**Replication Technology**

Replication is a set of technologies for copying and distributing data and database objects (**Tables i.e Articles, SP, Views & UDF etc**) from one database to another database. There will be three roles in replication i.e. Source DB is publisher and destination/Target DB is subscriber. In between there will be distributor DB this will communicate between publisher and subscriber.

**Distributor model design:**

You can have in publisher itself or on subscriber side or on dedicated distributor.

Two methods of subscription sending data – Push & Pull subscription

**When to use**

* Take out subset of data for reporting
* Remove PII or salary data in subscriber instead giving access to production
* To have more index for subscription for reporting
* To have a control on security by restricting access to user on subscriber
* You can run in simple recovery mode and can be used in different version & edition

**Types of Replication:**

**Snapshot replication** – Snapshot Agent will create snapshot of all tables from publisher and send or replicates to subscriber other database.

**When do you use it?**

**When to use SQL server Snapshot Replication**

SQL Server Snapshot replication is useful for the databases which are not critical and/or data records of the database do not change frequently. SQL Server Snapshot replication takes a snapshot until a specific time on the Publisher server and applies to the Subscriber server.

* Database records do not change frequently
* The database is not critical
* The database is for reporting purposes

To configure step by step Snapshot replication, I have set-up two server node SQL1 act as a Publisher server and SQL2 act as Subscriber server.

**Transactional replication -** Log Reader Agentwill create snapshot of all tables from publisher and send or replicates to subscriber other database.

**When do you use it?**

**When to use Transactional Replication:**

* Transactional replication used for the critical databases which require less downtime
* It is useful when where data need on an incremental basis
* It is useful for the database where a large amount of data changes frequently

Three agents used in Transactional replication. The Snapshot agent, The Log reader agent, and Distribution agent.

* Snapshot agent takes a snapshot of the published articles and put it into the snapshot folder
* The Log reader agent reads the transaction logs of the published database and transfers the committed transaction to the distribution database
* Distribution agent copied the snapshot data from the snapshot folder and the transaction log from the distribution database and applied to the subscriber server database

**How many agent or agent jobs created?**

It is based on the replication design and topology.

In general, snapshot agent is for each publication, Distributor agent is for each subscription and log reader agent is for each Published database.

All the agents are called from SQL agent job.

**Merge replication -** Merge Agent will create snapshot of all tables from publisher and send or replicates to subscriber other database.

Merge replication is used to replicate data from a publisher to a subscriber database and vice versa.

**When do you use it?**

Merge replication, like transactional replication, typically starts with a snapshot of the publication database objects and data. Subsequent data changes and schema modifications made at the Publisher and Subscribers are tracked with triggers. The Subscriber synchronizes with the Publisher when connected to the network and exchanges all rows that have changed between the Publisher and Subscriber since the last time synchronization occurred.

Merge replication is typically used in server-to-client environments. Merge replication is appropriate in any of the following situations:

* 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.
* The application requires net data change rather than access to intermediate data states. For example, if a row changes five times at a Subscriber before it synchronizes with a Publisher, the row will change only once at the Publisher to reflect the net data change (that is, the fifth value).