agent is used with transactional replication with the queued updating option.
- The agent runs at the Distributor and moves changes made at the Subscriber back to the Publisher.
- Unlike the Distribution Agent and the Merge Agent, only one instance of the Queue Reader Agent exists to service all Publishers and publications for a given distribution database.

11. How to monitor latency in replication?

Ans:

There are three methods.

- Replication monitor
- Replication commands
- Tracer Tokens

Replication Monitor: In replication monitor from the list of all subscriptions just double click on the desired subscription. There we find three tabs.

- Publisher to Distributor History
- Distributor to Subscriber History
- Undistributed commands

Replication Commands:

- **Publisher.SP_ReplTran:** Checks the pending transactions at p
- **Distributor.MSReplCommands and MSReplTransactions:** Gives the transactions and commands details. Actual T_SQL data is in binary format. From the entry time we can estimate the latency.

- **Distributor.SP_BrowseReplCmds:** It shows the exact_seqno along with the corresponding T-SQL command
- **sp_replmonitorsubscriptionpendingcmds:** It shows the total number of pending commands to be applied at subscriber along with the estimated time.

Tracer Tokens:

Available from Replication Monitor or via TSQL statements, Tracer Tokens are special timestamp transactions written to the Publisher's Transaction Log and picked up by the Log Reader. They are then read by the Distribution Agent and written to the Subscriber. Timestamps for each step are recorded in tracking tables in the Distribution Database and can be displayed in Replication Monitor or via TSQL statements.

When Log Reader picks up token it records time in MStracer_tokens table in the Distribution database. The Distribution Agent then picks up the Token and records Subscriber(s) write time in the MStracer_history tables also in the Distribution database.

Below is the T-SQL code to use Tracer tokens to troubleshoot the latency issues.

--A SQL Agent JOB to insert a new Tracer Token in the publication database.

```
USE [AdventureWorks]
```

```
Go
```

```
EXEC      sys.sp_posttracertoken      @publication      =
<PublicationName>
```

```
Go
```

--Token Tracking Tables

USE Distribution

Go

--publisher_commit

```
SELECT Top 20 * FROM MStracer_tokens Order by tracer_id desc
```

--subscriber_commit

```
SELECT Top 20 * FROM MStracer_history Order by parent_tracer_id desc
```

12. Can you quickly explain how to configure replication in a generic way?

Ans:

- Identify and configure Distributer
- Identify and configure Publisher
- **Publish Data and Database Objects:**

- ✓ Creating a publication and defining the data and database objects in the publication, setting options, and applying filters, if necessary.

➤ **Subscribing to Publications:**

- ✓ Creating PUSH and PULL subscriptions; Specify synchronization schedules and set other options.

➤ **Initializing Subscription:** Describes how to initialize the Subscriber

➤ **Synchronizing Data:** Specify options for synchronization, which occurs when the Distribution Agent or Merge Agent runs and updates are propagated between the Publisher and Subscribers

13. What are the different types in replication implementation?

Ans:

There are mainly 3 types of methodologies available in replication

- Snapshot Replication
- Merge Replication
- Transactional Replication

14. Can you brief about how Snapshot Replication works?

Ans:

The publisher simply takes a snapshot of the entire replicated database and shares it with the subscribers. There are two scenarios where snapshot replication is commonly used.

- It is used for databases that rarely change.
- It is used to set a baseline to establish replication between systems while future updates are propagated using

transactional or merge replication.

15. Can you explain how Transactional Replication works?

Ans:

Transactional replication is implemented by the SQL Server Snapshot Agent, Log Reader Agent, and Distribution Agent.

Snapshot Agent: Prepares snapshot files containing schema and data of published tables and database objects, stores the files in the snapshot folder, and records synchronization jobs in the distribution database on the Distributor.

Log Reader Agent: Monitors the transaction log of each database configured for transactional replication and copies the transactions marked for replication from the transaction log into the distribution database, which acts as a reliable store-and-forward queue.

Distribution Agent: Copies the initial snapshot files from the

snapshot folder and the transactions held in the distribution database tables to Subscribers. Incremental changes made at the Publisher flow to Subscribers according to the schedule of the Distribution Agent, which can run continuously for minimal latency, or at scheduled intervals.

16. Can you explain how Merge Replication works?

Ans:

- Merge Replication allows the publisher and subscriber to independently make changes to the database.
- Both entities can work without an active network connection.
- When they are reconnected, the merge replication agent checks for changes on both sets of data and modifies each database accordingly.

- If changes conflict with each other, it uses a predefined conflict resolution algorithm to determine the appropriate data.
- Merge replication is commonly used by laptop users and others who cannot be constantly connected to the publisher.
- Merge replication is implemented by the SQL Server Snapshot Agent and Merge Agent.
- ***Snapshot Agent:*** If the publication is unfiltered or uses static filters, the Snapshot Agent creates a single snapshot. If the publication uses parameterized filters, the Snapshot Agent creates a snapshot for each partition of data.
- ***Merge Agent:*** The Merge Agent applies the initial snapshots to the Subscribers. It also merges incremental data changes that occurred at the Publisher or Subscribers after the

initial snapshot was created, and detects and resolves any conflicts according to rules you configure.

17. Can you be able to explain what you know about Transactional Replication with Updatable Subscriptions?

Ans:

Transactional replication supports updates at Subscribers through updatable subscriptions and peer-to-peer replication. The following are the two types of updatable subscriptions:

Immediate updating: The Publisher and Subscriber must be connected to update data at the Subscriber. When data is updated at a Subscriber, it is first propagated to the Publisher and then propagated to other Subscribers. If immediate updating is used, the changes are propagated immediately using the two-phase commit protocol.

Queued updating: The Publisher and Subscriber do not have to

be connected to update data at the Subscriber. Updates can be made while the Subscriber or Publisher is offline. When data is updated at a Subscriber, it is first propagated to the Publisher and then propagated to other Subscribers. If queued updating is used, the changes are stored in a queue; the queued transactions are then applied asynchronously at the Publisher whenever network connectivity is available. Because the updates are propagated asynchronously to the Publisher, the same data may have been updated by the Publisher or by another Subscriber and conflicts can occur when applying the updates. Conflicts are detected and resolved according to a conflict resolution policy that is set when creating the publication.

18. In which replication methodology we can update at subscribers?

Ans:

- Merge replication
- Peer-to-peer transactional replication
- Transactional replication with updating subscriptions

19. Your customer given a business requirement and asked you to configure the replication topology. How do you suggest the correct replication topology?

Ans:

Before suggesting the type of replication first we need to understand these replication types and their usage.

Snapshot:

- Data changes infrequently.
- It is acceptable to have copies of data that are out of date

with respect to the Publisher for a period of time.

- Replicating small volumes of data.
- A large volume of changes occurs over a short period of time.

Transactional:

- Incremental changes to be propagated to Subscribers as they occur.
- The application requires low latency between the time changes are made at the Publisher and the changes arrive at the Subscriber.
- The Publisher has a very high volume of insert, update, and delete activity.
- The Publisher or Subscriber is a non-SQL Server database, such as Oracle.

Merge:

- Multiple Subscribers might update the same data at various times and propagate those changes to the Publisher and to other Subscribers.
 - Subscribers need to receive data, make changes offline, and later synchronize changes with the Publisher and other Subscribers.
 - Each Subscriber requires a different partition of data.
 - Conflicts might occur and, when they do, you need the ability to detect and resolve them.
20. Have you ever configured both log shipping and replication on same database? If yes can you tell me the things that we need to consider?

Ans:

No! I never implemented this scenario but we know the limitations. Replication does not continue after a log shipping failover. If a failover occurs, replication agents do not connect to the secondary, so transactions are not replicated to Subscribers. If a failback to the primary occurs, replication resumes. All transactions that log shipping copies from the secondary back to the primary are replicated to Subscribers.

21. Have you ever configured both Database Mirroring and replication on same database? If yes can you tell me the things that we need to consider?

Ans:

Yes! We have implemented both Mirroring and Replication on same database

- The principal and mirror must share a Distributor. Remote Distributor provides greater fault tolerance if the Publisher

has an unplanned failover.

- Replication supports mirroring the publication database for merge replication and for transactional replication with read-only Subscribers or queued updating Subscribers
- Immediate updating Subscribers, Oracle Publishers, Publishers in a peer-to-peer topology, and republishing are not supported.
- Metadata and objects that exist outside the database are not copied to the mirror, including logins, jobs, linked servers, and so on. If you require the metadata and objects at the mirror, you must copy them manually.

22. Can you explain how snapshot agent works?

Ans:

The Snapshot Agent performs the following steps:

- ***Locking:*** Establishes a connection from the Distributor to the Publisher, and then takes locks on published tables if necessary:
 - ✓ For merge publications, the Snapshot Agent does not take any locks.
 - ✓ For transactional publications, by default the Snapshot Agent takes locks only during the initial phase of snapshot generation.
 - ✓ For snapshot publications, locks are held during the entire snapshot generation process.
- ***.SCH file:*** Writes a copy of the table schema for each article to a .sch file.
- ***Additional Script Files:*** It generates additional script files if database objects are published, such as indexes, constraints, stored procedures, views, user-defined

functions etc.

- **BCP Files:** Copies the data from the published table at the Publisher and writes the data to the snapshot folder. The snapshot is generated as a set of bulk copy program (BCP) files.
- **MSrepl_commands and MSrepl_transactions:** For snapshot and transactional publications, the Snapshot Agent appends rows to these tables in the distribution database. The entries in the **MSrepl_commands** table are commands indicating the location of .sch and .bcp files, any other snapshot files, and references to any pre- or post-snapshot scripts.
- Releases any locks on published tables.
- During snapshot generation, you cannot make schema changes on published tables. After the snapshot files are

generated, you can view them in the snapshot folder using Windows Explorer.

23. Can you explain how Distribution Agent works?

Ans:

For snapshot and transactional replication, the Distribution Agent performs the following steps:

- Establishes a connection to the Distributor.
- Examines the **MSrepl_commands** and **MSrepl_transactions** tables in the distribution database on the Distributor. The agent reads the location of the snapshot files from the first table and Subscriber synchronization commands from both tables.
- Applies the schema and commands to the subscription database.

24. Can you explain how Merge Agent works?

Ans:

For an unfiltered merge replication publication, the Merge Agent performs the following steps:

- Establishes a connection to the Publisher.
- Examines the **sysmergeschemachange** table on the Publisher and determines whether there is a new snapshot that should be applied at the Subscriber.
- If a new snapshot is available, the Merge Agent applies to the subscription database the snapshot files from the location specified in **sysmergeschemachange**.

25. What is the useful system CATALOGS/tables related to Replication?

Ans:

On PUBLISHER DB:

- sysarticles
- sysarticleupdates
- syspublications
- syssubscriptions
- master..sysservers

On DISTRIBUTION DB:

- master..sysservers
- msarticles
- msdistribution_agents
- msdistribution_history
- mslogreader_agents

- mslogreader_history
- mspublication_access
- mspublications
- mspublisher_databases
- msrepl_backup_lsns
- msrepl_commands
- msrepl_errors
- msrepl_identity_range
- msrepl_transactions
- mssnapshot_agents
- mssnapshot_history
- mssubscriber_info
- mssubscriber_schedule
- msubscriptions

- mssync_states

On SUBSCRIBER DB:

- msreplication_subscriptions
- mssqlsubscription_agents
- mssqlsubscription_properties

DMV:

sys.dm_repl_articles
sys.dm_repl_schemas
sys.dm_repl_tranhash
sys.dm_repl_traninfo

26. What is the useful replication related system stored procedures available?

Ans:

`sp_replshowcmds:`

Using `sp_replshowcmds`, you can view transactions that currently are not distributed (those transactions remaining in the transaction log that have not been sent to the Distributor). This stored procedure is executed at the Publisher on the publication database.

`sp_repltrans:`

Returns a result set of all the transactions in the publication database transaction log that are marked for replication but have not been marked as distributed. This stored procedure is executed at the Publisher on a publication database.

`sp_replcmds:`

Returns the commands for transactions marked for replication. This stored procedure is executed at the Publisher on the publication database.

`sp_browsereplcmds:`

Returns a result set in a readable version of the replicated commands stored in the distribution database, and is used as a diagnostic tool. This stored procedure is executed at the Distributor on the distribution database.

Ex:

```
sp_browsereplcmds  
@xact_seqno_start='0x0004453600000F8C000B',  
@xact_seqno_end='0x0004453600000F8C000B',  
@publisher_database_id=4,  
@command_id=1 [Optional]
```

`sp_browsesnapshotfolder:`

Returns the complete path for the latest snapshot generated for a publication. This stored procedure is executed at the Publisher on the publication database.

27. What are the various ways to monitor replication?

Ans:

- Replication Monitor
- T-SQL Commands & system CATALOGS
- Tracer Tokens
- Performance Monitor
- Extended Events
- Verbose Logging

28. Which is the best way to monitor replication? Have you ever implemented automations in replication monitoring?

Ans:

Replication Monitor:

One of the best ways to monitor but not all the time

- It showcases the current situation to answer the question “How are things right Now?,” but it doesn’t baseline.
- Also replication monitor internally runs the commands to get the required details. On a busy topology it might be a performance intensive.

T-SQL Commands & system CATALOGS:

- We absolutely suggest this as it directly get only the required things
- It allows us to baseline and customizing the monitoring

Tracer Tokens:

- We don’t suggest this. We have seen some scenarios where it took lot of time to respond.

Performance Monitor:

It's also a resource intensive but sometimes it might be helpful. Even for the performance counters we would suggest to use T-SQL scripts with the required DMV.

Extended Events:

We never used this method. Moreover in SQL Server 2012 Books Online Microsoft mentioned that this method is for customer support engineers (Microsoft FTE) to collect the information for troubleshooting.

Verbose Logging:

This method may not useful for monitoring replication but it's really helpful at the time of troubleshooting. Using this method we just enable Agent History profile for the detailed logging and it helps us understanding the detailed progress and error message. Make sure we are using this only for the

troubleshooting as it's also a performance intensive

We suggest:

- Use a customized solution; create a list of tables to capture the replication agents, latency and pending transactions/commands details using T-SQL code.
- Define baselines for parameters
- Create scripts to capture the replication health details and store it on pre-designed tables
- Create scripts to compare the captured values with the baseline values and to send a health check report to the DBA team if there are any issues or delays

29. Can you tell me the performance counters which can help us in monitoring replication performance?

Ans:

SQLServer: Replication Agents: This group monitors the number of Log Reader and Snapshot agents currently running on the server.

SQLServer: Replication Dist: This group monitors Distribution agents' performance, including the following counters

- Dist: Delivered Cmds/Sec—Number of commands delivered to subscribers per second.
- Dist:Delivered Trans/Sec—Number of transactions delivered to subscribers per second
- Dist:Delivery Latency—Number of milliseconds it takes to deliver transactions from distributor to subscribers.

SQLServer: Replication Logreader: This group monitors Log Reader agents' performance.

- Logreader: Delivered Cmds/Sec—Number of commands delivered to subscribers per second.
- Logreader: Delivered Trans/Sec—Number of transactions delivered to subscribers per second
- Logreader: Delivery Latency—Number of milliseconds it takes to deliver transactions from publisher to distributor

SQLServer: Replication Snapshot: This group monitors the snapshot agent. Counters in this group include the following:

Snapshot: Delivered Cmds/Sec: Number of seconds delivered to the distributor

Snapshot: Delivered Trans/Sec: Number of transactions delivered to the distributor.

30. What are the things that we need to monitor periodically to know the replication health?

Ans:

There are basically three things that helps us to understand the health of replication:

- Replication Agent status
- Latency – Publisher → Distributer → Subscriber
- Outstanding commands to be subscribed

31. What recovery model is required on a replicated database?

Ans:

Replication functions properly using any of the recovery models: simple, bulk-logged, or full. Merge replication tracks change by storing information in metadata tables. Transactional replication tracks changes by marking the transaction log, but this marking process is not affected by the

recovery model.

32. How do I manage constraints on published tables?

Ans:

- Transactional replication requires a primary key constraint on each published table. Merge replication does not require a primary key, but if one is present, it must be replicated. Snapshot replication does not require a primary key.
- By default, primary key constraints, indexes, and check constraints are replicated to Subscribers.
- The NOT FOR REPLICATION option is specified by default for foreign key constraints and check constraints; the constraints are enforced for user operations but not agent operations.

33. How do I manage identity columns?

Ans:

Replication provides automatic identity range management for replication topologies that include updates at the Subscriber. If a row is inserted at the Publisher and the identity value is, for example, 21, that value is replicated to each Subscriber. When replication inserts data at each Subscriber, it does not increment the identity column value in the Subscriber table; instead, the literal value 21 is inserted. Only user inserts, but not replication agent inserts cause the identity column value to be incremented.

34. What are the ports required in replication?

Ans:

Replication connections to SQL Server use the typical regular Database Engine ports (TCP port 1433 for the default instance,

etc.)

Web synchronization and FTP/UNC access for replication snapshot require additional ports to be opened on the firewall. To transfer initial data and schema from one location to another, replication can use FTP (TCP port 21), or sync over HTTP (TCP port 80) or File Sharing. File sharing uses UDP port 137 and 138, and TCP port 139 if it is using NetBIOS. File Sharing uses TCP port 445.

35. Does replication affect the size of the transaction log?

Ans:

Merge replication and snapshot replication do not affect transaction log size, but transactional replication can. If a database includes one or more transactional publications, the log is not truncated until all transactions relevant to the publications have been delivered to the distribution database. If

the transaction log is growing too large, and the Log Reader Agent is running on a scheduled basis, consider shortening the interval between runs. Or, set it to run in continuous mode. If it is set to run in continuous mode (the default), ensure that it is running.

36. Have you ever heard the word “sync with backup” in transactional replication?

Ans:

Yes! This can be set on the distribution database and the publication database. When this option enabled transactions in the log will not be truncated until they have been backed up. The sync with backup option ensures consistency between the publication database and the distribution database, but the option does not guarantee against data loss. For example, if the transaction log is lost, transactions that have been committed

since the last transaction log backup will not be available in the publication database or the distribution database. This is the same behavior as a non-replicated database.

Replication Scenarios

37. Do you have any idea how to manually remove replication? If yes what is the sequence?

Ans:

You can manually remove a replication by using system stored procedures and other Transact- SQL statements. To completely remove a replication, follow these steps:

- Drop all subscriptions that are configured for the replication
- Drop all publications that are configured for the replication.

- Drop the distributor that is configured for the replication.
38. Scenario: Transactional replication configured on a database. A transaction failed on Publication. Does the failed transaction replicated to subscriber?

Ans:

No! Only committed transactions are replicated to subscribers. We can cross check using the system procedure “sp_browsereplcmds.”

39. Scenario: A transaction successfully completed on publisher but subsequently failed on subscriber. What happens to this transaction on subscriber and publisher?

Ans:

On publisher the transaction already got committed but it's failed to apply on subscriber. Transactions fails at subscriber

automatically rolled back. Until the problem is fixed, the error will prevent the distribution agent from processing any separate part of the transaction - the ACID properties are maintained. Note that this is the default behavior which may be overridden

40. Scenario: There is an update statement that affects 1 Million rows on the publisher which generates 1 Million records in transaction log. Due to this large transaction we are getting Log Reader Agent timeouts. How to handle these kinds of large transactions when replication configured? Or How to deal with the Log Reader Agent and distribution Agent Time Out Errors?

(This is must question if you are showcasing replication in your experience/profile)

Ans:

When there are large transactions executed on publisher database we might see timeout errors at Log Reader Agent or Distribution Agent.

RCA:

- When update happens for 1 million records there will be 1M update statements needs to be applied at subscriber
- The log reader agent will try to read each of these commands and transfer them to the distribution database. Timeout error can stop this process and the resulting messages may include:
 - ✓ “The process could not execute ‘sp_replcmds’ on ‘xxxxxx’.”
 - ✓ “Status: 2, code: 0, text: **‘Timeout expired’**”
 - ✓ “A time out occurred while waiting for memory

resources to execute the query.”

- ✓ “Agent ‘xxx’ is retrying after an error. 0 retries attempted. See agent job history in the Jobs folder for more details.”
- ✓ “The step was cancelled (stopped) as the result of a stop job request”
- These errors are each related to the number of records marked for replication in the log which the log reader has to parse and process. There are a few profile parameters which can be used to modify the process and avoid the error:

Resolution:

When we need to deal with the larger/huge transactions we need to play with various parameters to get rid of these timeout errors:

For Log Reader Agent Timeout:

- Increase “QueryTimeout”
- Decrease “MaxCmdsInTran”
- Decrease “ReadBatchSize”
- Decrease “ReadBatchThreshold”

For Distribution Agent Timeout:

- Increase “QueryTimeout”
- Decrease “CommitBatchSize”
- Decrease “CommitBatchThreshold”

Increase “QueryTimeout”: It won’t decrease the latency of a big transaction, but it is often sufficient on its own to get things working again.

Decrease “MaxCmdsInTran”:

- It indicates the number of commands in a transaction to be processed per batch.
- If the problem is that the transaction contains a huge amount of commands, then it could be an advantage to chop it up into several smaller transactions.
- Let's say 1 Million record update is applied in 10 transactions with each transaction having 1 lakh records.
- This will massively decrease latency and prevent associated timeouts.
- But there is a problem if one of the batches failed to update at subscriber the transaction doesn't follow ACID properties as the previous batches are already committed.

Decrease “ReadBatchSize”:

- This is the maximum number of transactions read out of the transaction log of the publishing database “per processing cycle.”
- The lowest this figure can prevent timeout errors when huge number of transactions causing the problem.
- But in this case it’s a single transaction with 1 M commands.

Decrease “ReadBatchThreshold”:

- This is very similar to the “ReadBatchSize” parameter but rather than relating to transactions, it relates to commands.
- Setting this to a small value along with increasing the QueryTimeout can help remove a backlog, especially if the problem is caused by large transactions.

Decrease “CommitBatchSize”:

- The value indicates the number of transactions to be issued to the Subscriber before a COMMIT statement is issued. The default is 100.

Decrease “CommitBatchThreshold”:

- The value indicates the number of replication commands to be issued to the Subscriber before a COMMIT statement is issued. The default is 1000.

41. Scenario: In transactional replication log reader agent is failing with the error message “The process could not execute ‘sp_repldone/sp_replcounters’ on ‘Publisher.’” Have you ever encountered this issue? If yes can you explain why this happens and how to resolve it?

Ans:

Yes! We had the same issue when we were working for one of the client.

RCA:

- We have checked Log Reader Agent output file for detailed error message:
The process could not execute
'sp_repldone/sp_replcounters' on 'MyPublisher'. Repl Agent Status: 5 Status: 2, code: 18752, text: '**Another log reader is replicating the database.**'
- From this error message we could conclude two things:
 - ✓ There is an orphan Log Reader Agent that is competing with the active log reader
 - ✓ When user manually executes sp_repldone,

- sp_replcmds, or sp_replshowcmds against the same database and forgot to close the connection
- ✓ Some resources on the published database are not available for the active Log Reader Agent.

Resolution:

- Check the replication monitor if any extra log reader agent is running against the given database. It mostly never showcase even an orphan log reader running
- Check the output of sp_who for specific process identification numbers (SPIDs) that are connected to the published database. Close any connections that might have run sp_repldone, sp_replcmds, or sp_replshowcmds.
- Restart the Log Reader Agent.
- Execute sp_replflush at the Publisher on the publication

database, and then restart the Log Reader Agent.

42. Scenario: The log reader agent fails with this error message: "Timeout expired" How to deal with this?

Ans:

- Quickly check the network connection between publisher and distributor (if it is a remote distributor). Also cross check the service/logins are working without any issue
- If you are able to connect to the publisher, run the following query under the published database:
`sp_replcmds 1
go
sp_replflush`
- The sp_replcmds is to Log Reader Agent will call to find out what transactions must be picked up and must be sent to

the distribution database.

- If the sp_replcmds stored procedure returns data, gradually increase the parameter value of the sp_replcmds Ex. 10, 50, 100 etc. Repeat this until the same timeout expired error occurs. We can know number of transactions that the Log Reader Agent can pick up.
- Reduce the value for the “ReadBatchSize” parameter from the default of 500 transactions to the working value you found through your sp_replcmds testing.
- Increase the “QueryTimeOut” parameter from 300 seconds to 3,000 seconds
- If no record returns when you run “sp_replcmds 1,” but you know that there are replication transactions that are waiting in the log then we must run DBCC CHECKDB

43. In replication setup how can we confirm that subscriber is in sync with Publisher without using GUI?

Ans:

- Check and make sure all replication agents are running without any issue
- On subscriber connect to subscriber database and run query.
`Select * from <subscriber database>.dbo.msreplication_subscriptions;`
- This table contains one row for each distribution agent. From the output get the “transaction_timestamp.” Ex: 0x000004570001E3FE000100000000
- Transaction_Timestamp represents the last xact_seqno/transaction got loaded in the subscription database. Ex.xact_seqno: 0x000004570001E3FE0001

(Excluded all zeros at the end)

- Connect to distribution database and run below query
Select * from Msrepl_commands where xact_seqno > 0x000004570001E3FE0001
- If there are no rows returned from the below query, indicates the replication is in sync.

44. Replication has broken as the subscriber was down. When it comes online we were trying to reinitialize the subscription. It was failed with the error message “Cannot Reinitialize the Subscriber.” Any idea how to fix this?

Ans:

RCA:

Subscription needs to be reinitialized if any snapshot needs to be applied for any broken replication. In this case Subscriptions

cannot be reinitialized for non-immediate_sync publications.

Resolution:

- Connect to the subscriber database, in the Msreplication_subscriptions table; the value of immediate_sync should be changed to 1. 0 indicates non-immediate sync publications

Then it allowed us to reinitialize the subscribers and we could generate snap shot from the publisher successfully.

45. In transactional replication setup Distribution agent failed with the error message “Cannot insert duplicate key row in object ‘XXXX’ with unique index ‘PK_XXXXXX’. The statement has been terminated..” Have you ever seen this kind of errors? If yes can you explain how to resolve?

Ans:

This problem occurs if when a row which is to be delivered on the subscriber already exists on the subscriber database. There are 3 ways that we can fix this:

- ***Skip the error:*** Failed transaction will be skipped and continue applying from the next transaction.
- ***Update @xact_seqno:*** To skip the specific transaction Update @xact_seqno with the help of sp_helpsubscriptionerrors and sp_setsubscriptionxactseqno
- ***Delete the row:*** Delete already existing row at subscriber

46. In transactional replication setup Distribution agent failed with the error message “The row was not found at the Subscriber when applying the replicated command.” Have you ever seen this kind of errors? If yes can you

explain how to resolve?

Ans:

This problem occurs if when a row which is to be deleted/updated on the subscriber on which the row is already deleted or not exists:

- ***Skip the error:*** Failed transaction will be skipped and continue applying from the next transaction.
- ***Update @xact_seqno:*** To skip the specific transaction Update @xact_seqno with the help of sp_helpsubscriptionerrors and sp_setsubscriptionxactseqno

47. How to skip Distributed Agent errors in transactional replication?

Ans:

- By default, when the Distribution Agent encounters an error, the agent stops.
- If you use the -SkipErrors parameter, and specify expected errors or errors that you do not want to interfere with replication, the agent will log the error information and then continue running.
- For example, if you want to specify that the Distribution Agent should log duplicate key violations but continue to process subsequent transactions, specify that the agent should skip errors 2601 (Cannot insert duplicate key row in object '%.*ls' with unique index '%.*ls'.) and 2627 (Violation of %ls constraint '%.*ls'. Cannot insert duplicate key in object '%.*ls'.):
 - -SkipErrors 2601:2627
 - The most common way to use the -SkipErrors parameter is

to use the Distribution Agent profile titled Continue on Data Consistency Errors.

- The Distribution Agent will then skip errors 2601, 2627, and 20598 (The row was not found at the Subscriber when applying the replicated command).
48. How to resolve the error “Could not execute sp_MSadd_repl_commands..” Here is the full error message: “The process could not execute ‘sp_MSadd_replcmds’ on ‘server\instance’. (Source: MSSQLServer, Error number: 1007) Cannot insert duplicate key row in object ‘dbo.MSrepl_commands’ with unique index ‘ucMSrepl_commands’. (Source: MSSQLServer, Error number: 1007)”

Ans:

RCA:

- If any other user is executing the repl commands in the publisher and his session is quite open for a long time will produce this error.
- **OR**
- One of the publications receives many inserts in one transaction.
- After the transaction is committed, the Log Reader Agent starts to process and split the transaction according to the MaxCmdsInTran parameter.
- The Snapshot Agent on another publication starts before the Log Reader Agent finishes the task.
- In this scenario, the Log Reader Agent fails, and you receive the following error message:

Resolution:

- If this issue occurs 2 or 3 times, we can ignore it and simply start the logreader jobs, it will succeed.
- If it's continuing then find out the session that's causing the problem, kill the SPID and restart the agent.
- If it's frequently occurring set parameters as below
 - ✓ @sync_method <> "concurrent"
 - ✓ @immediate_sync <> "True"
 - ✓ MaxCmdsInTran = 0
- Microsoft declared it's a known issue in SQL Server 2008 R2 SP1 and SQL Server 2012. To prevent these error messages apply the required Cumulative Update Packages released after SQL Server 2012 and SQL Server 2008 R2 SP1

49. Log reader fails with “The process could not execute

‘sp_replcmds’ ” error. How do you troubleshoot it?

Ans:

RCA:

- When we checked Snapshot Agent is also failing with the error message: “The concurrent snapshot for publication ‘PUB_XXXX is not available because it has not been fully generated or the Log Reader Agent is not running to activate it”
- We checked log reader agent history and it’s been failing with the error message: “The process could not execute ‘sp_replcmds’ on [ServerName]”
- To get the complete error details we added verbose log to the log reader agent and we have got the correct error details as: “Status: 0, code: 15517, text: ‘Cannot execute as the database principal because the principal “dbo” does not

exist, this type of principal cannot be impersonated, or you do not have permission.'."

- From the above error message we could identify that the current job owner is not having the DBO rights on publication database.

Resolution:

- We have given DBO rights to the job owner and then the replication agents started working without any issue.

50. How to enable replication agents for logging to output files in SQL Server?

OR

How to deal with the verbose logging in SQL Server replication?

Ans:

You can configure the SQL Server replication agents such as, the Snapshot Agent, Log Reader Agent, Queue Reader Agent, Distribution Agent, and Merge Agent to capture more information about error messages and procedure calls to a text file. To enable verbose logging you must add two parameter values to the Run Agent step of the replication agent job for which you wish to review more detailed ***logging***:

- -OutputVerboseLevel [0|1|2]
- -Output [Full Output Path and File Name]
- 0: No verbose logging
- 1: Minimal Logging (Default Logging)
- 2: Detailed Logging

Verbose logging can be enabled at agent profile properties as

below:

- In Replication Monitor “right-click” a Replication Agent and select “Agent Profile”
- Click on “Verbose history agent profile.”
- Click the details “...” button to review the Agent Profile settings.
- Update the value to “2” for the parameter “-HistoryVerboseLevel”
- We need to restart the Replication Agent for the setting to take effect.
- After restart if you check “View Details” we can see the detailed statistics

Note: We usually do not enable detailed logging as it may impact the performance. But this is very useful while

troubleshooting.

51. Have you ever encountered the error message "Subscriptions getting expired" and "subscriptions being marked as inactive"? What does it mean?

Ans:

- ***Subscription Expired:*** When subscriptions get expired, the subscription entries will get deleted at the publisher. So to get the replication back to sync, the subscription has to be recreated.
- ***Subscription Inactive:*** When subscriptions are marked as inactive, it means that the subscription can no longer receive the replicated commands and has to be re-initialized to get back to sync with the publisher.

Distribution database has a retention period of how many

commands it can store (the commands to be distributed to the subscribers). If a subscriber is not able to get the commands from the distribution database within this time, then as per the retention period, the commands are removed from the distribution database by an automatic cleanup job (distribution cleanup job). Since the sequence of transactions to be applied is lost, subscriber cannot get the next available commands and hence it will be marked as inactive. Thus by re-initializing, it will get a new snapshot and the subsequent commands will be applied from the distribution database after the snapshot is created.

52. We have configured transactional replication for one of our production database (3.4 TB) which is in SQL Server 2014. Replication is running successfully and now we have a requirement to add a new article. Can we be able to add a

new article without generating snapshot for all existing articles? I.e. The snapshot should be generated only for the newly added article. Is it possible? If yes can you explain how?

Ans:

Yes! It is possible. To generate snapshot only for the newly added articles we need to change two publication properties: “allow_anonymous,” “immediate_sync”.

Disable Publication Properties:

```
EXEC sp_changepublication  
    @publication=N'<Publication Name>',  
    ,@property=N'allow_anonymous'  
    ,@value='false';  
go  
exec sp_changepublication
```

```
@publication=N'<Publication Name>'  
, @property=N'immediate_sync'  
, @value='false';  
go
```

Add new Article: In SSMS from the publication properties add new article

Start the Snapshot Agent: right click the publication and select “View Snapshot Agent Status” and start the snapshot agent. If you check the snapshot folder you can see the snapshot generated only for the newly added article.

53. I heard that there is a problem in replication when SQL Server 2014 released. Is that true?

Ans:

Yes! Not with all topologies but yes there is a problem in dealing with snapshot replication in SQL Server 2014. But most of these

problems got resolved in the service pack. Errors we encountered in SQL Server 2014 snapshot:

Error 21820 “Cannot write to the script file in the snapshot folder at the Distributor (%ls).”:

- Make sure there is enough disk space available
- Crosscheck if the account under which the Snapshot Agent runs has permissions to write to the snapshot folder and its subdirectories.

Error 21331 “Unable to copy script file to the snapshot folder at the Distributor”:

- Make sure there is enough disk space available
- Crosscheck if the account under which the Snapshot Agent runs has permissions to write to the snapshot folder and its

subdirectories.

Error: 20690 “Cannot set up the Publisher identity range for table %s.”

- To resolve this problem, you should verify that appropriate ranges were specified when the article was created, and then rerun the Snapshot Agent.

Error: 20605 “Invalidated the existing snapshot of the publication.”

- To resolve this problem, we need to rerun the snapshot agent to generate a new snapshot.

Error: 14098 may occur when you try to drop Distribution Publisher

- This problem occurs when the remote Publisher is using as Distributor.
- To resolve this, you should disable publishing at the Publisher before attempting to drop this relationship.

Error: 14071 “Could not find the Distributor or the distribution database for the local server.”

- To resolve this, ensure that the Distributor is installed at the publisher instance

The Distribution Agent does not deliver commands to the snapshot subscriber even when Distribution Agent is running:

- This problem occurs when you specify the Distribution Agent to use multiple subscription streams by setting the -SubscriptionStreams parameter to a value of 2 or higher.

- To resolve this set the -SubscriptionStreams parameter to a value of 1.

The replication agent jobs may fail If SQL Server Agent STARTUP account changed to a domain account using the Windows Service Control Manager:

- This problem occurs because the Windows Service Control Manager does not grant the required permissions to the new domain account.
- To resolve this problem, you can use SQL Server Configuration Manager to change the STARTUP account to a domain account.

54. My publisher is in SQL Server 2014 and we need to subscribe to SQL Server 2012 instance. Is it possible?

Ans:

Yes! It is possible with some limitations:

- When subscribing to the previous version it is limited to the functionality of that version, both in terms of replication-specific functionality and the functionality of the product as a whole.
- Merge publications use a compatibility level, which determines what features can be used in a publication and allows you to support Subscribers running previous versions of SQL Server.

55. As per the business requirement one of the articles needs to be published in two publications. Is that possible?

Ans:

Yes, but with some restrictions:

- If an article is published in a transactional publication and a merge publication, ensure that the @published_in_tran_pub property is set to TRUE for the merge article.
- An article cannot be published in both a merge publication and a transactional publication with queued updating subscriptions.
- Articles included in transactional publications that support updating subscriptions cannot be republished.
- Etc.

56. Can multiple publications use the same distribution database?

Ans:

Yes. There are no restrictions on the number or types of publications that can use the same distribution database.

57. Does replication encrypt data?

Ans:

No. Replication does not encrypt data that is stored in the database or transferred over the network.

58. How can we replicate data over the Internet?

Ans:

- A Virtual Private Network (VPN).
- The Web synchronization option for merge replication

59. Does replication resume if a connection is dropped

Ans:

Yes. Replication processing resumes at the point at which it left off if a connection is dropped.

60. Does replication work over low bandwidth connections?

Does it use compression?

Ans:

Yes, replication does work over low bandwidth connections. For connections over TCP/IP, it uses the compression provided by the protocol but does not provide additional compression. For Web synchronization connections over HTTPS, it uses the compression provided by the protocol and also additional compression of the XML files used to replicate changes.

61. Are logins and passwords replicated automatically?

Ans:

No. You could create a SSIS package to transfer logins and passwords from a Publisher to one or more Subscribers.

62. Why can't I run TRUNCATE TABLE on a published table?

Ans:

TRUNCATE TABLE is a non-logged operation that does not fire

triggers. It is not permitted because replication cannot track the changes caused by the operation: transactional replication tracks changes through the transaction log; merge replication tracks changes through triggers on published tables. Use DELETE command without WHERE clause.

63. What is the effect of running a bulk insert command on a replicated database?

Ans:

For transactional replication, bulk inserts are tracked and replicated like other inserts. For merge replication, you must ensure that change tracking metadata is updated properly.

64. How can we rebuild indexes in replicated databases?

Ans:

There is no specific limitation on this. We can rebuild indexes as we do in a normal database except for the primary keys on

published articles. We can't drop and recreate primary keys on these tables.

65. How can we add or change indexes on publication and subscription databases?

Ans:

- Indexes can be added at the Publisher or Subscribers with no special considerations for replication.
- CREATE INDEX and ALTER INDEX are not replicated, so if you add or change an index at, for example, the Publisher, you must make the same addition or change at the Subscriber if you want it reflected there.

66. How do I drop a table that is being replicated?

Ans:

First drop the article from the publication using `sp_droparticle`,

`sp_dropmergearticle`, or the Publication Properties - <Publication> dialog box, and then drop it from the database using `DROP <Object>`. You cannot drop articles from snapshot or transactional publications after subscriptions have been added; you must drop the subscriptions first.

67. How can we add or drop columns on a published table?

Ans:

SQL Server supports a wide variety of schema changes on published objects, including adding and dropping columns. For example, execute `ALTER TABLE ... DROP COLUMN` at the Publisher, and the statement is replicated to Subscribers and then executed to drop the column.