SQL Server replication: Overview of components and topography

September 11, 2018 by Prashanth Jayaram

The volume of data retained, managed, and accessed today is unprecedented. Businesses expect the IT department to keep data online and accessible indefinitely, putting intense pressure on the databases required to store and manage it. To meet today's needs; we need to replace outdated and inefficient legacy processes with new, more agile techniques. SQL Server Replication is one of the techniques to accommodate such demands.

In this article, let's you shape your understanding of the full SQL Server replication topography including components, internals and the SQL to bind it all together. After you complete reading this article, you'll understand:

- SQL Server replication, in general

- SQL Server Replication agents
- And more…

Replication

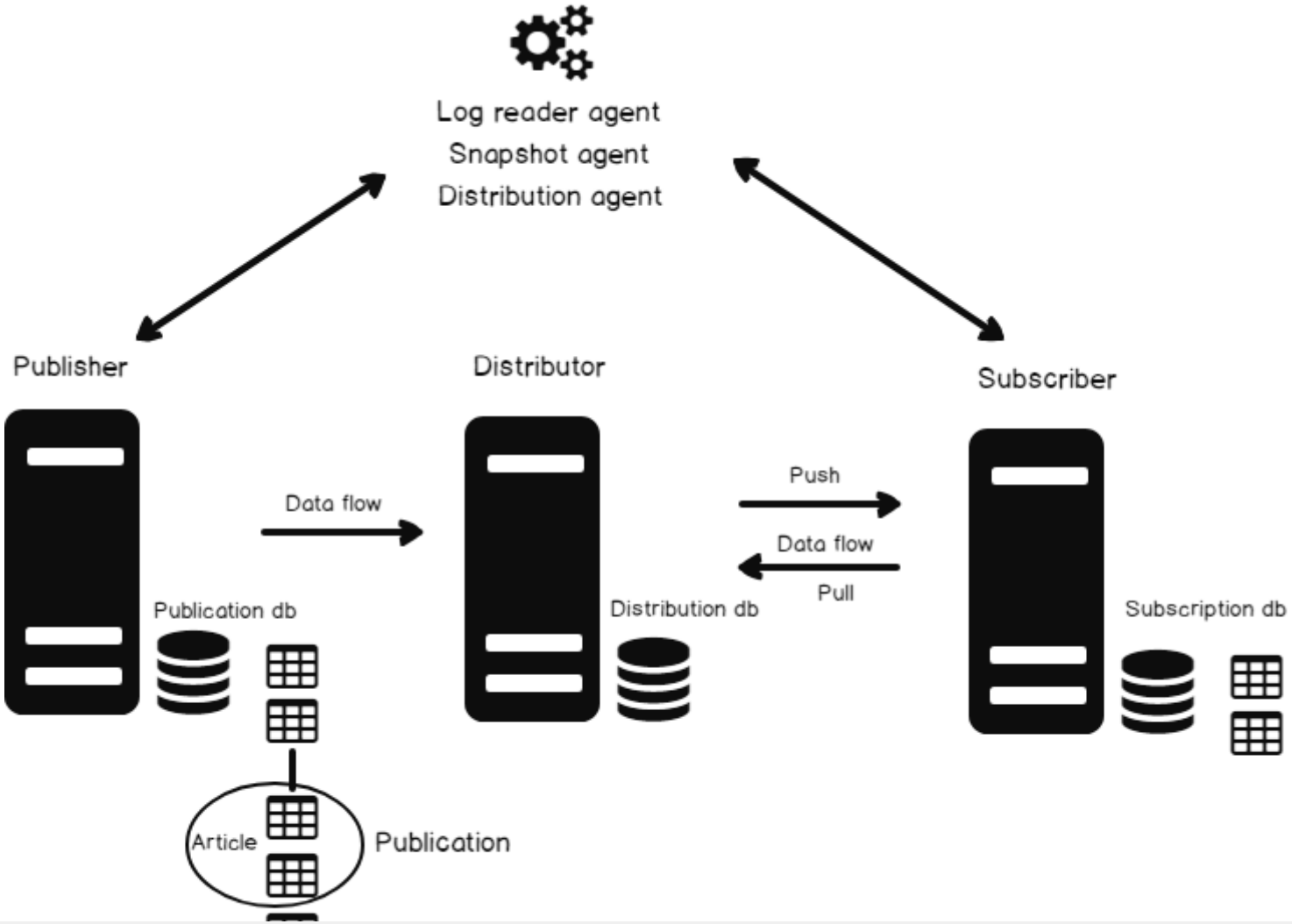
SQL Server replication is a technology for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency and integrity of the data. In most cases, replication is a process of reproducing the data at the desired targets. SQL Server replication is used for copying and synchronizing data continuously or it can also be scheduled to run at predetermined intervals. There are several different replication techniques that support a variety of data synchronization approaches; one-way; one-to-many; many-to-one; and bi-directional, and keep several datasets in sync with each other.

Transactional SQL Server replication components

The following diagram depicts the components of transactional SQL Server replication.

Including the SQL Server replication …

- Publisher
- Publication database
- Publication
- Articles
- Distributor



article, horizontally and vertically using a filter option. We can also create multiple articles on the same object with some restrictions and limitations.

Using the New Publication wizard, the **Article can be navigated**. It allows us to view the properties of an article and provide options to set properties for the articles. In some case, the properties can be set during the time of publication creation and it's a read-only property.

After the creation of a SQL Server replication publication, for instance, if some property requires a change, it will, in turn, require a new replication snapshot to be generated. If the publication has one or more subscriptions then the change requires all subscriptions to be reinitialized. For more information, see How to add/drop articles to/from existing publication in SQL Server article.

To list all the articles that are published, run the following T-SQL

```
SELECT
     Pub.[publication]    [PublicationName]
    ,Art.[publisher_db]   [DatabaseName]
    ,Art.[article]        [Article Name]
    ,Art.[source_owner]   [Schema]
    ,Art.[source_object]  [Object]
FROM
    [distribution].[dbo].[MSarticles]  Art
    INNER JOIN [distribution].[dbo].[MSpublications] Pub
        ON Art.[publication_id] = Pub.[publication_id]
ORDER BY
    Pub.[publication], Art.[article]
```

To get the details of articles in transactional or merge SQL Server replication in a published database, run the following T-SQL.

```
SELECT st.name [published object], st.schema_id, st.is_published , st.is_merge_published, is_schema_published
FROM sys.tables st WHERE st.is_published = 1 or st.is_merge_published = 1 or st.is_schema_published = 1
UNION
SELECT sp.name, sp.schema_id, 0, 0, sp.is_schema_published
```

```
EXEC sp_helparticle
  @publication = @publication;
GO
```

To get column level details, run the following T-SQL

```
USE MES_PROD_AP
GO
sp_helparticlecolumns  @publication = N'PROD_HIST_Pub' ,  @article =  'tb_Branch_Plant'
```

Results Messages

	column id	column	published	publisher type	subscriber type
1	1	PlantCode	1	nvarchar(20)	nvarchar(20)
2	2	TAPPIPlantCode	1	char(1)	char(1)
3	3	TimeZone	1	nvarchar(40)	nvarchar(40)
4	4	TAPPICompanyId	1	char(2)	char(2)
5	5	TAPPIPosition13	1	char(1)	char(1)
6	6	TAPPIPosition15	1	char(1)	char(1)
7	7	TAPPIEndTime	1	datetime	datetime
8	8	LogInboundSuccess	1	bit	bit
9	9	UseTAPPIInitials	1	bit	bit
10	10	TrimMergeOptionsAvailable	1	bit	bit
11	11	CrossWinding	1	bit	bit
12	12	LastUpdateId	1	nvarchar(100)	nvarchar(100)
13	13	LastUpdate	1	datetime	datetime
14	14	UpdateStamp	1	timestamp	timestamp
15	15	PurgeInterval	1	int	int
16	16	LotNumLetterStart	1	char(1)	char(1)

```
SELECT object_name(object_id) [published table], name [published column] FROM sys.columns sc WHERE sc.is_replicated =
1;
```

Publications

A Publication is a logical collection of articles from a database. The entity allows us to define and configure article properties at the higher level so that the properties are inherited to all the articles in that group.

```
EXEC sp_helppublication;
```

Publisher database

The publisher is a database that contains a list of objects that are designated as SQL Server replication articles are known as *publication database*. The publisher can have one or more publications. Each publisher defines a data propagation mechanism by creating several internal replication stored procedures.

```
USE Distribution
GO
select * from MSpublications
```

Results Messages

	publisher_id	publisher_db	publication	publication_id	publication_type	thirdparty_flag	independent_agent	immediate_sync	allow_push	allow_pull	allow_anonymous	description	vendor_name	retention
1	0	[illegible]_PROD_[illegible]	[illegible]_PROD_REPL_[illegible]	10	0	0	0	0	1	1	0		Microsoft SQL Server	336
2	0	[illegible]_PROD_[illegible]	[illegible]_PROD_HIST_[illegible]	11	0	0	0	0	1	1	0		Microsoft SQL Server	336

Distributor

The Distributor is a database that acts as a storehouse for replication specific data associated with one or more Publishers. In many cases, the distributor is a single database that acts as both the Publisher and the Distributor. In the context of SQL Server replication, this is commonly known as a “local distributor”. On the other hand, if it’s configured on a separate server, then it is known as a “remote distributor”. Each Publisher is associated with a single database known as a “distribution database” aka the “Distributor”.

The distribution database identifies and stores SQL Server replication status data, metadata about the publication, and, in some cases, acts as a queue for data moving from the Publisher to the Subscribers.

Depending on the replication model, the Distributor might also be responsible for notifying the Subscribers that have subscribed to a publication that an article has changed. Also, the distribution database maintains the integrity of the data.

Distribution databases

Each Distributor must have at least one distribution database. The distribution database consists of article detail, replication meta-data and data. A Distributor can hold more than one distribution database; however, all publications defined on a single Publisher must use the same distribution database.

To find out whether …

- a distribution database installed or not?

```
SELECT name FROM sys.databases WHERE is_distributor = 1
```

- a Publisher is using this Distributor or not?

```
EXEC sp_get_distributor
```

Results Messages

	installed	distribution server	distribution db installed	is distribution publisher	has remote distribution publisher
1	1	...DP01	1	1	0

- or just to interrogate various Distributor and Distribution database properties?

```
EXEC sp_helpdistributor;
EXEC sp_helpdistributiondb;
EXEC sp_helpdistpublisher;
```

Subscriber

A database instance that consumes SQL Server replication data from a publication is called a Subscriber. The subscriber can receive data from one or more publishers and publications. The subscriber can also pass data changes back to the publisher or republish the data to other subscribers depending on the type of the replication design and model.

There are two types of subscriptions: push subscriptions and pull subscriptions

- Push subscription: Distributor directly updates the data in the Subscriber database
- Pull subscription: the Subscriber is scheduled to check at the Distributor regularly if any new changes are available, and then updates the data in the subscription database itself.

```
EXEC sp_helpsubscription;
```

Subscription databases

A target database of a replication model is called a subscription database.

Replication agents

SQL Server replication uses a pre-defined set of standalone programs and events are known as agents, to carry out the tasks associated with data. By default, SQL Server replication agents run as scheduled jobs under SQL Server Agent. Replication agents can also be run from the command line and by applications that use Replication Management Objects (RMO). SQL Server replication agents can be monitored and administered using Replication Monitor and SQL Server Management Studio.

Replication snapshot Agent

The Log Reader Agent is used only with transactional replication. It moves replication transactions from the online transaction log of the publication database to the distribution database.

Distribution Agent

The Distribution Agent is used only with Replication snapshot and Transactional SQL Server replication. This agent applies the initial replication snapshot to the subscription database and later, the data changes are tracked and recorded in the distribution database and applied to the subscription database.

Merge Agent

The Merge Agent is used with the merge replication model. By default, the Merge Agent uploads changes from the Subscriber to the Publisher and then downloads changes from the Publisher to the Subscriber. Each subscription has its own Merge Agent that connects to both the Publisher and the Subscriber and updates both. The Merge Agent runs at either the Distributor for push subscriptions or the Subscriber for pull subscriptions. Here, the synchronization is bi-directional. The data conflicts are handled by a set of triggers that supports the entire process

Summary

Thus far, we've seen a walk-through of some of the important concepts of SQL Server replication. Also, T-SQL scripts are shown to query system tables and replication stored procedures to answer most of the commonly asked question about SQL Server replication.

SQL Replication: Basic setup and configuration

How to Add/Drop articles from existing publications in SQL Server

How to do a quick estimated compare of data in two large SQL Server databases to see if they are equal

SQL Server transactional replication: How to reinitialize a subscription using a SQL Server database backup

How to setup a custom SQL Server transaction replication model with a Central Subscriber and Multiple Publisher databases

How to setup custom SQL Server transactional replication with a central publisher and multiple subscriber databases

How to set up a DDL and DML SQL Server database transactional replication solution

How to setup cross-platform transactional SQL Server replication for database reporting on Linux

SQL Server database migrations with zero data loss and zero downtime

Using transactional data replication to replay and test production loads on a staging server

How to setup SQL Server database replication for a reporting server

SQL Server transactional replication: How to reinitialize a subscription using a “Replication support only” –TBA

SQL Server Replication Monitoring and setting alerts using PowerShell –TBA

See more

Certified Professional and backed with a Degree in Master of Computer Application.

My specialty lies in designing & implementing High availability solutions and cross-platform DB Migration. The technologies currently working on are SQL Server, PowerShell, Oracle and MongoDB.

View all posts by Prashanth Jayaram

Related Posts:

1. **Top SQL Server Books**
2. **SQL Server replication: Configuring Snapshot and Transactional Replication**
3. **SQL Server database migration to Azure SQL Database using SQL Server Transactional Replication**
4. **SQL Server Transactional Replication on Linux**
5. **SQL Server replication configuration: Peer to Peer and Merge Replication**

Replication, Transaction log

87,736 Views

ALSO ON **SQL SHACK**

OLAP Cubes in SQL | How to install MySQL | Visualize Coronavirus | Lock Confi

Jonathan Nickle • a year ago

Thank you for these published articles... very well written. I've been dev in SQL Server for web technologies for over 20 years now, but just now looking into possible replication solution. Your info is a great asset. One question, this appears to be the first article in the series, and you reference "run this query" ... is there a sample DB that you can share (don't see a link) so that I can run them locally?