Merge Replication 101

SQL Saturday #389

April 11, 2015

Diana Dee

DrDianaDee@verizon.net

Consultant

TribalSQL Author

Adjunct Instructor, Kaplan University

Contents

- SQL Server Replication Overview
- Main Types of SQL Server Replication
- Merge Replication Overview
- Merge Replication Setup
- Merge Replication Demonstration
- Why I Love Merge Replication

SQL Server Replication Overview

- Replication enables copying and distributing data and database objects from one database to another, then synchronizing between databases to maintain consistency.
- Data can be made available closer to users or applications that query the data.
- Data changes made at multiple locations can be integrated.

Reference: SQL Server Books Online, “SQL Server Replication”

Types of SQL Server Replication

1. Snapshot replication
   - Point-in-time copy
   - Used to initialize transactional and merge data
2. Transactional replication
   - Best for placing “read-only” data closer to the users
3. Merge replication
   - Best for allowing changes to be made in any copy

Replication Paradigm

Publisher

the “original” data

Subscriber

a copy of the data

Distributor

the technology for distributing the data

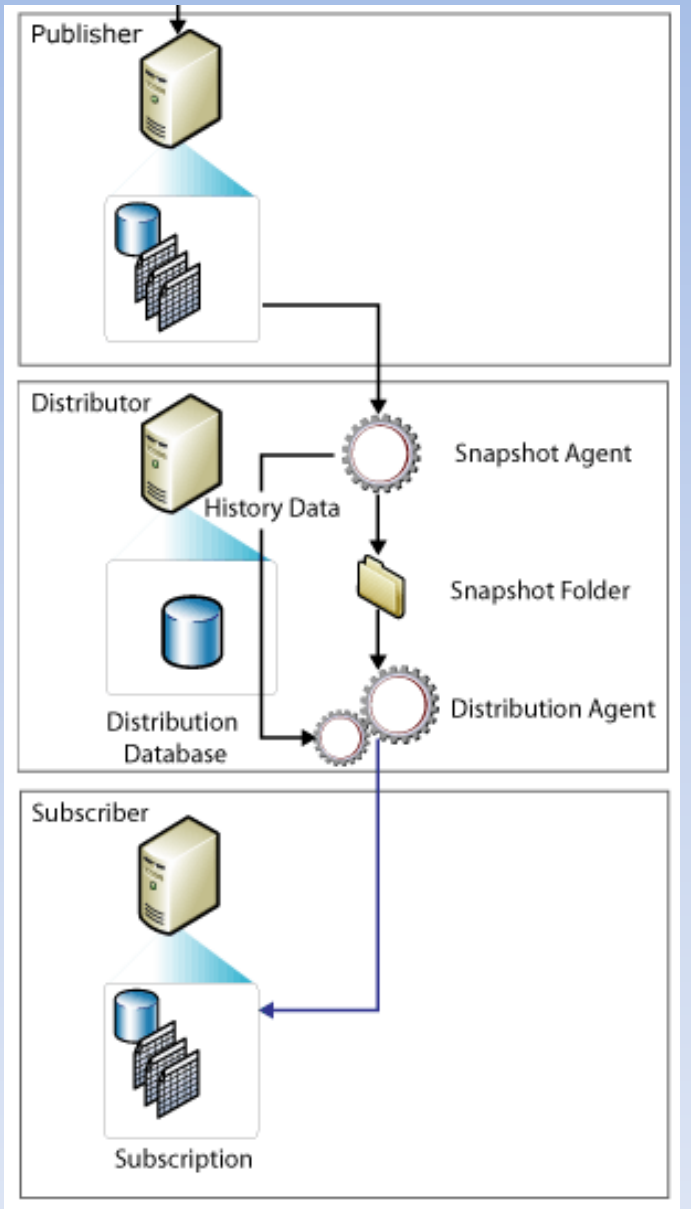
Article

an object (e.g. a table) that is replicated

Snapshot Replication

- Point-in-time copy
- Applying a snapshot to a subscriber overwrites the previous data
- Useful if data volume is small and if it is acceptable that data at the subscriber is out of date with respect to data at the publisher
- Appropriate when data changes are substantial but infrequent, as in a price list update or a purchased business list update
- Used to initialize a transactional or merge subscriber

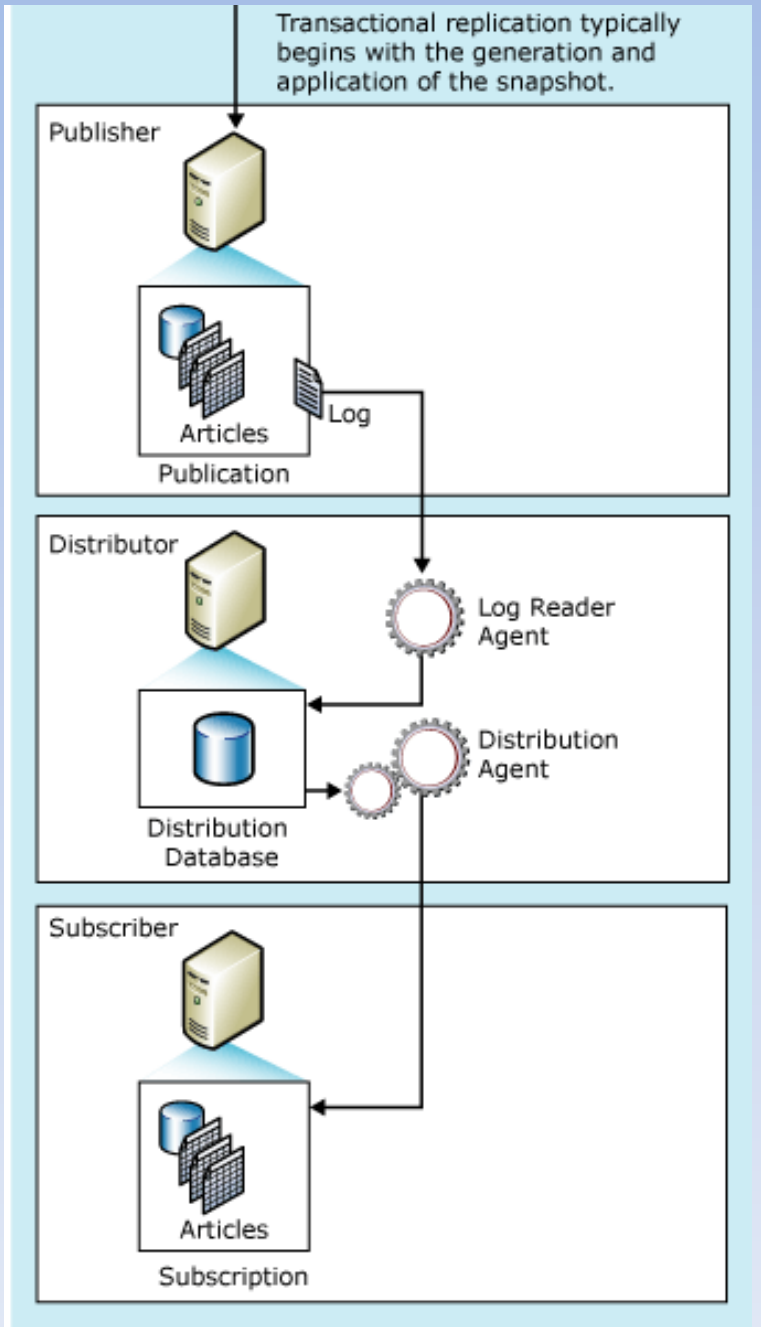
Snapshot Replication



Transactional Replication

- Use when subscriber should receive incremental data changes when they occur (very low latency)
- Use when there is very frequent insert, update, and delete activity at the publisher
- Treat subscriber as read-only. Although there are options to allow changes at the subscriber to be sent back to the publisher, those options require a good network connection or an external storage program.
- Changes are kept in and read from the transaction log

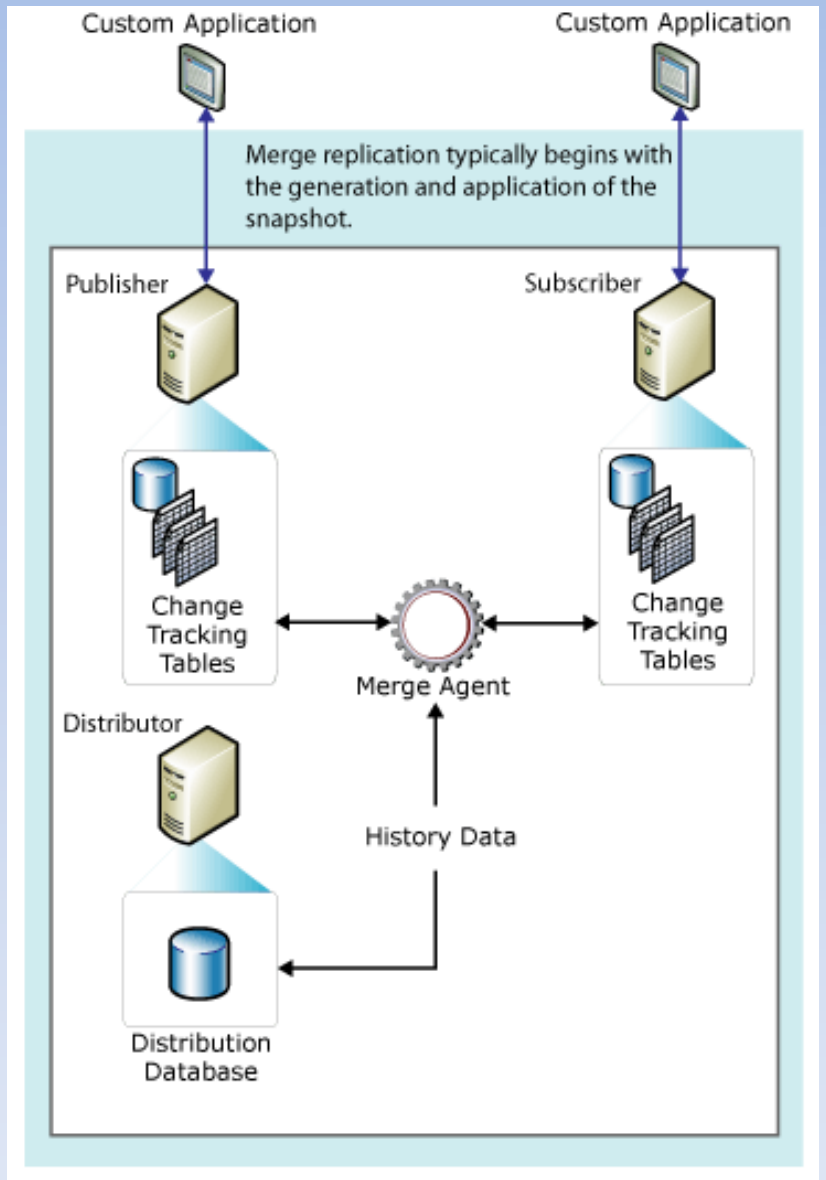
Transactional Replication



Merge Replication Overview

- Changes may be made at any copy; changes are merged to all other copies
- A copy may be offline for quite a while; when back online, data are merged
- Conflicts might occur; they can be detected and resolved
- Possible uses
  - "High availability" solution
  - Writable copies available in many locations
  - A subscriber may have a subset of the data

Merge Replication



Merge Replication Setup

- A merge subscriber is initialized using a snapshot
  - The snapshot files contain table schema, constraints and indexes, triggers, system tables, conflict tables, and table data
- If only a subset of tables are replicated, you can specify that foreign keys and/or triggers are not replicated
- Columns with the IDENTITY property (NOT FOR REPLICATION) may be assigned a value range for the publisher and subscriber(s), for each table

Merge Replication Setup, cont.

- Merge replication tracks changes by adding metadata
  - Each replicated table requires a column named "rowguid" of data type uniqueidentifier and default value newid().
  - Many system tables are added to the database
- The Distributor is usually on the same instance as the Publisher
  - The merge agent runs on the Distributor
  - The Distributor stores the snapshot

Merge Replication Demonstration

- Merge replication setup and conflict resolution will now be demonstrated
- Only the SalesOrderHeader (SOH) and SalesOrderDetails (SOD) tables from Microsoft's AdventureWorks2012 sample database are replicated
- The following slides show key screen images from the demonstration

Merge Replication Demonstration

Preliminary Tasks

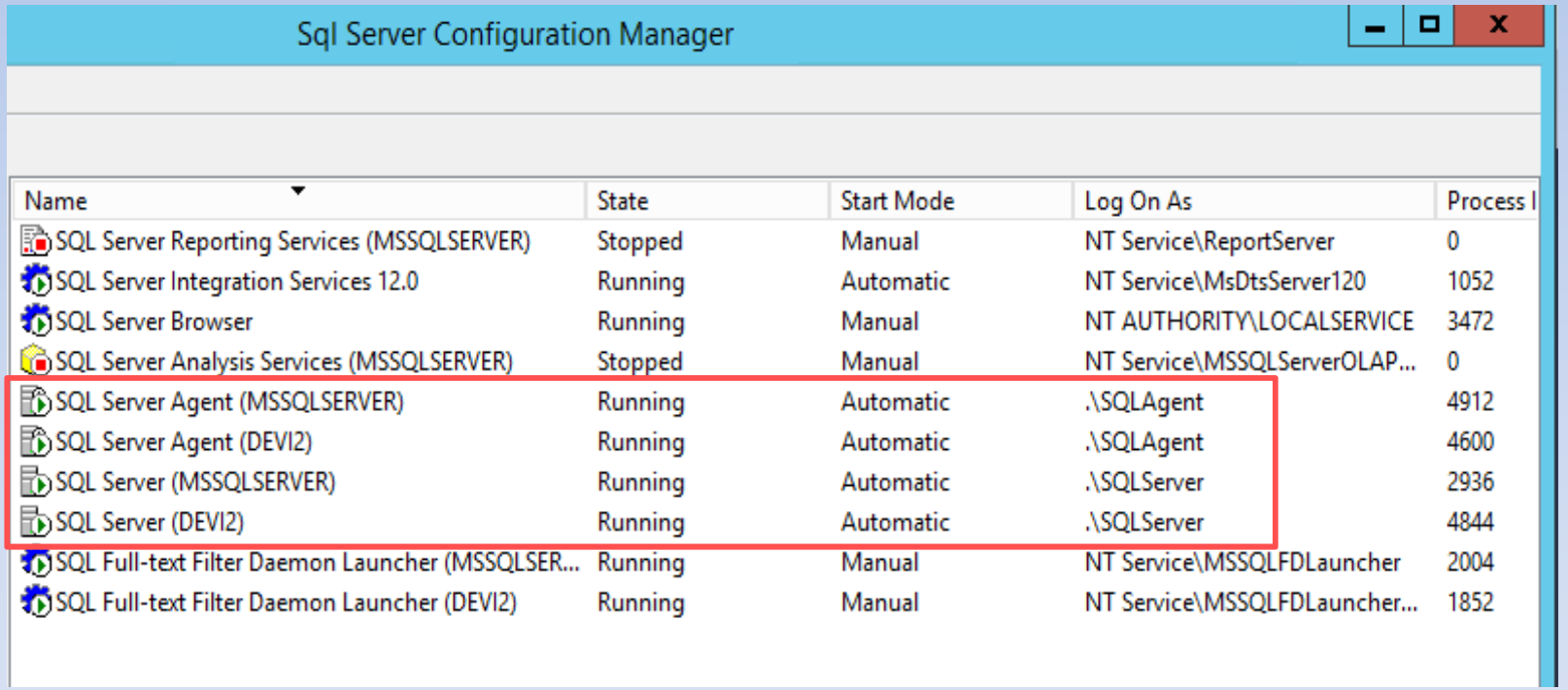
- Ensure the SQL Server and SQL Server Agent accounts are the same on all instances and have sysadmin membership on all instances

For this demonstration (AdventureWorks2012)

- Triggers and some Foreign Keys have been dropped from the SOH and SOD tables
- A stored procedure, Sales.AddOrder, has been created to simulate the addition of an order
- Scripts to accomplish these are available on request

Demo: Creating the Publication

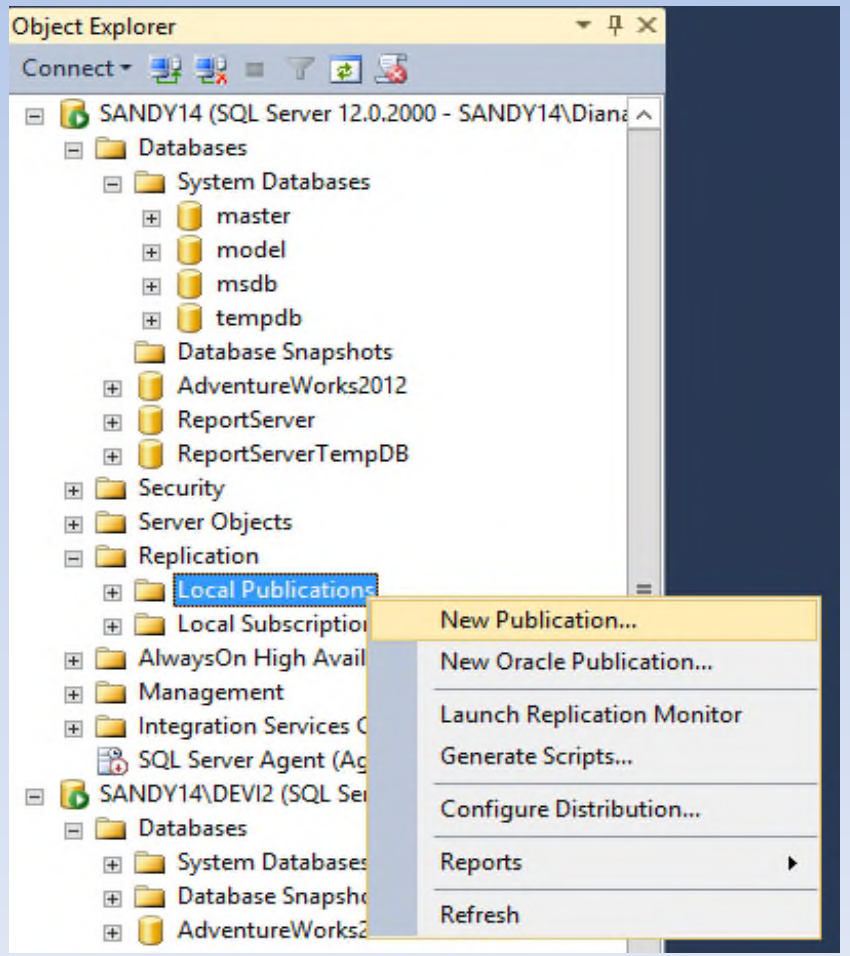
Service accounts for the default (SANDY14) and named (SANDY14\DEVI2) instances have been set



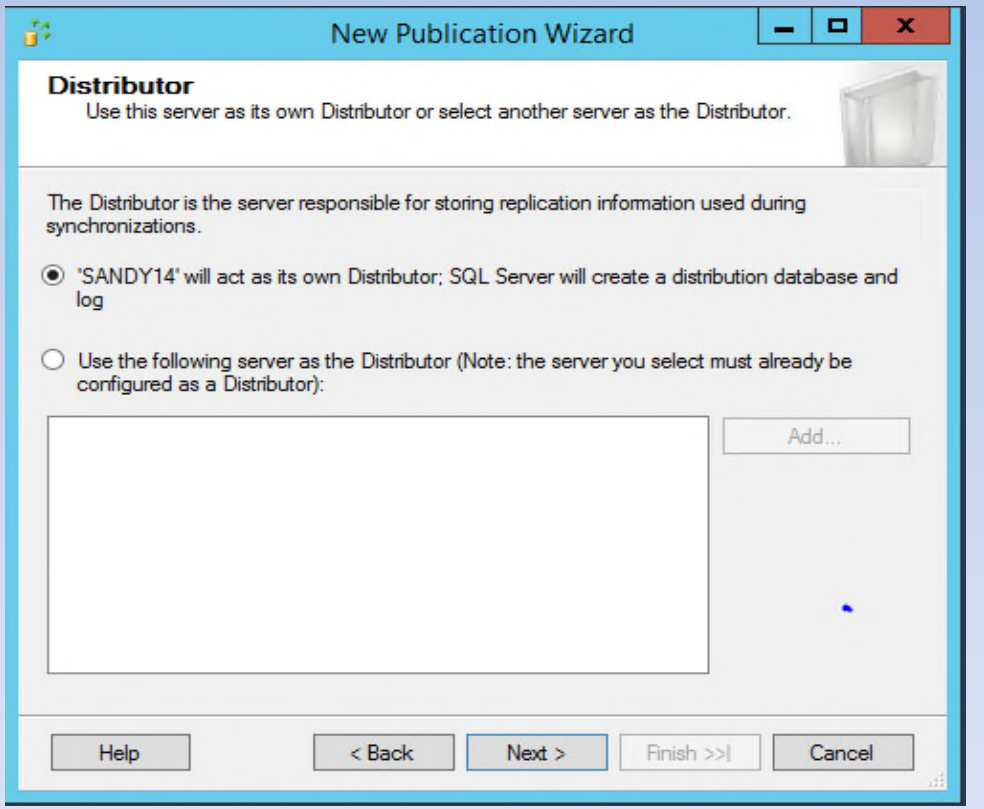
Name	State	Start Mode	Log On As	Process I
SQL Server Reporting Services (MSSQLSERVER)	Stopped	Manual	NT Service\ReportServer	0
SQL Server Integration Services 12.0	Running	Automatic	NT Service\MsDtsServer120	1052
SQL Server Browser	Running	Manual	NT AUTHORITY\LOCALSERVICE	3472
SQL Server Analysis Services (MSSQLSERVER)	Stopped	Manual	NT Service\MSSQLServerOLAP...	0
SQL Server Agent (MSSQLSERVER)	Running	Automatic	.\SQLAgent	4912
SQL Server Agent (DEVI2)	Running	Automatic	.\SQLAgent	4600
SQL Server (MSSQLSERVER)	Running	Automatic	.\SQLServer	2936
SQL Server (DEVI2)	Running	Automatic	.\SQLServer	4844
SQL Full-text Filter Daemon Launcher (MSSQLSER...	Running	Manual	NT Service\MSSQLFDLauncher	2004
SQL Full-text Filter Daemon Launcher (DEVI2)	Running	Manual	NT Service\MSSQLFDLauncher...	1852

Demo: Creating the Publication

Right-click **Local Publications** and then select **New Publication**

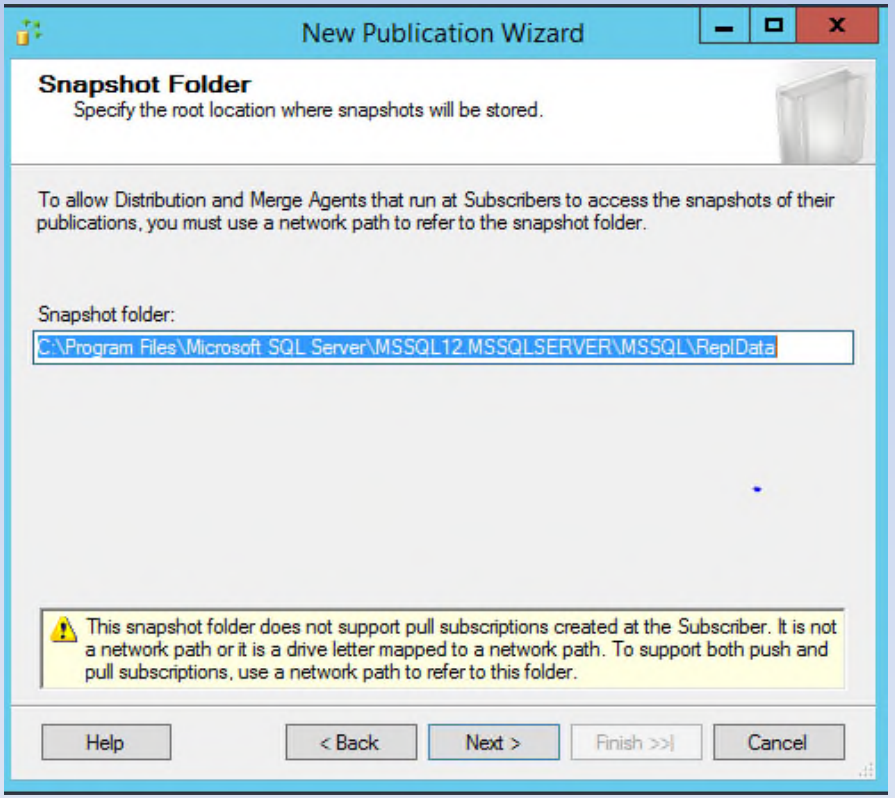


In the **New Publication Wizard** the publisher acts as its distributor

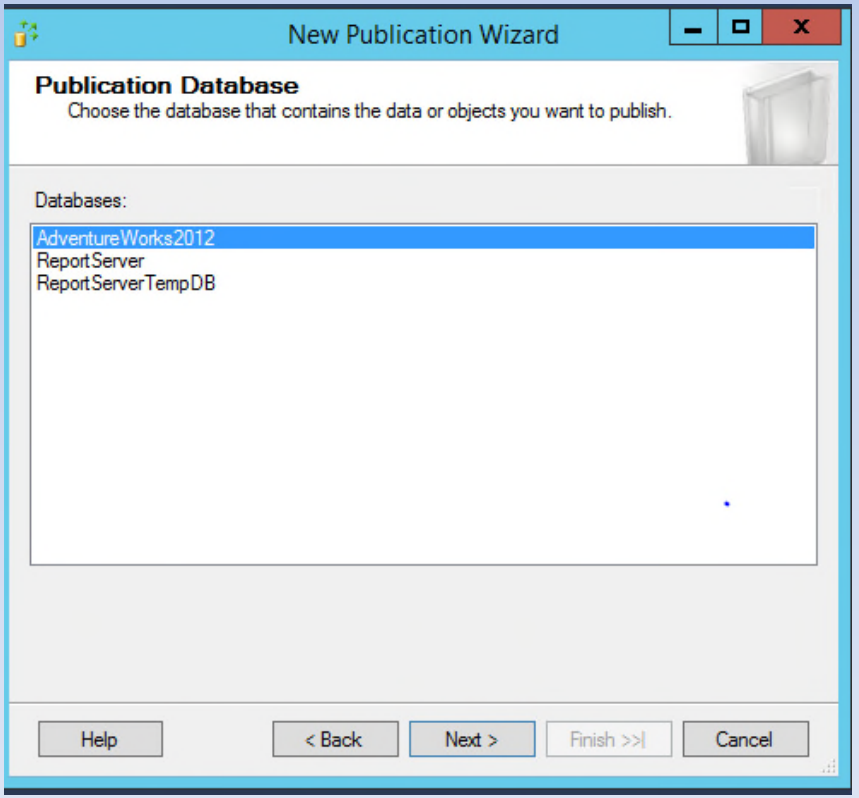


Demo: Creating the Publication

The snapshop folder, repldata, must be changed to a share if you need a *pull* subscriber.

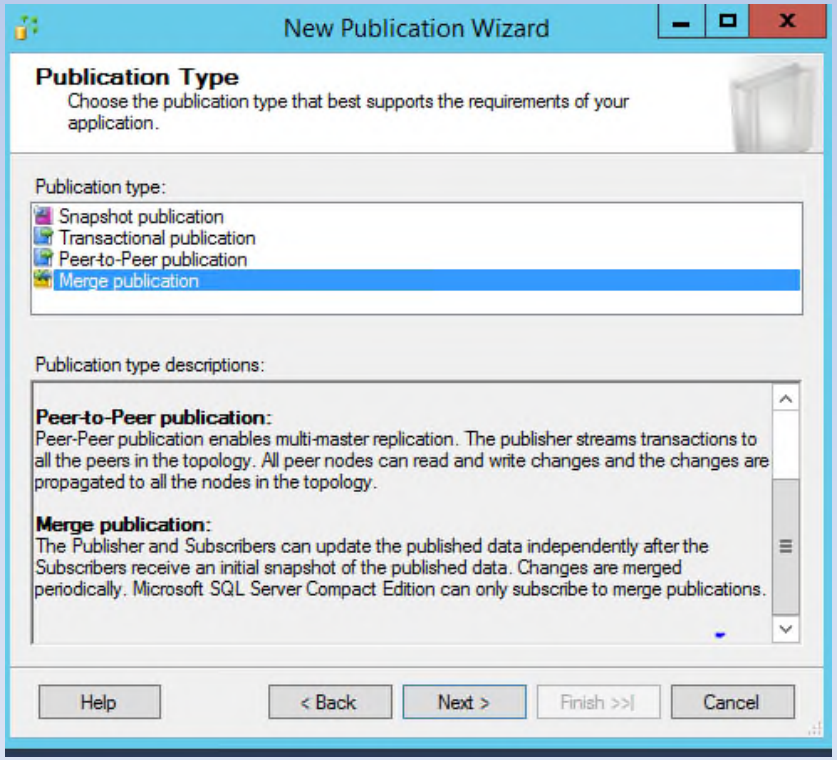


The publication database is chosen.

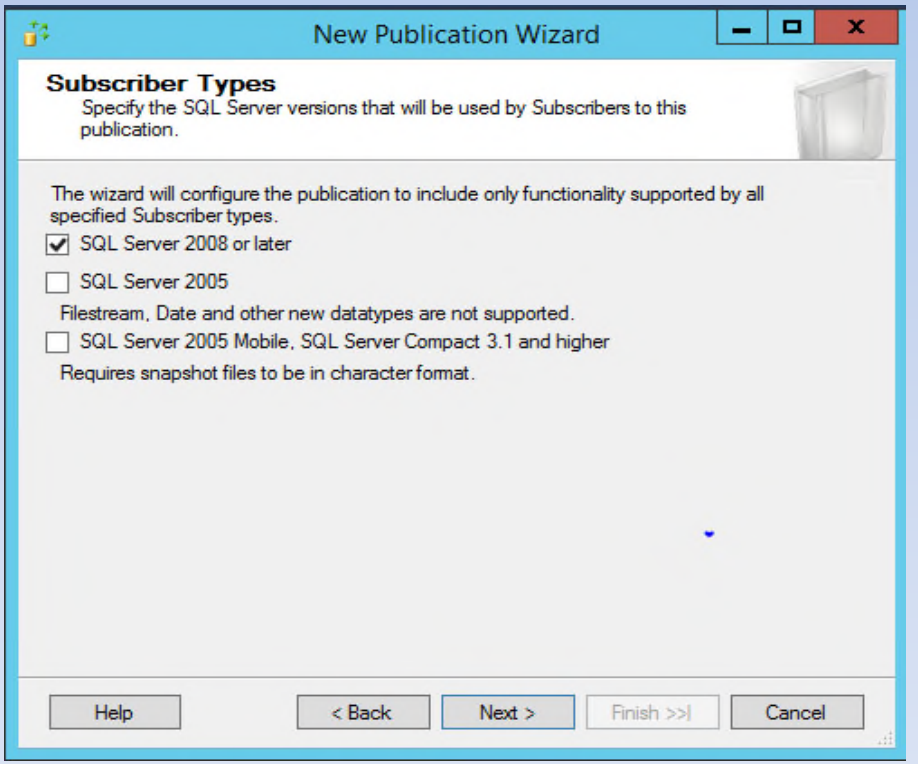


Demo: Creating the Publication

The type of replication, *merge*, is chosen.

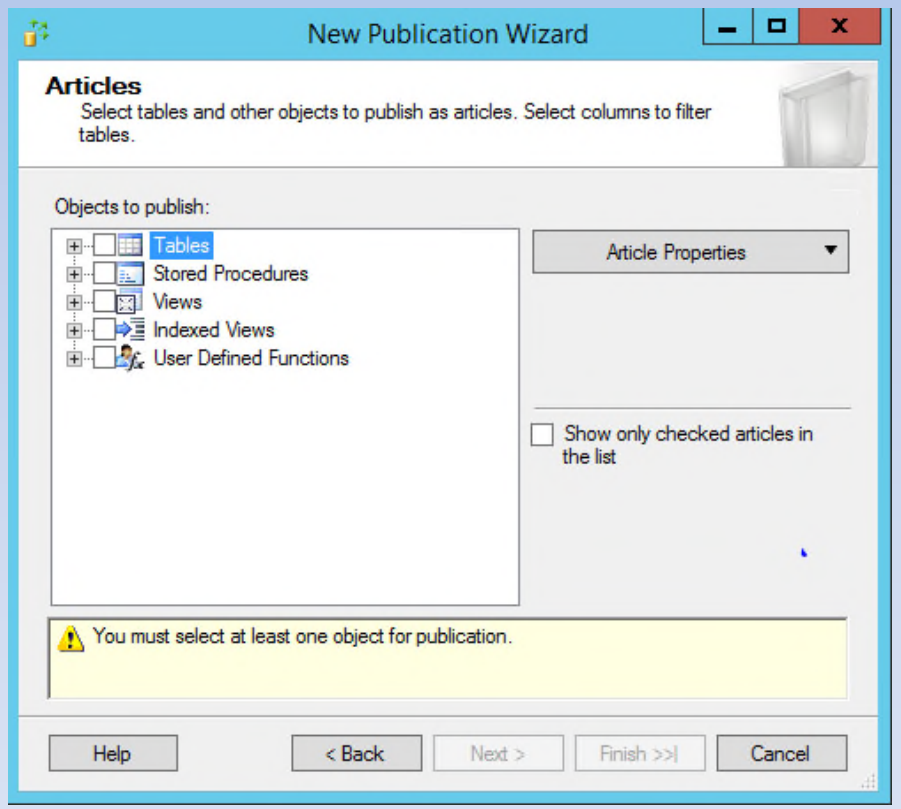


Subscriber versions are chosen. The publisher version must be equal to or later than the version of the subscriber.

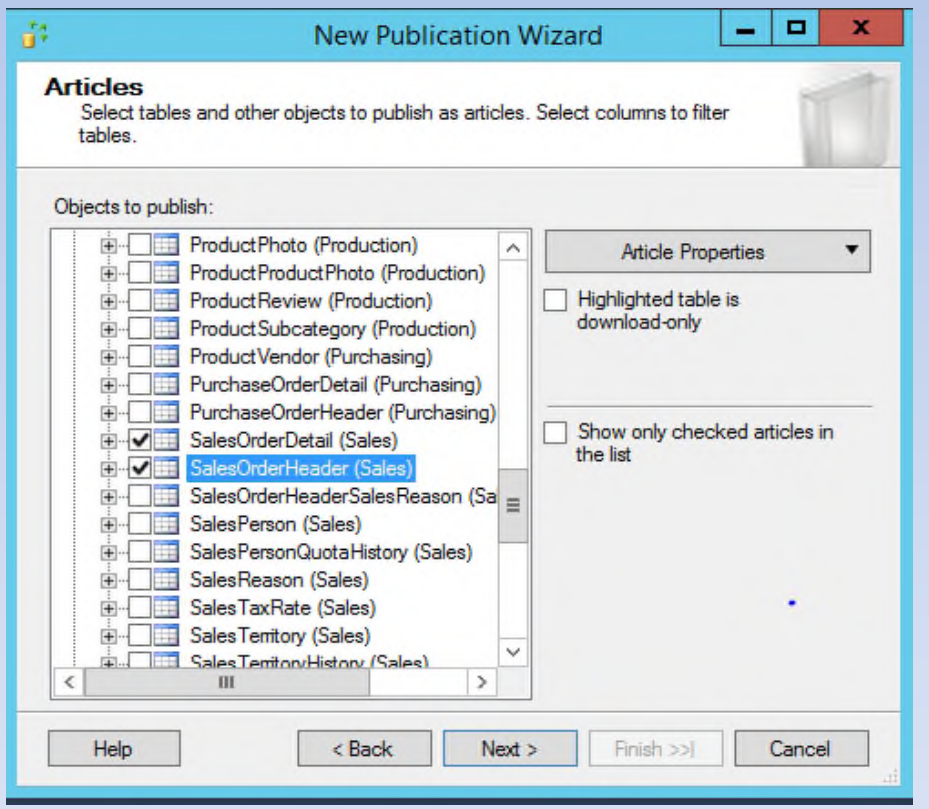


Demo: Creating the Publication

The tables to replicate are chosen.



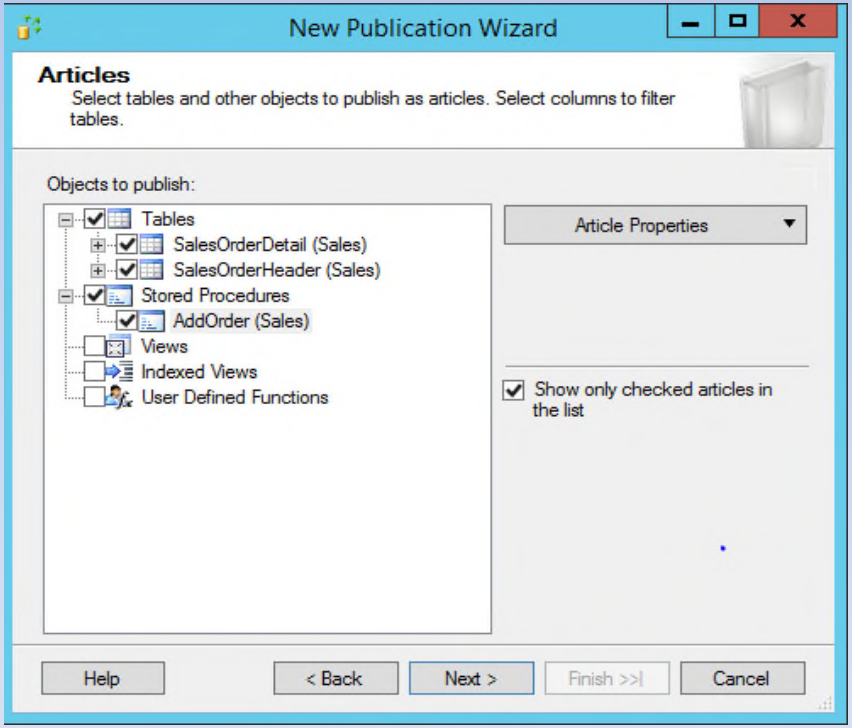
(Usually all tables would be replicated.)

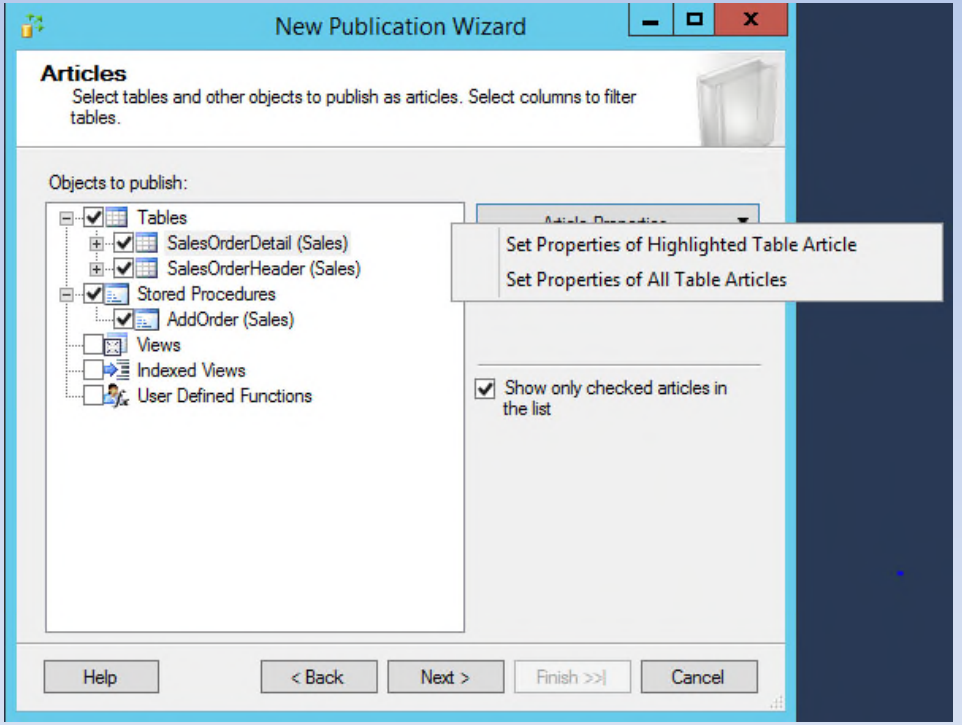


Demo: Creating the Publication

Also replicated is a stored procedure, Sales.AddOrder.

Click **Article Properties** to set properties of an article

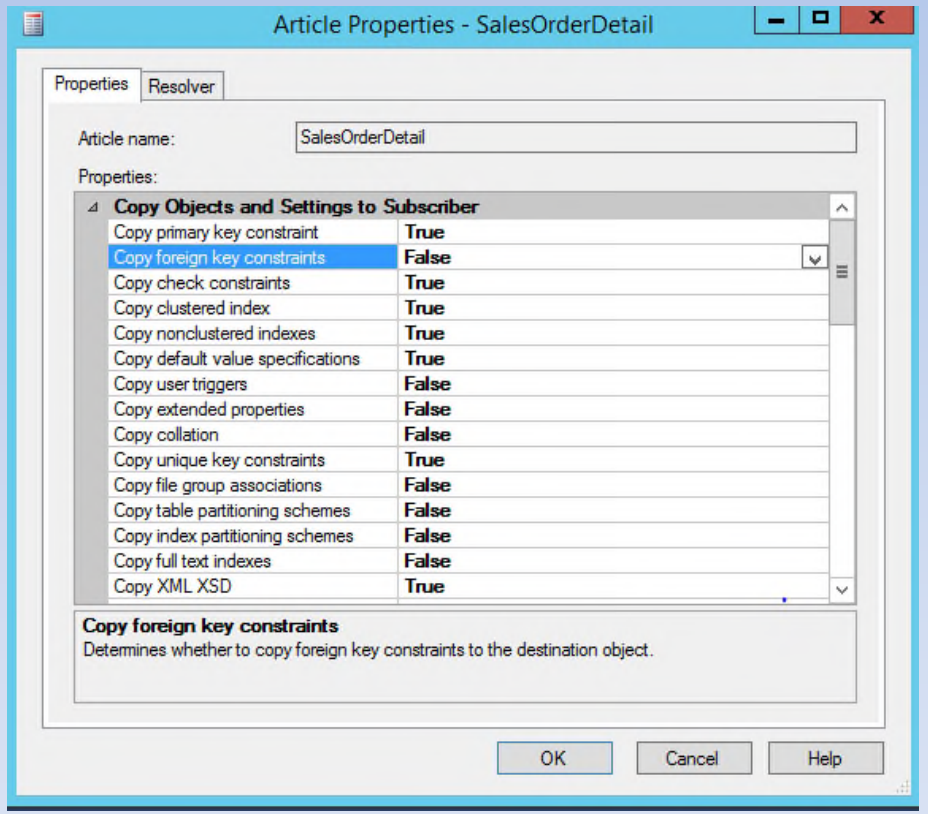


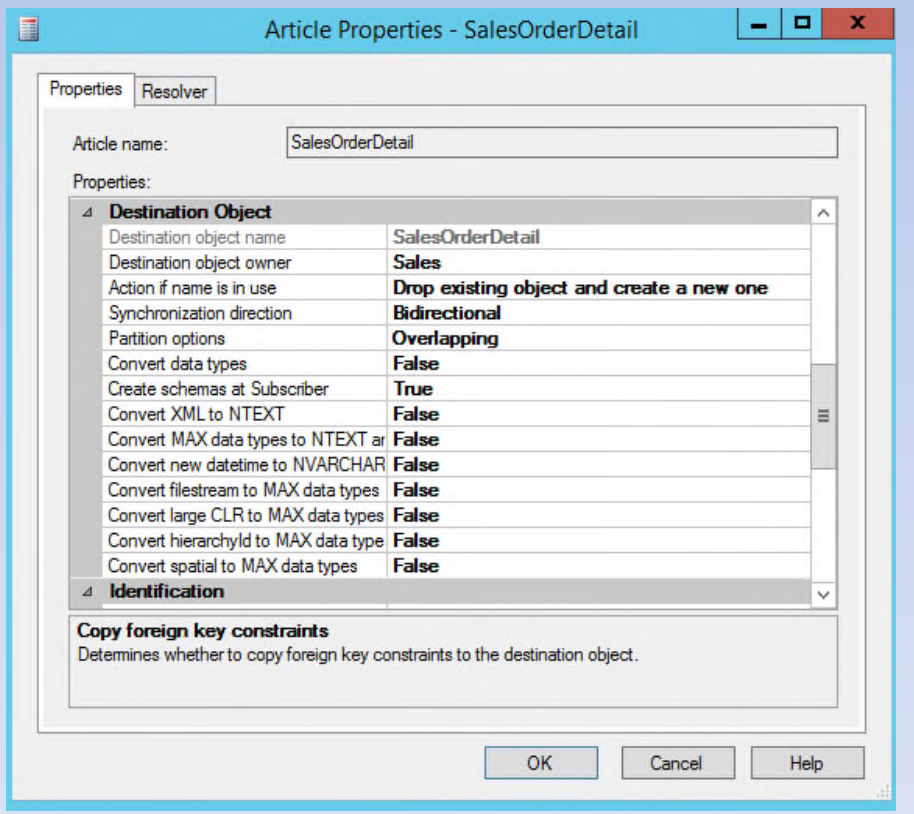


Demo: Creating the Publication

In this case, in SOD we chose not to replicate F.K.s or Triggers.

Other article properties (that we did not change).

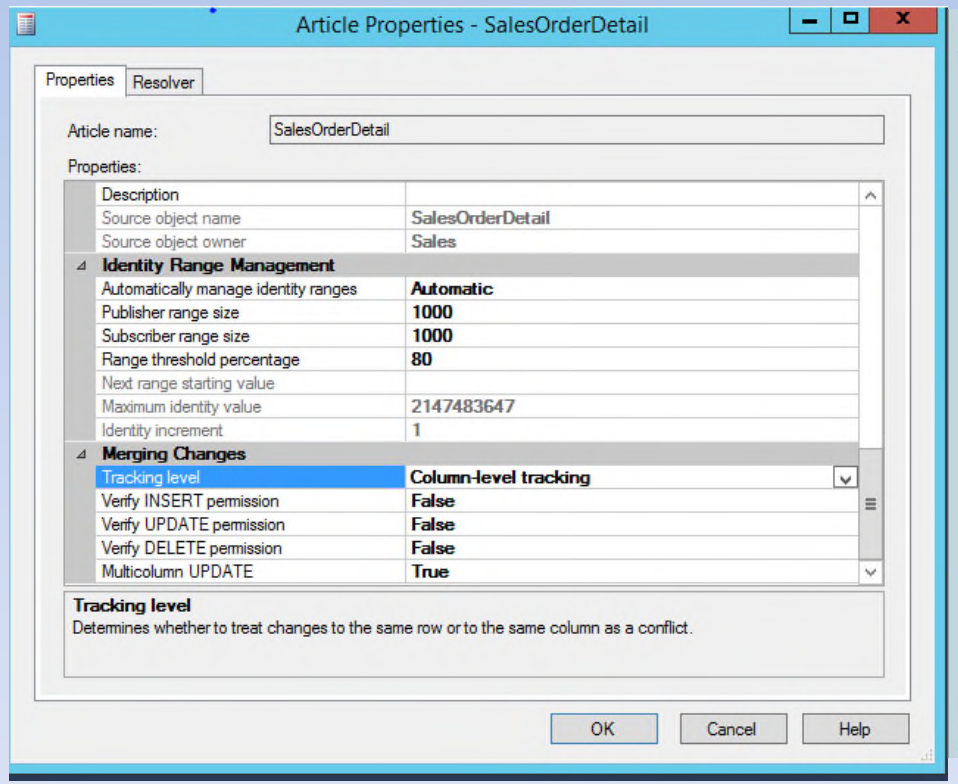


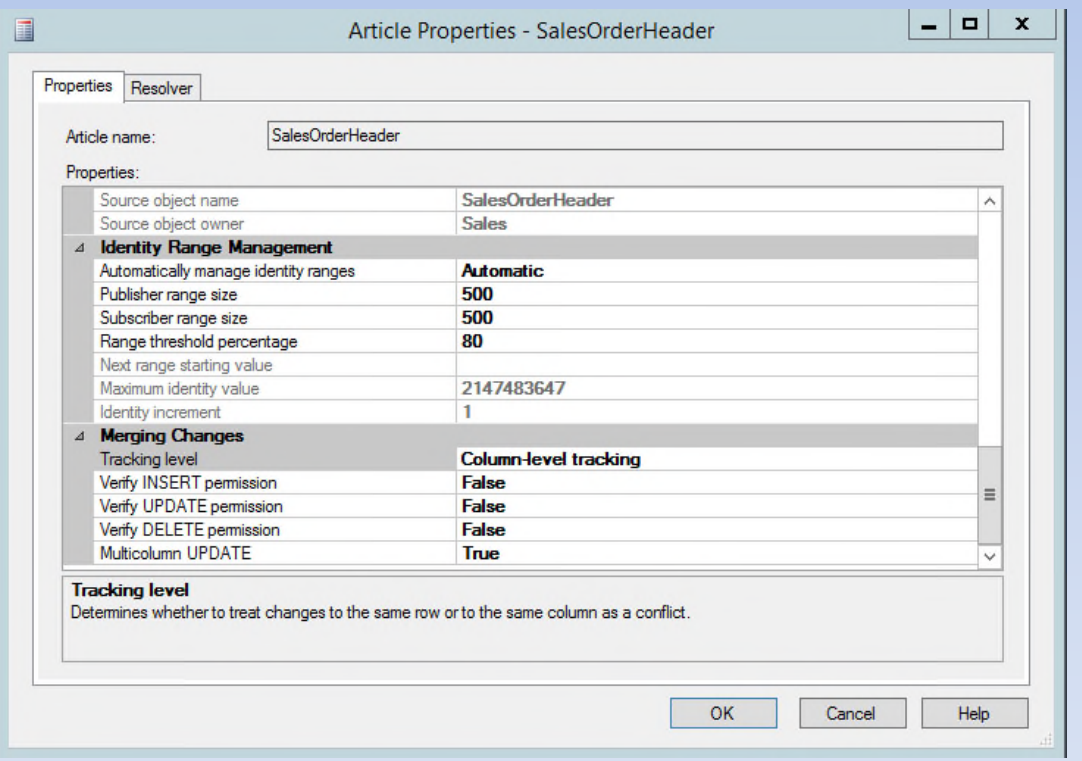


Demo: Creating the Publication

SOD dentity ranges and Column-level tracking were set.

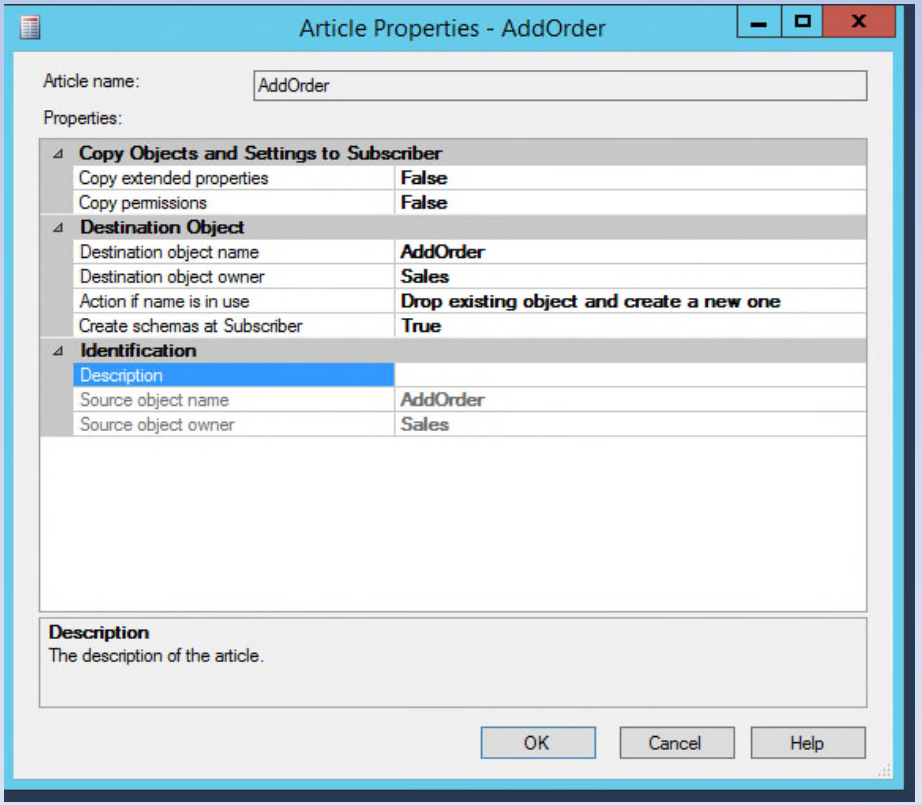
Not replicating F.K.s and triggers, column-level tracking, and identity ranges were set for SOH also.



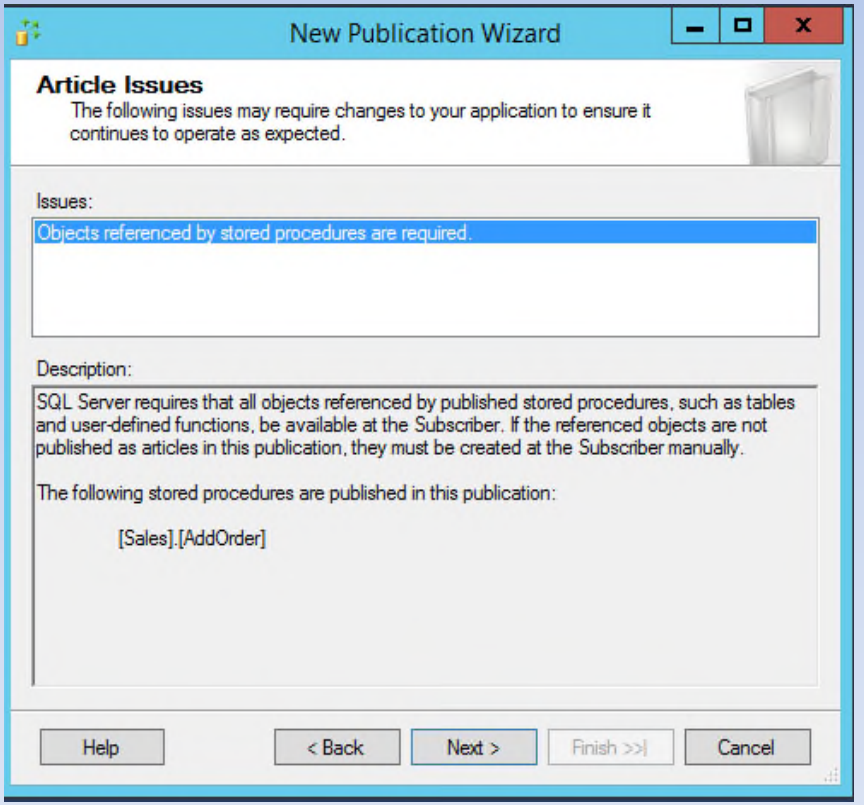


Demo: Creating the Publication

No article properties for the stored procedure was changed



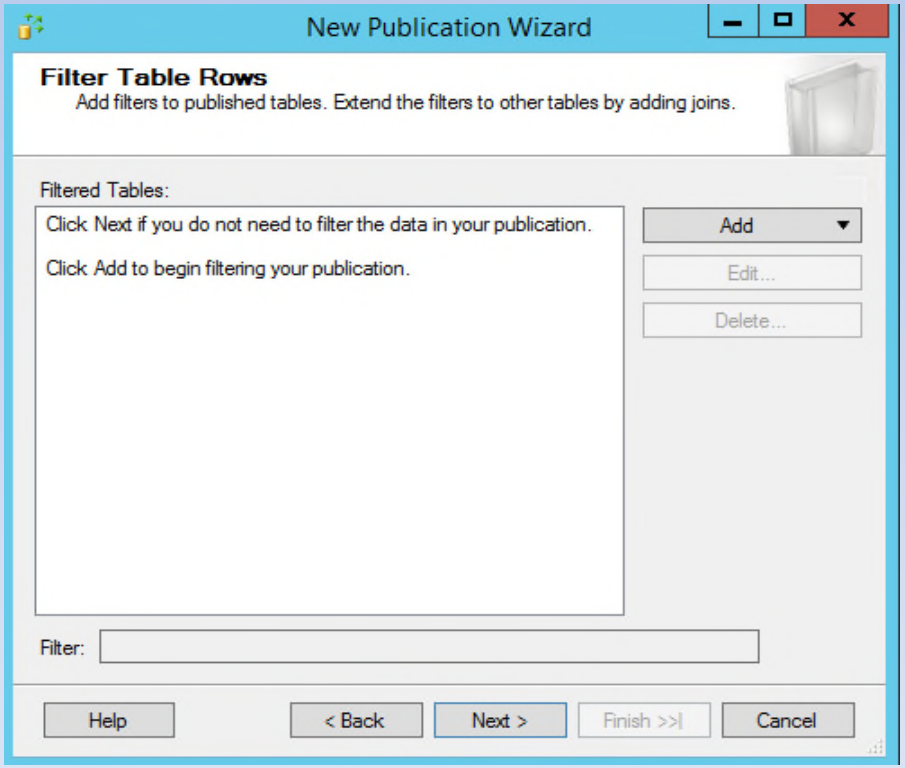
The Wizard warns of an issue when replicating a stored proc

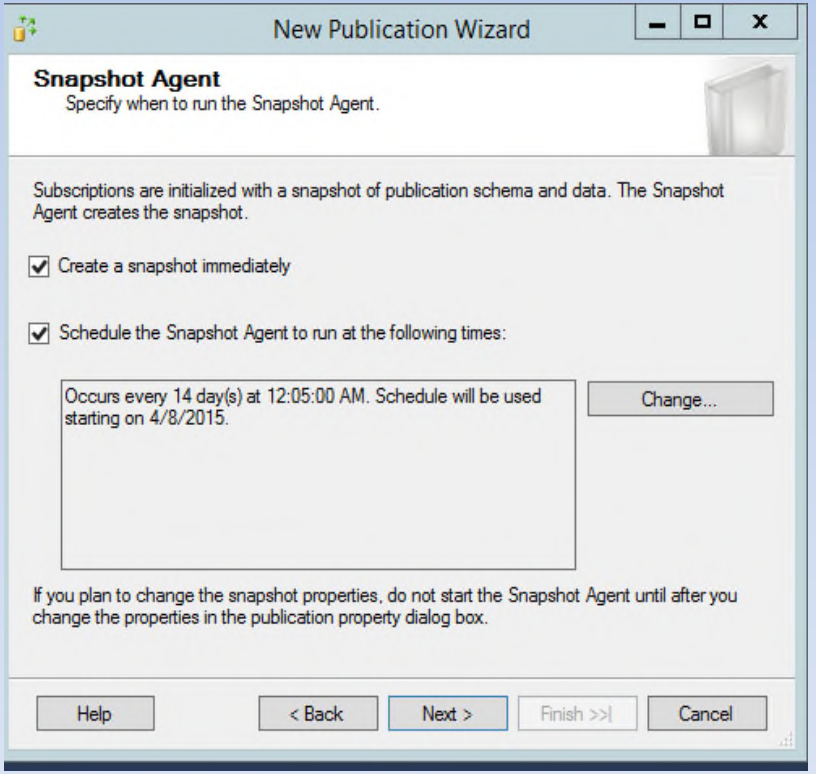


Demo: Creating the Publication

Table rows may be filtered; we did not.

We chose to generate the snapshot immediately and to make the expiration interval 14 days.

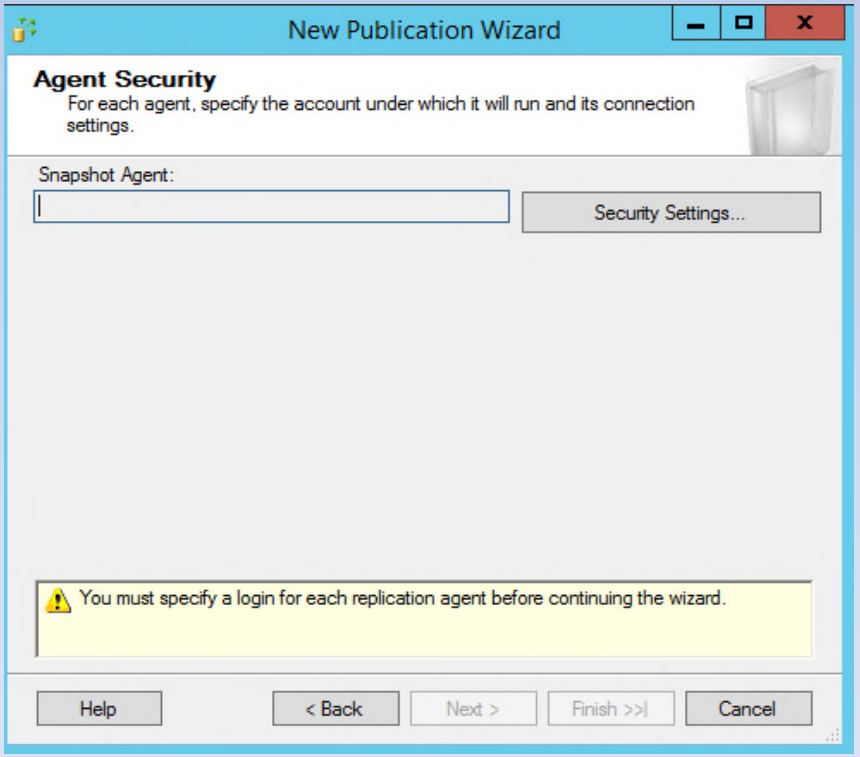




Demo: Creating the Publication

The Snapshot Agent account must be specified. Click the **Security Settings** button.

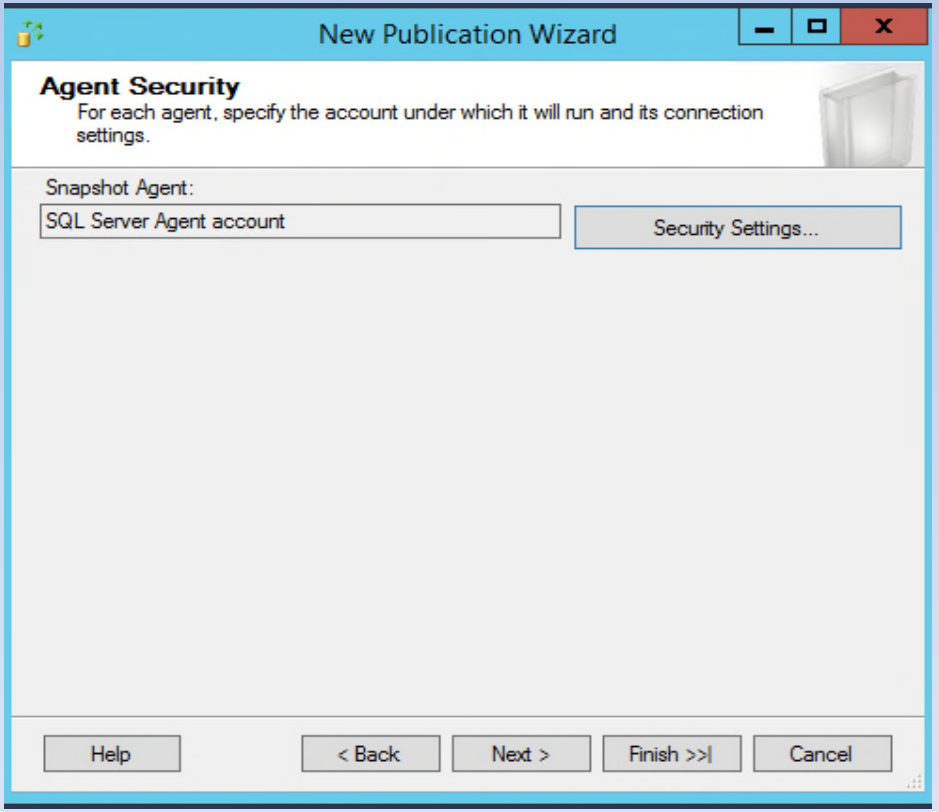
It is recommended that you have a unique user account for each replication agent; we chose to use the SQL Agent.

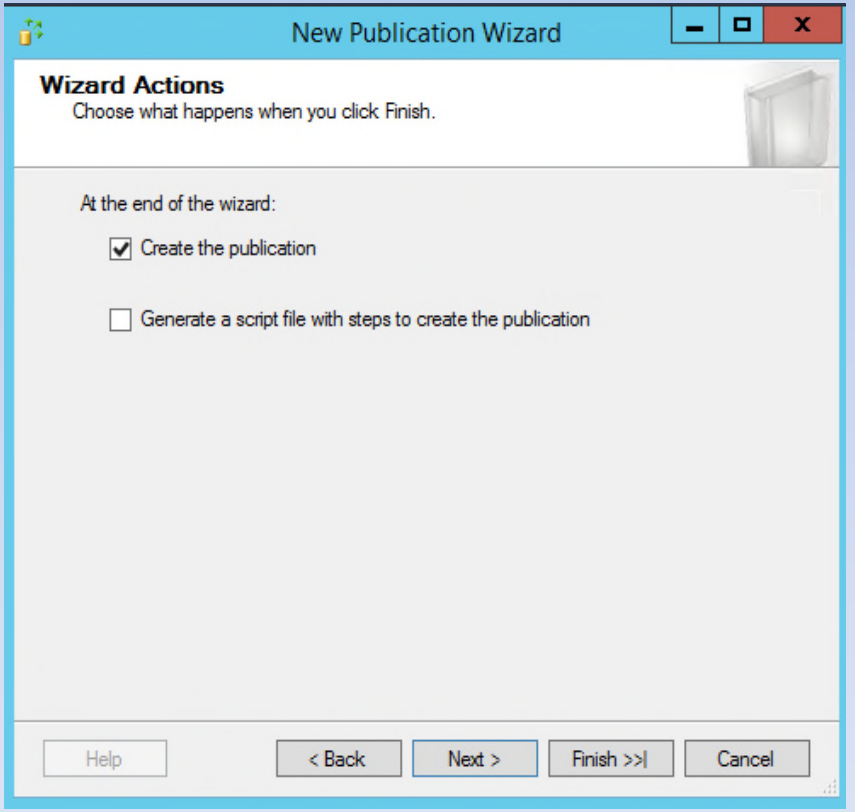


Demo: Creating the Publication

The **Agent Security** dialog shows the SQL Server Agent choice for the snapshot agent.

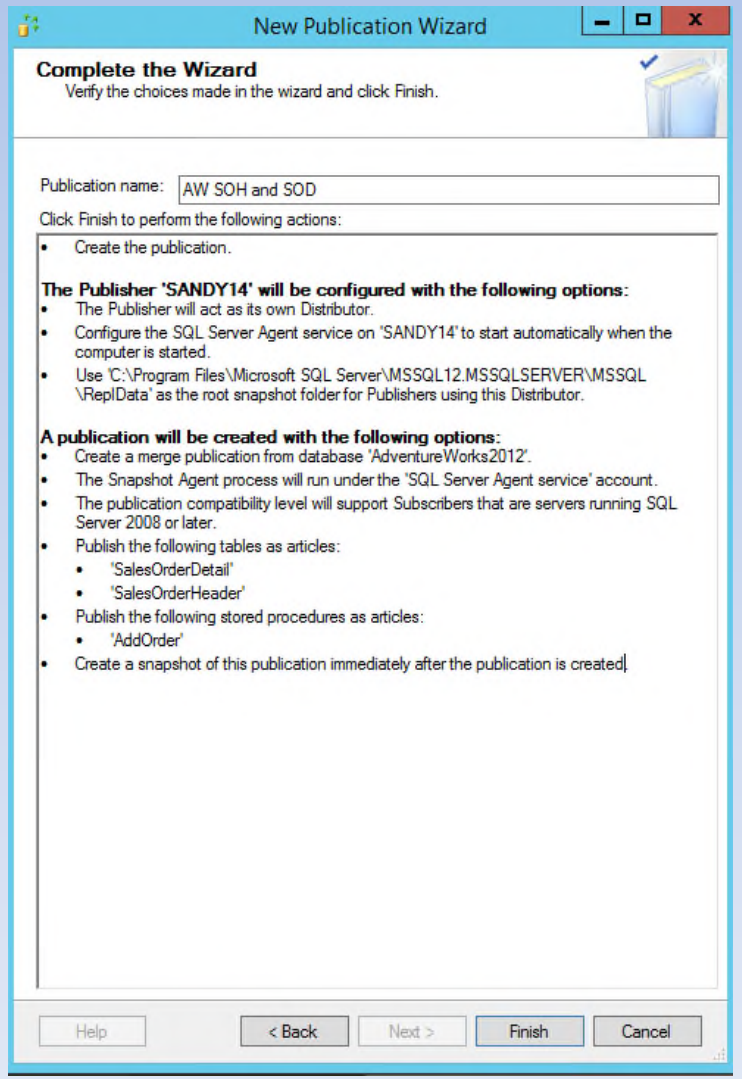
We choose to create the publicaton



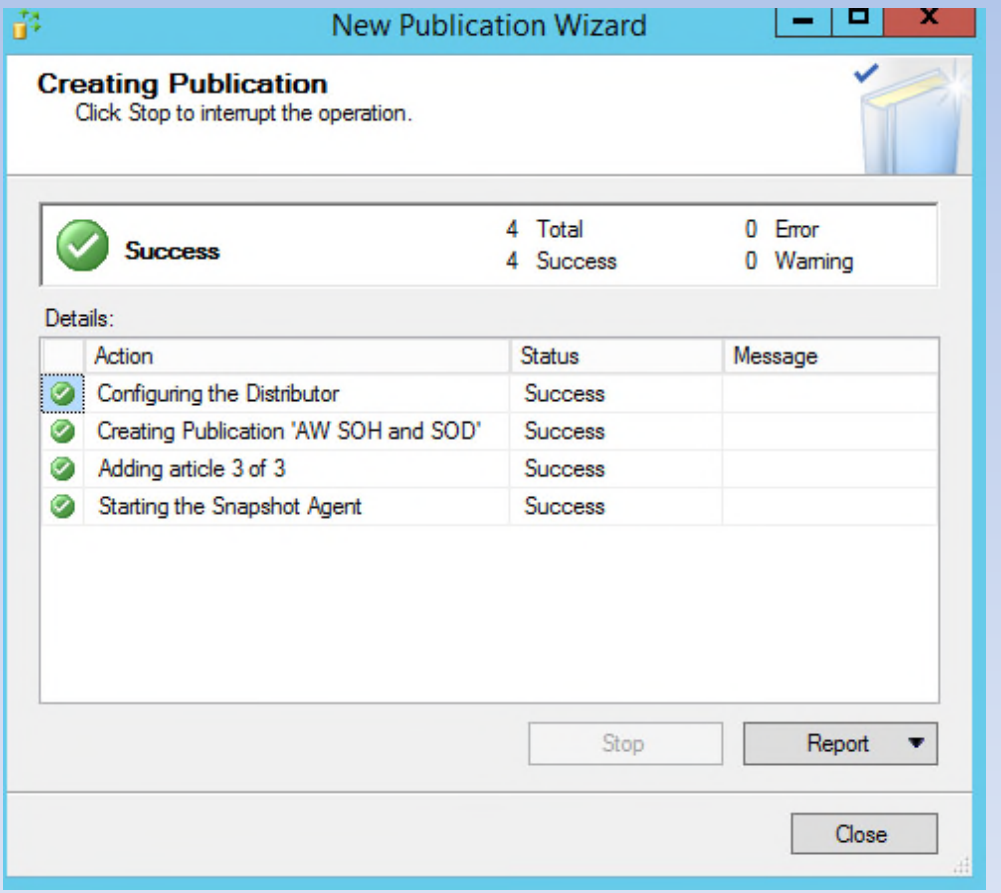


Demo: Creating the Publication

The actions the Wizard will perform are listed.

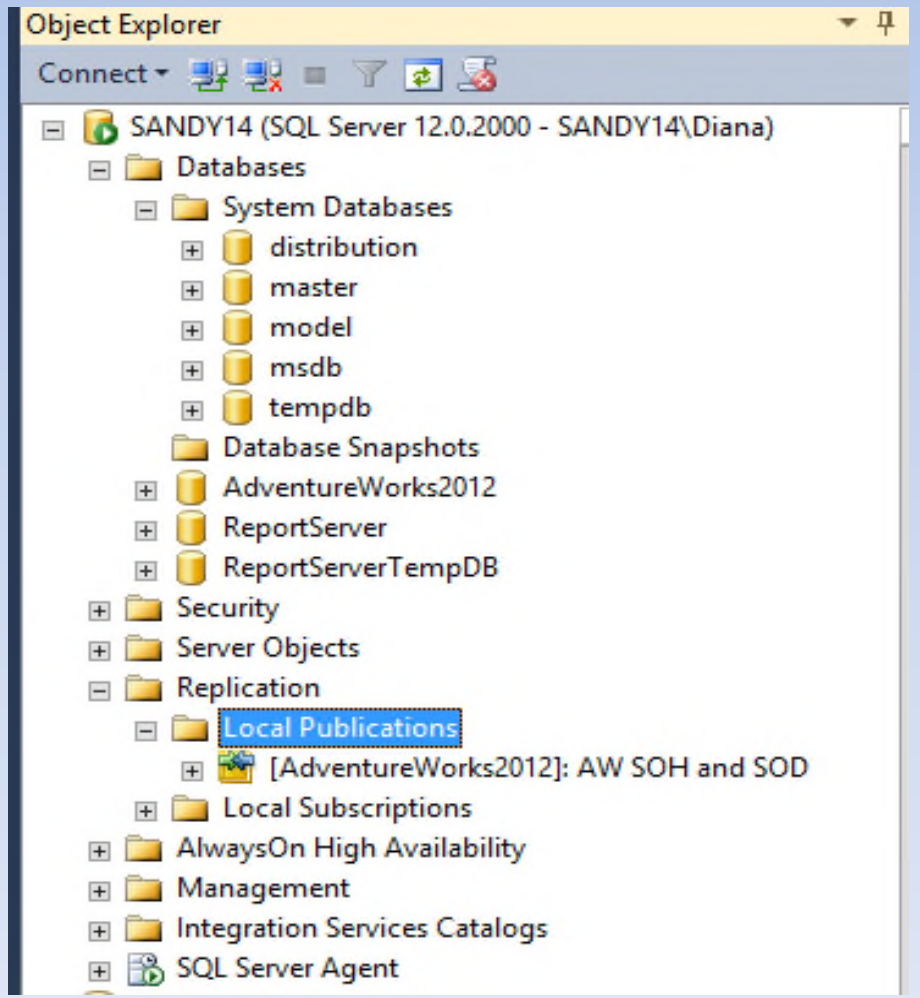


The publication is successfully created.

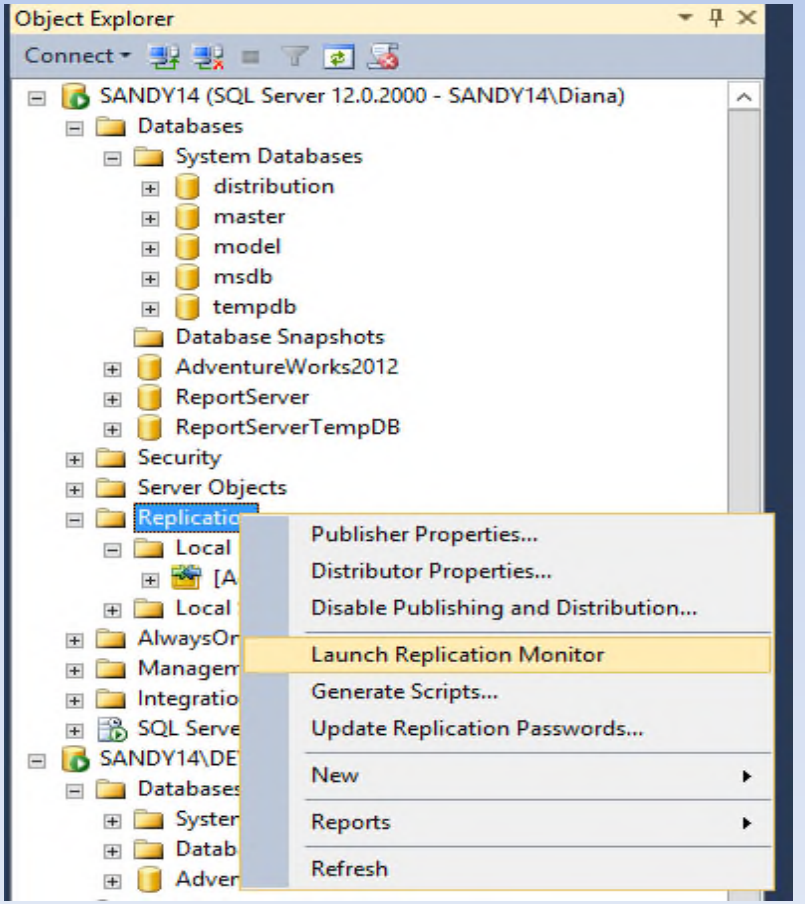


Demo: Creating the Publication

Object Explorer shows the publication.

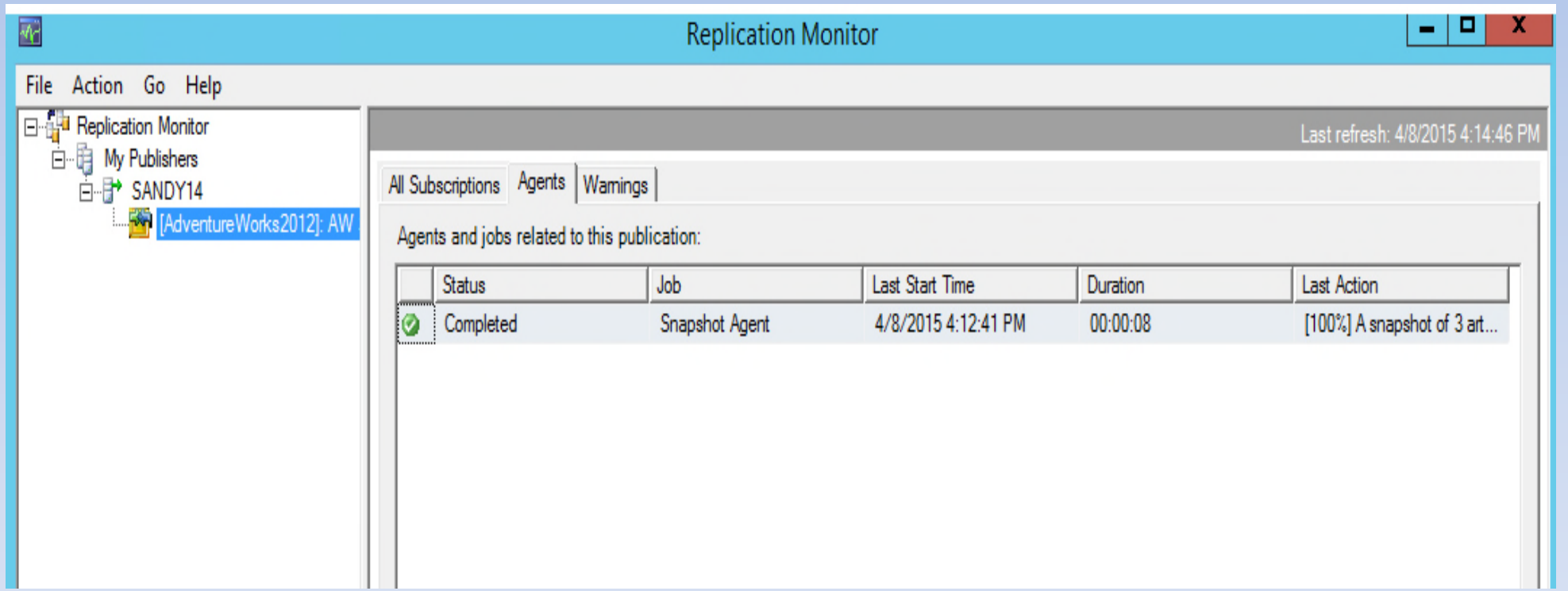


Right-click **Replication** and select **Launch Replication Monitor** to see if there are any problems.



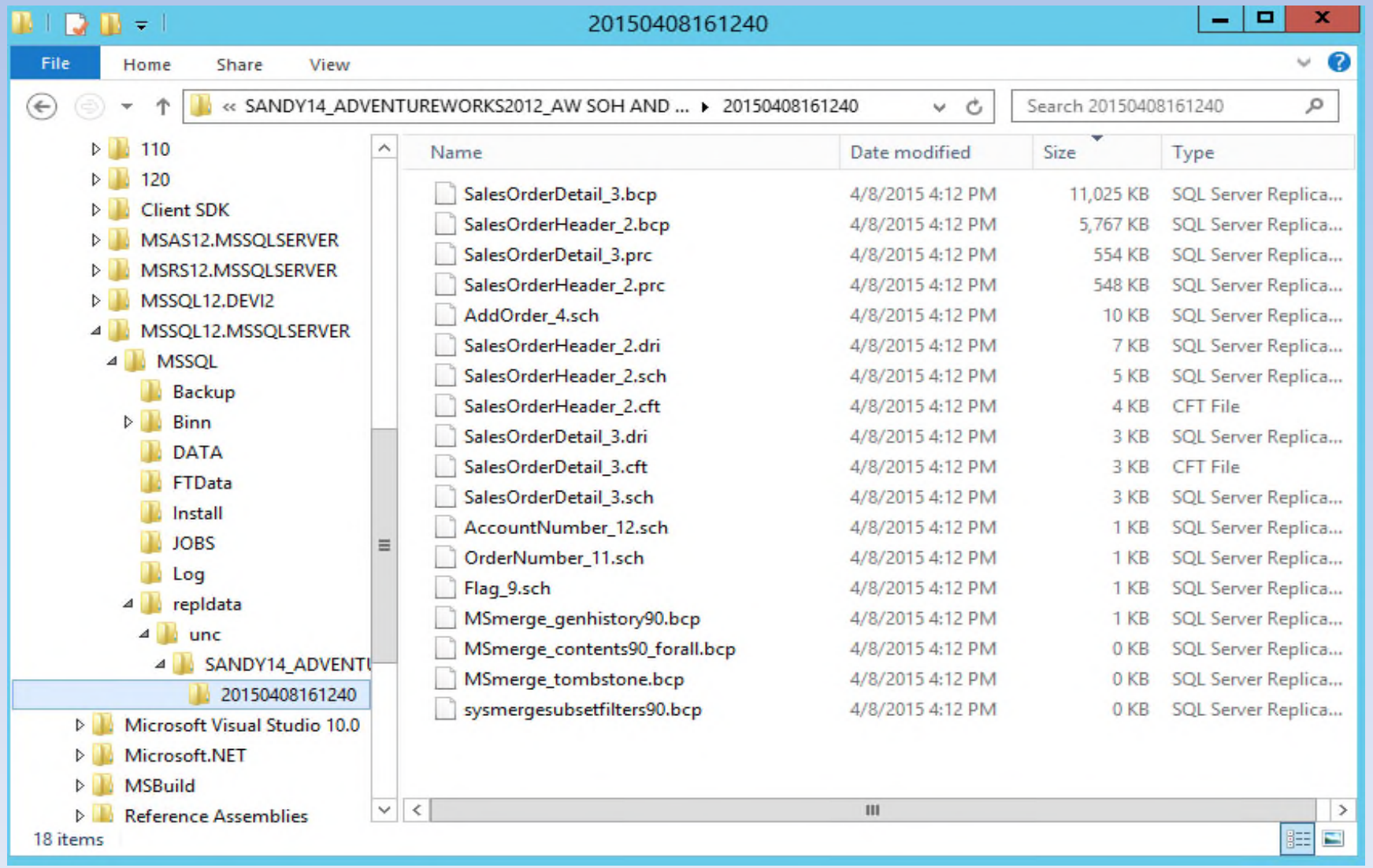
Demo: Creating the Publication

Replication Monitor shows the snapshot was created.



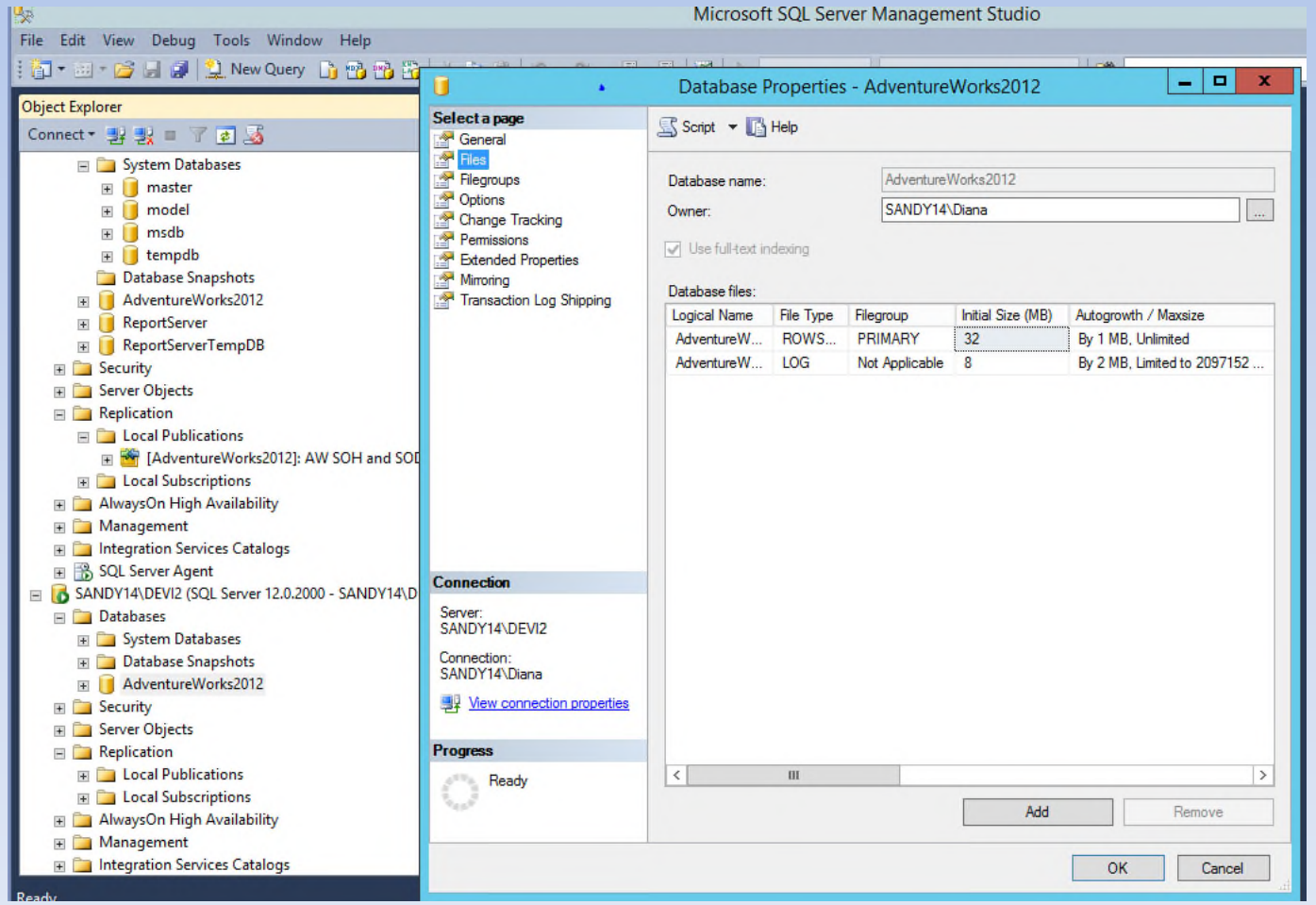
Demo: Creating the Publication, concl.

These are the snapshot files in the **repldata** folder.



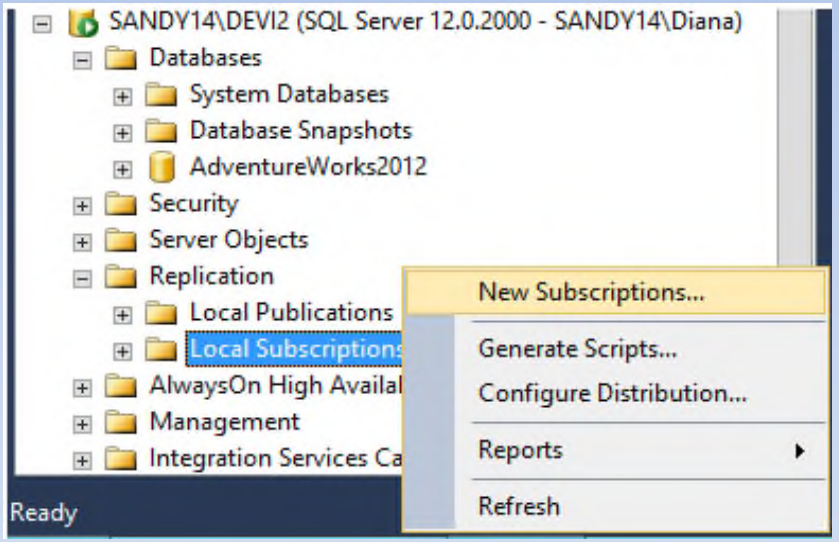
Demo: Creating the Subscription

A database with the same name, AdventureWorks2012, was created on the SQL Server instance, to hold the subscription.

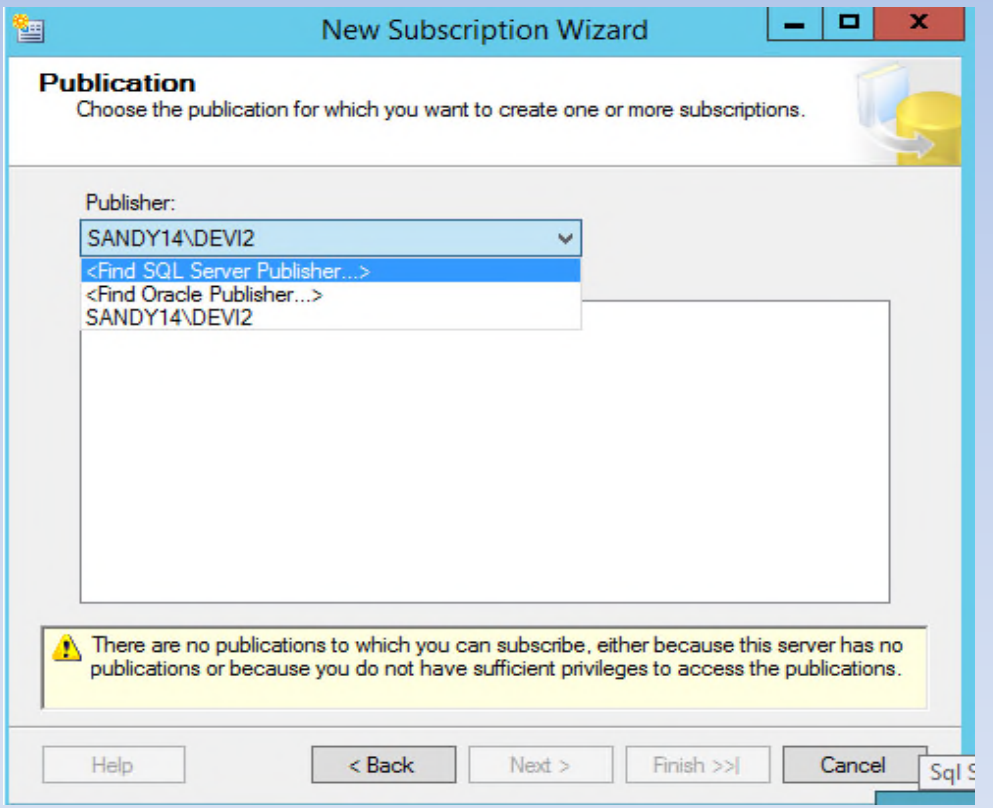


Demo: Creating the Subscription

On the subscription instance, start the **New Subscription** Wizard



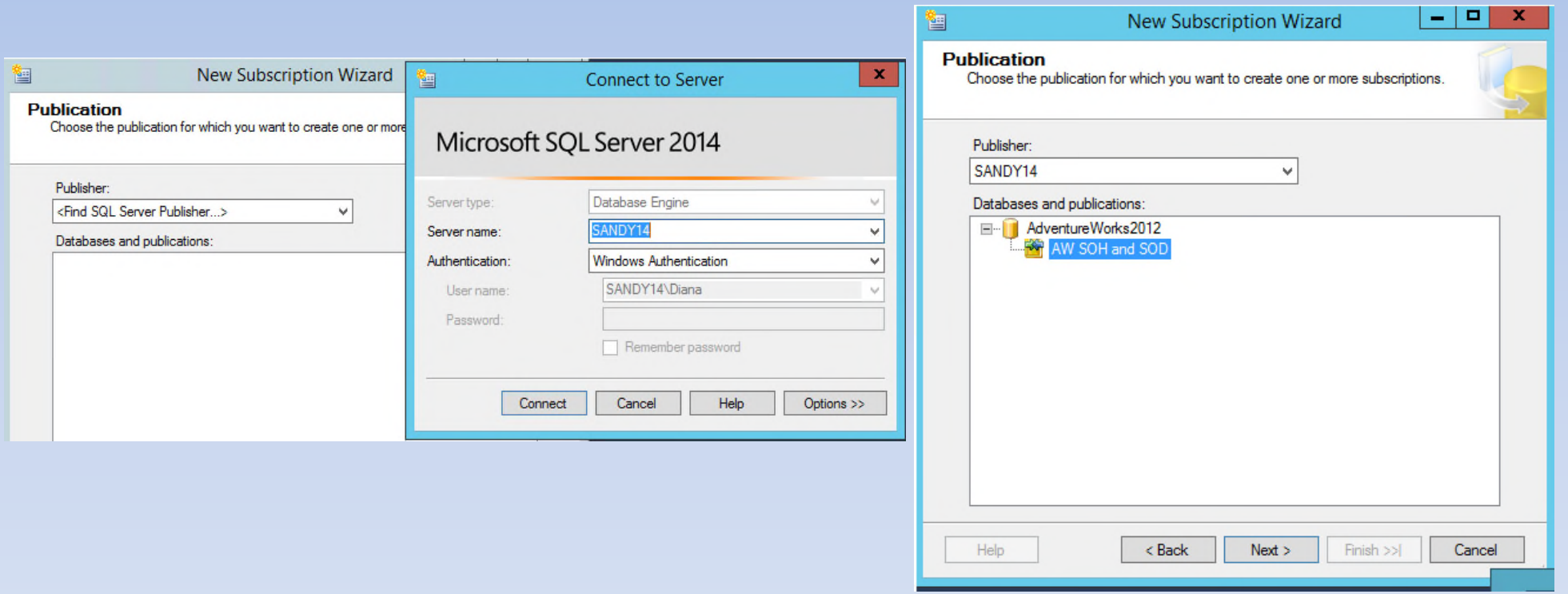
Find the SQL Server publisher.



Demo: Creating the Subscription

Log into (connect to) the publisher, and then …

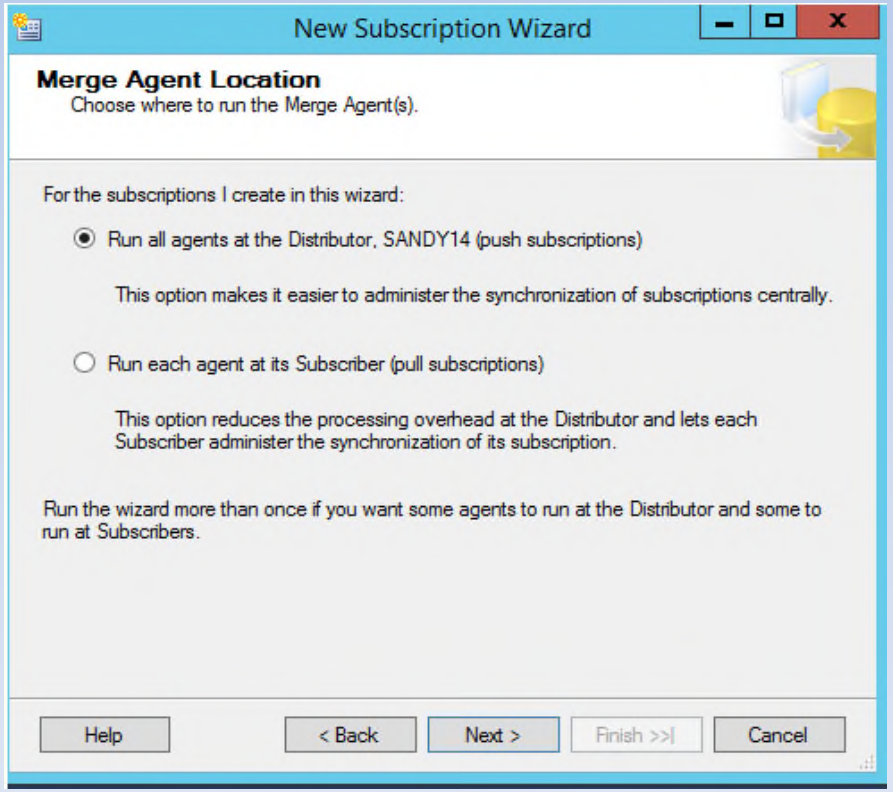
… select the publication.

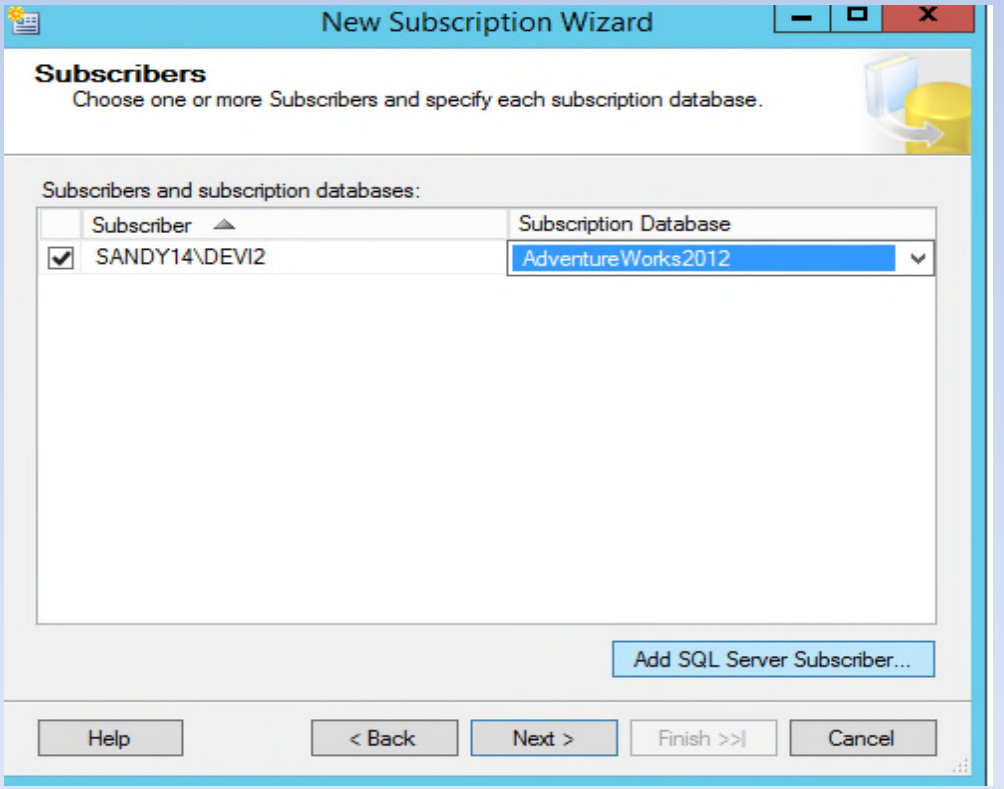


Demo: Creating the Subscription

Select either a push or a pull subscription. We chose push.

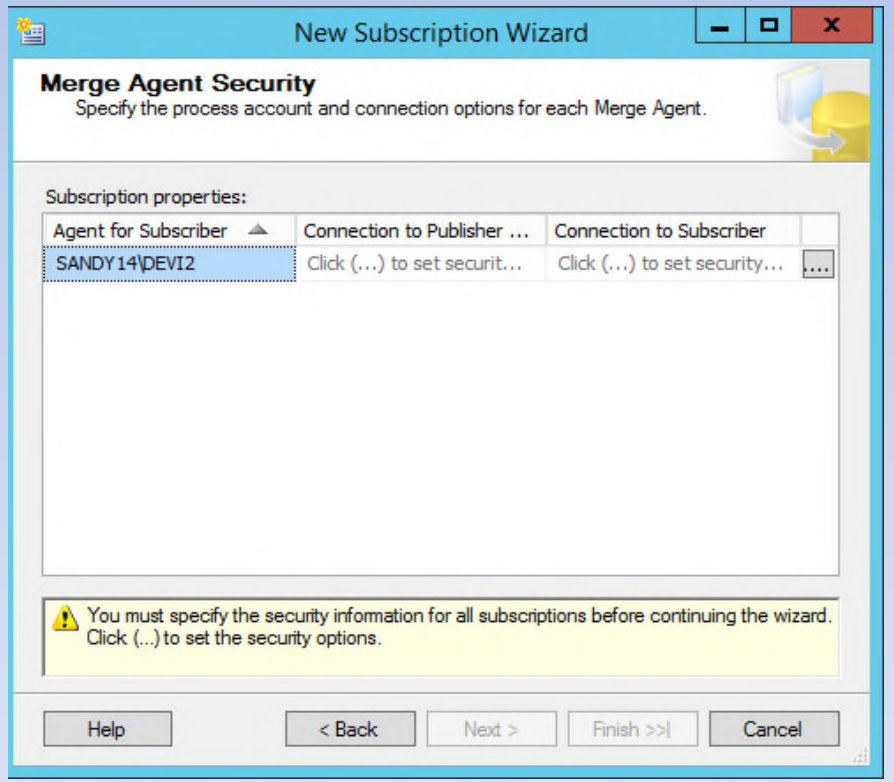
Select the subscriber database. You could create a new database at this time. You can also add other push subscribers.



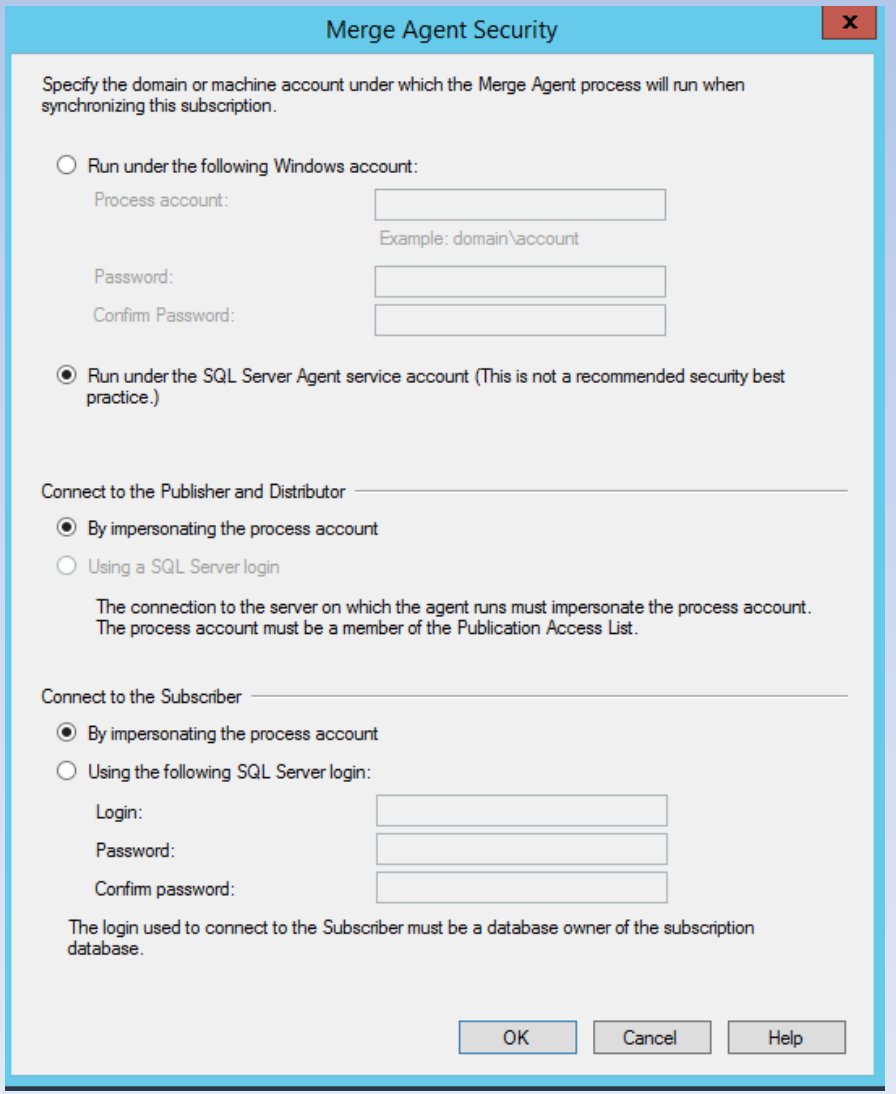


Demo: Creating the Subscription

Set the merge agent account(s).



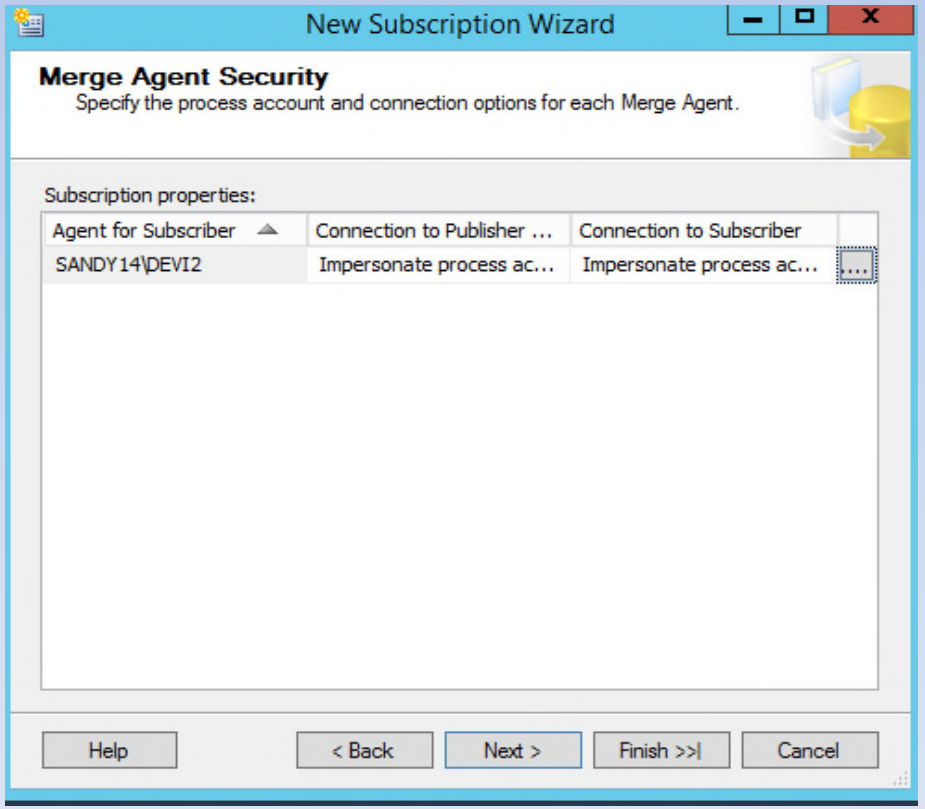
Again, we chose to use the SQL Agent account.

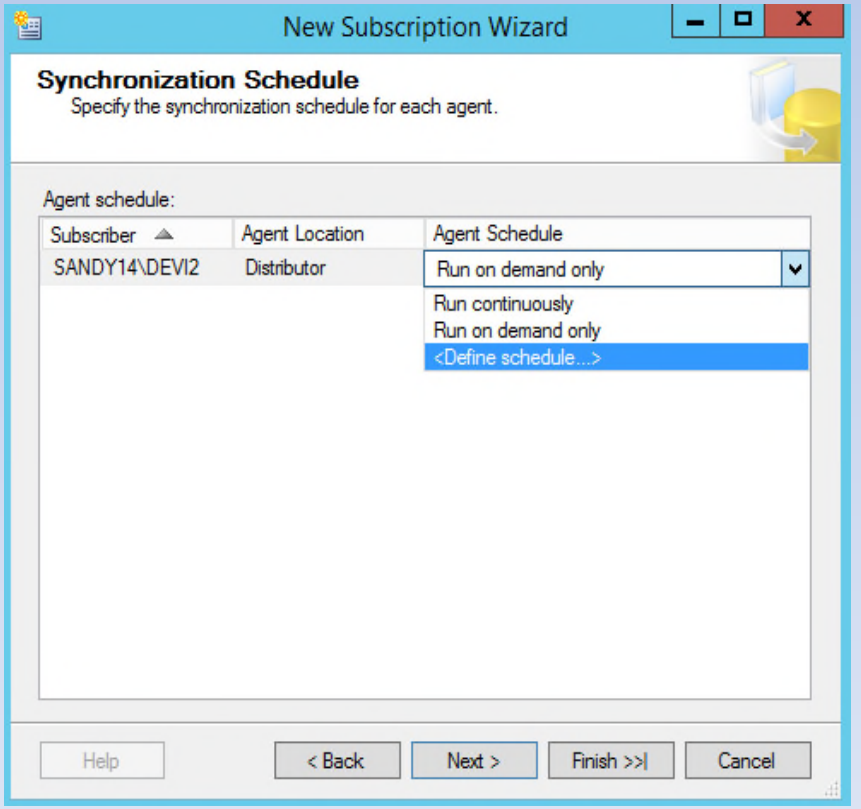


Demo: Creating the Subscription

The Security dialog displays our choices.

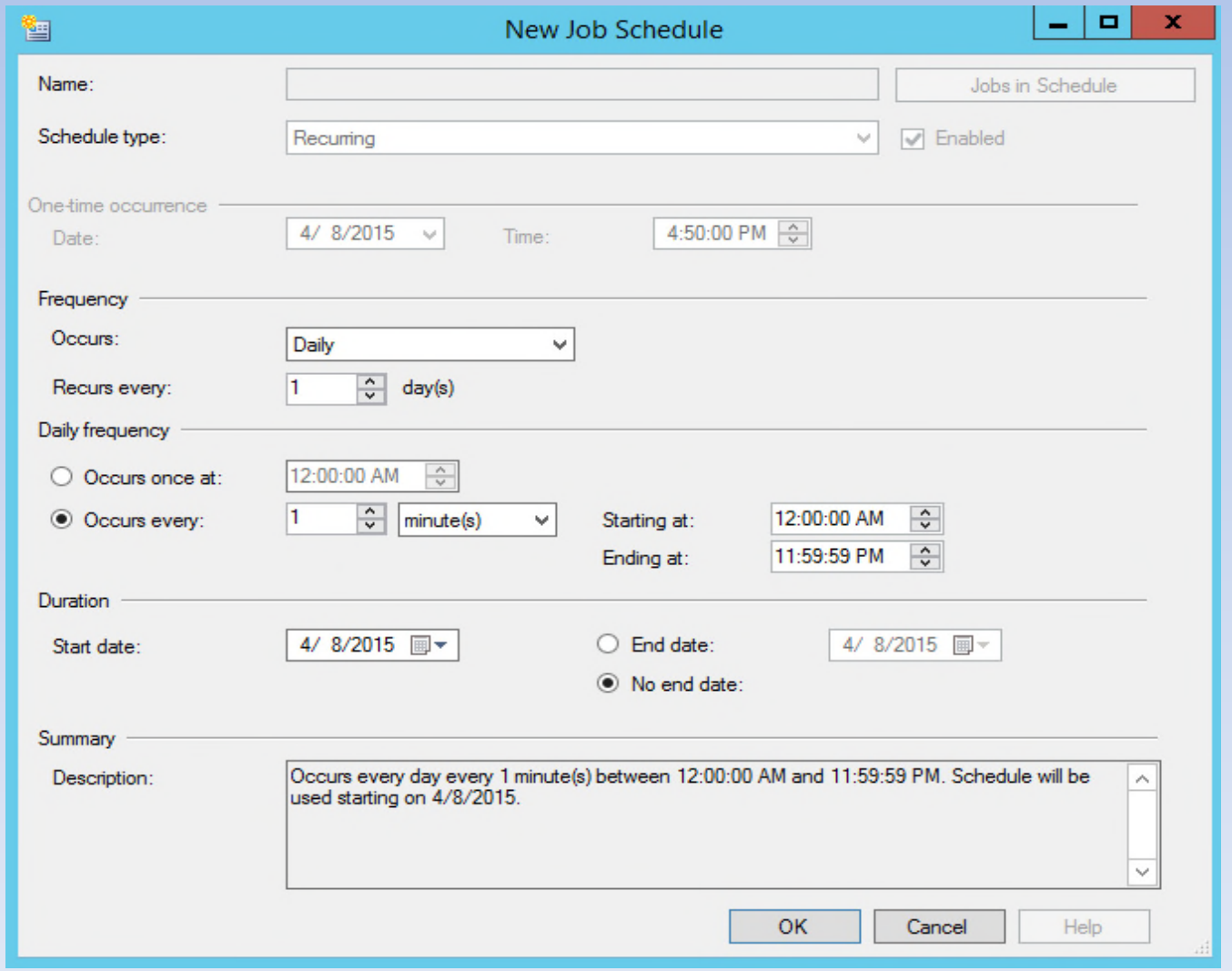
Next, we choose a schedule. **Run Continuously** is recommended. We chose a schedule so we could demonstrate conflicts.





Demo: Creating the Subscription

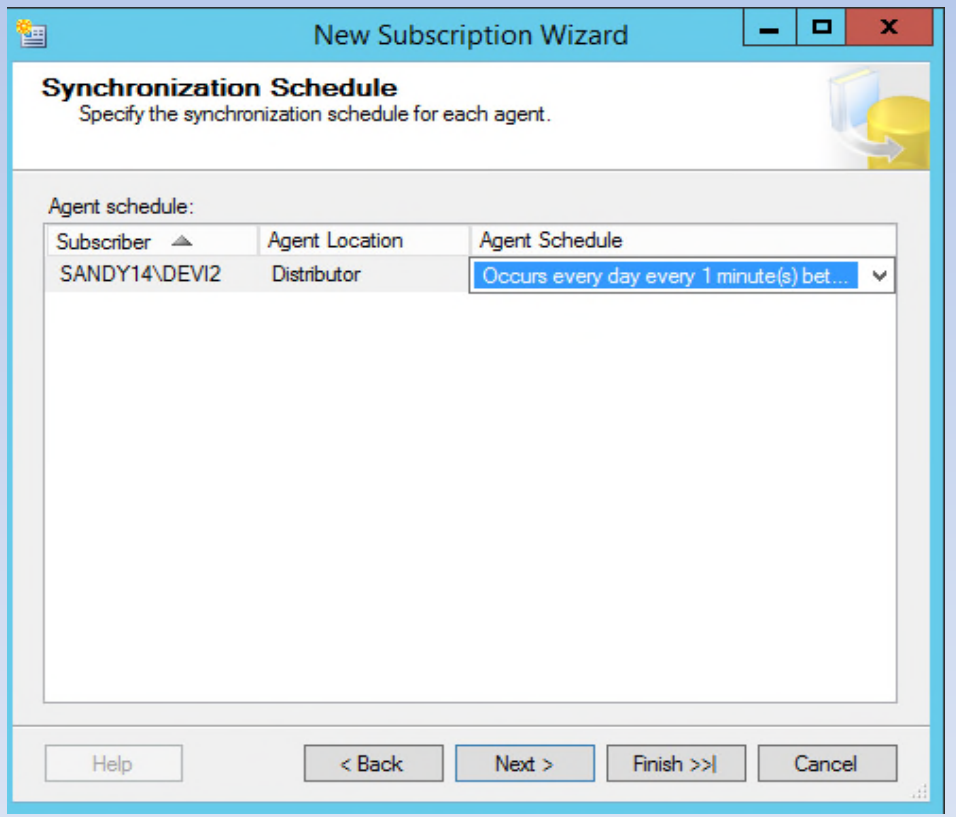
The “every one minute” schedule will allow us to demo replication conflicts. Most people choose “Run Continuously”

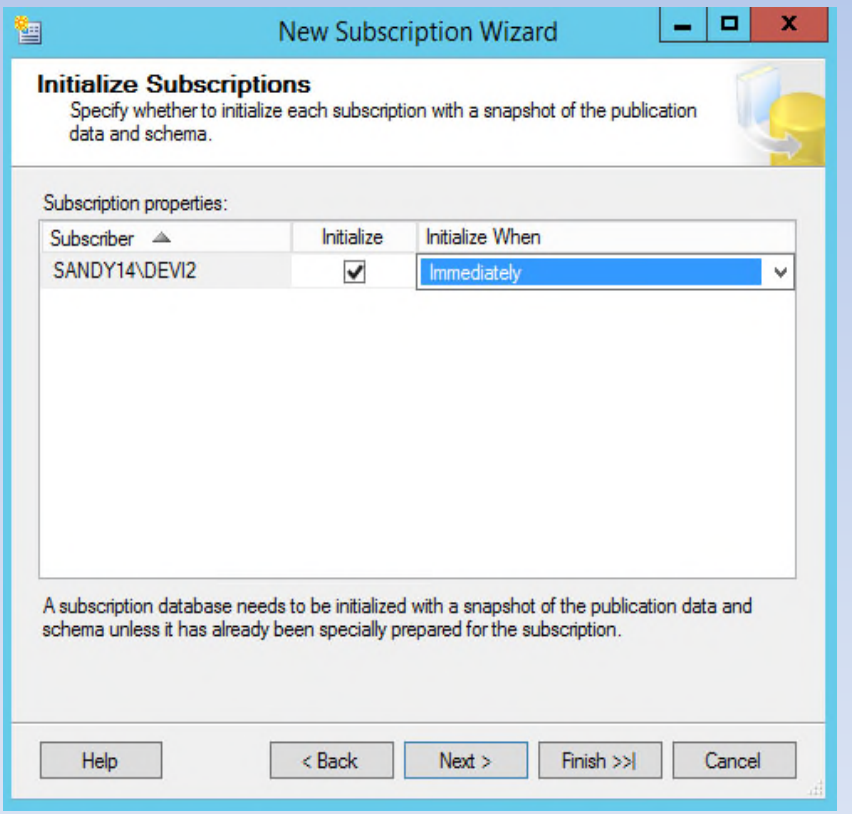


Demo: Creating the Subscription

The Schedule dialog shows the replicating agent schedule.

We chose to initialize the subscription (apply the snapshot) immediately.



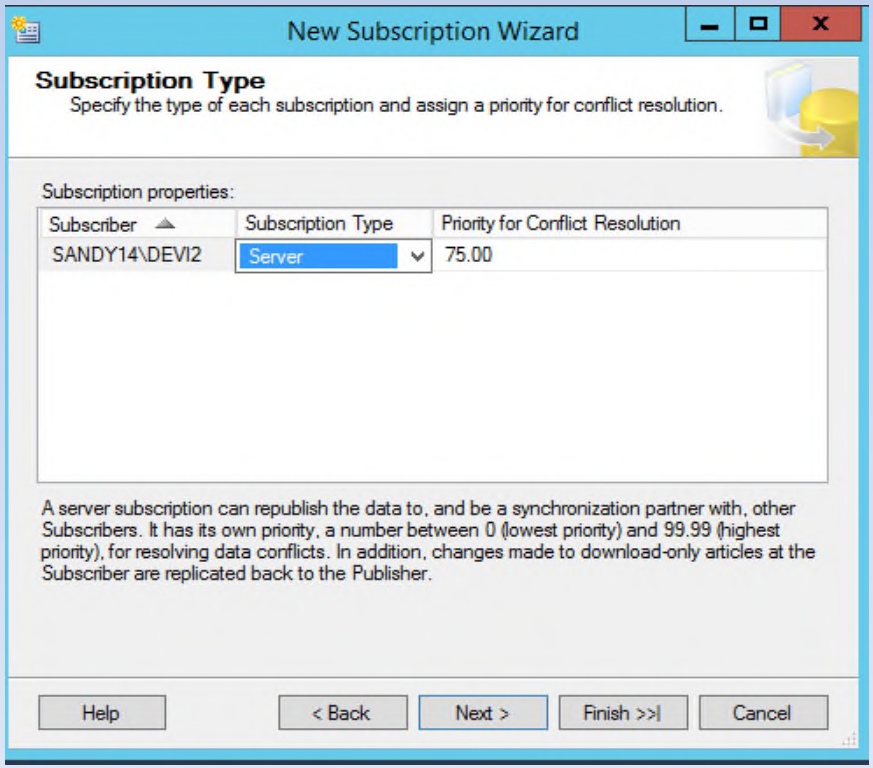


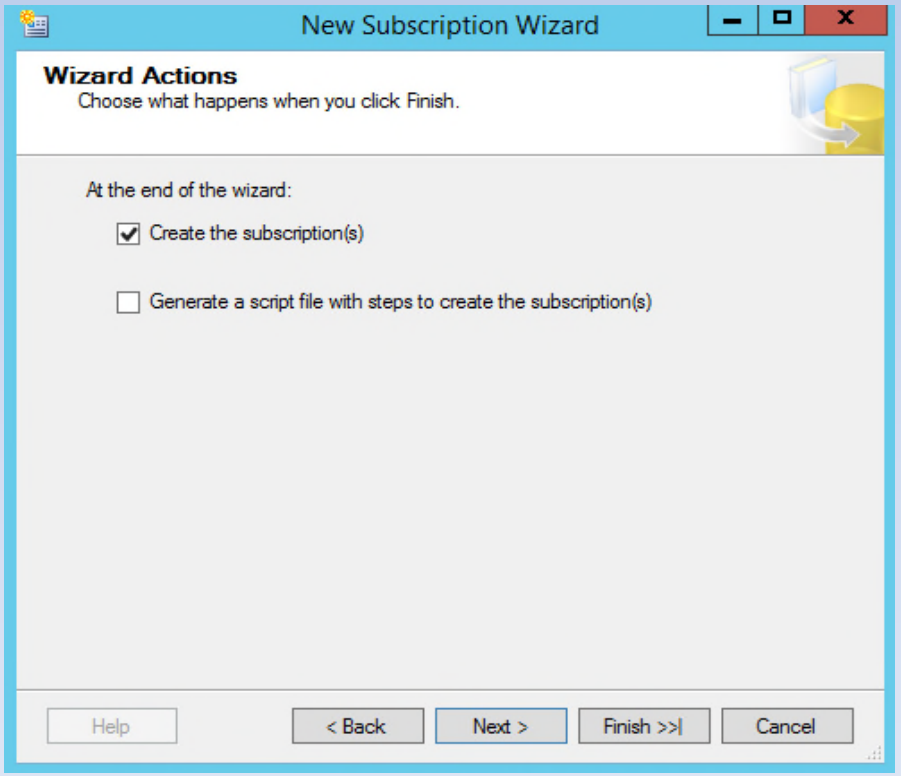
Demo: Creating the Subscription

Subscription Type may be

- Server (May assign priority values to individual subscribers. Publisher priority is 100.)
- Client (First change written to Publisher wins.

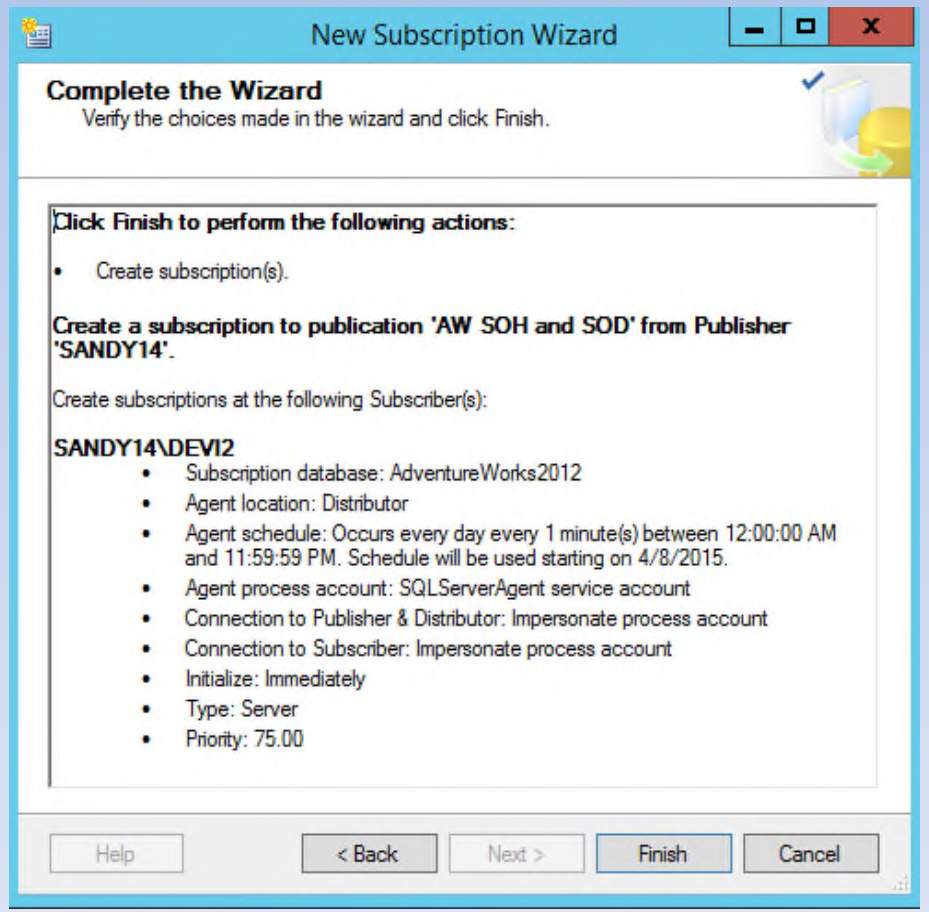
We chose to create the subscription.



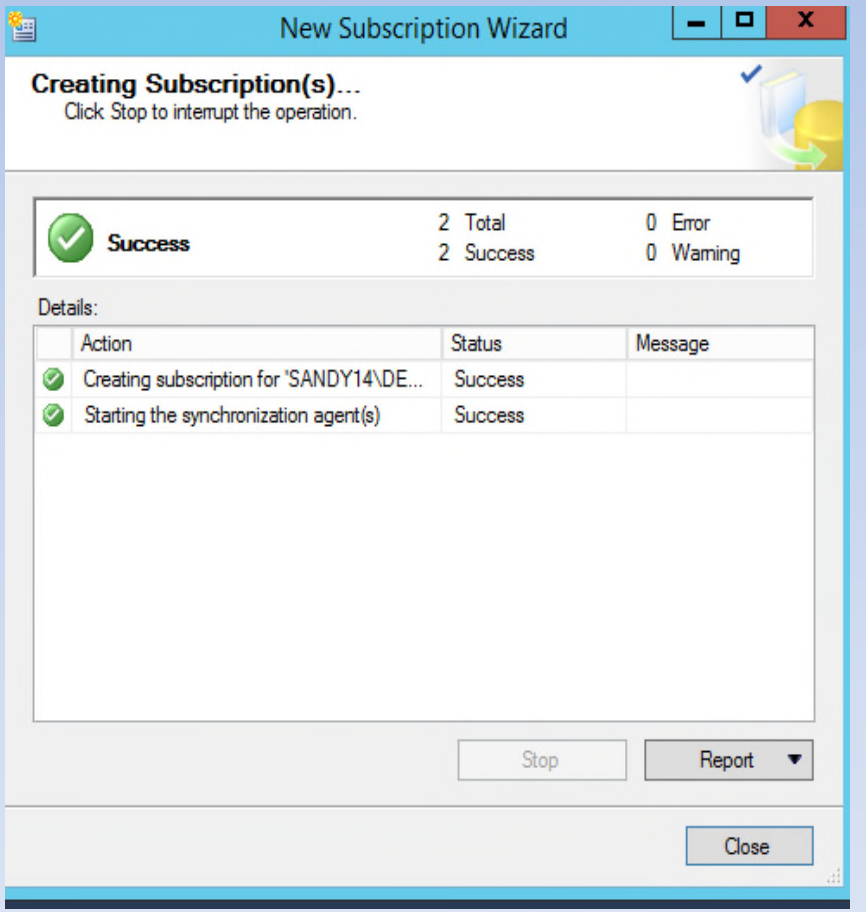


Demo: Creating the Subscription

The actions to be performed by the Wizard are listed.

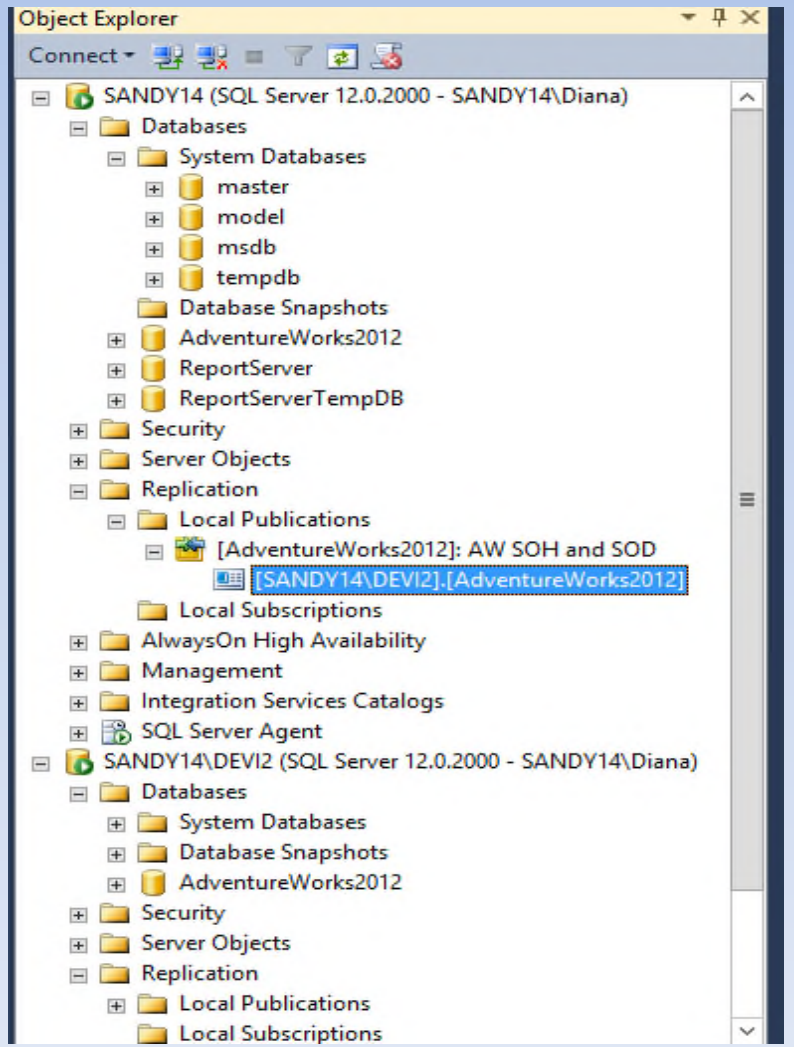


The Subscription is successfully created.



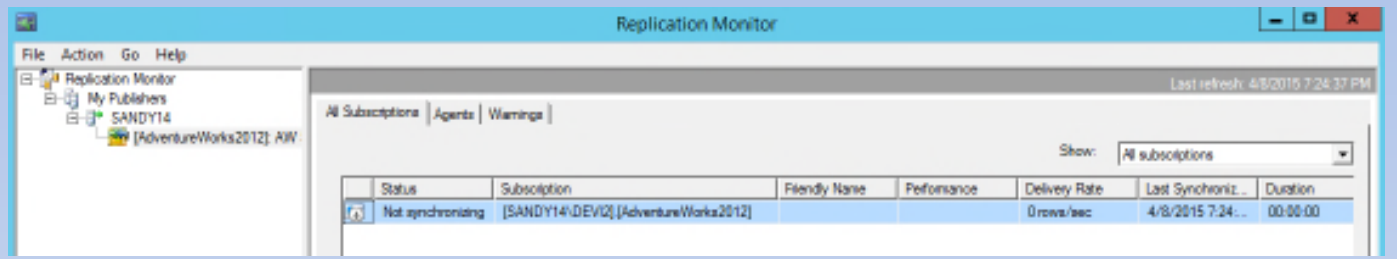
Demo: Creating the Subscription

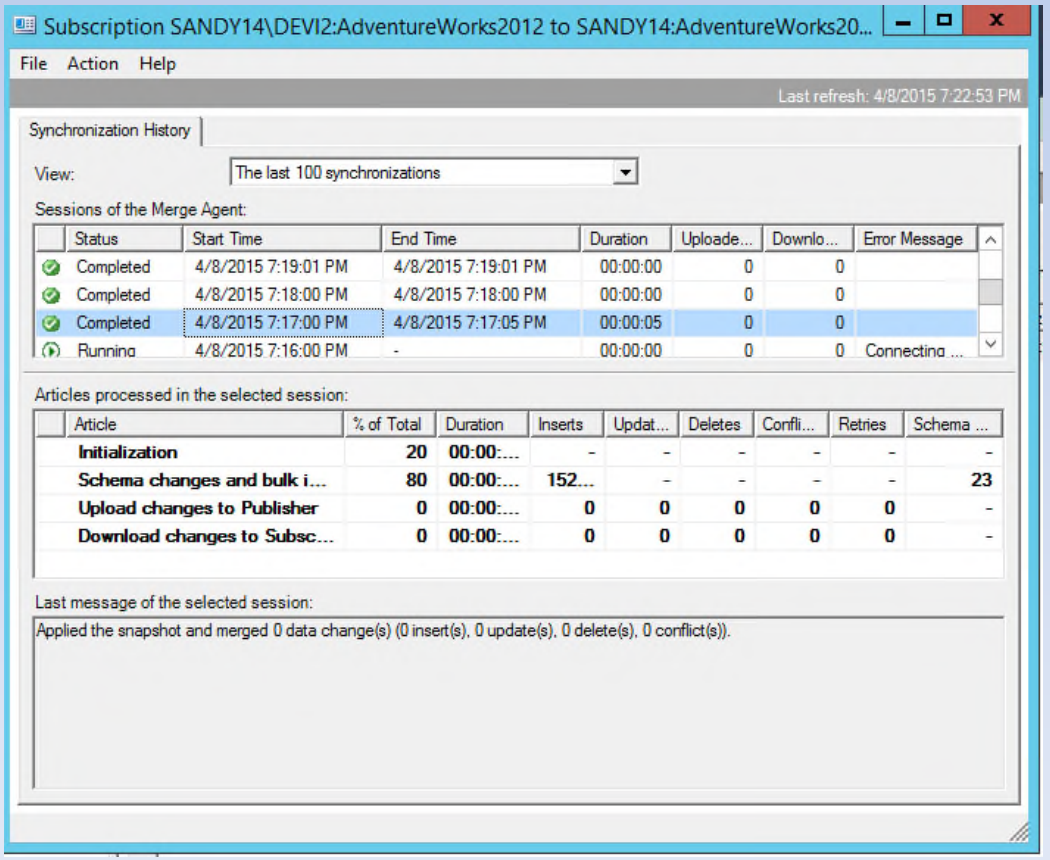
The push subscription shows on the Publisher instance. (A pull subscription would show on the Subscriber instance.)



Demo: Creating the Subscription

Replication Monitor shows that the Subscriber has been synchronized. (Double-click on the Subscription to see the Synchronization History.

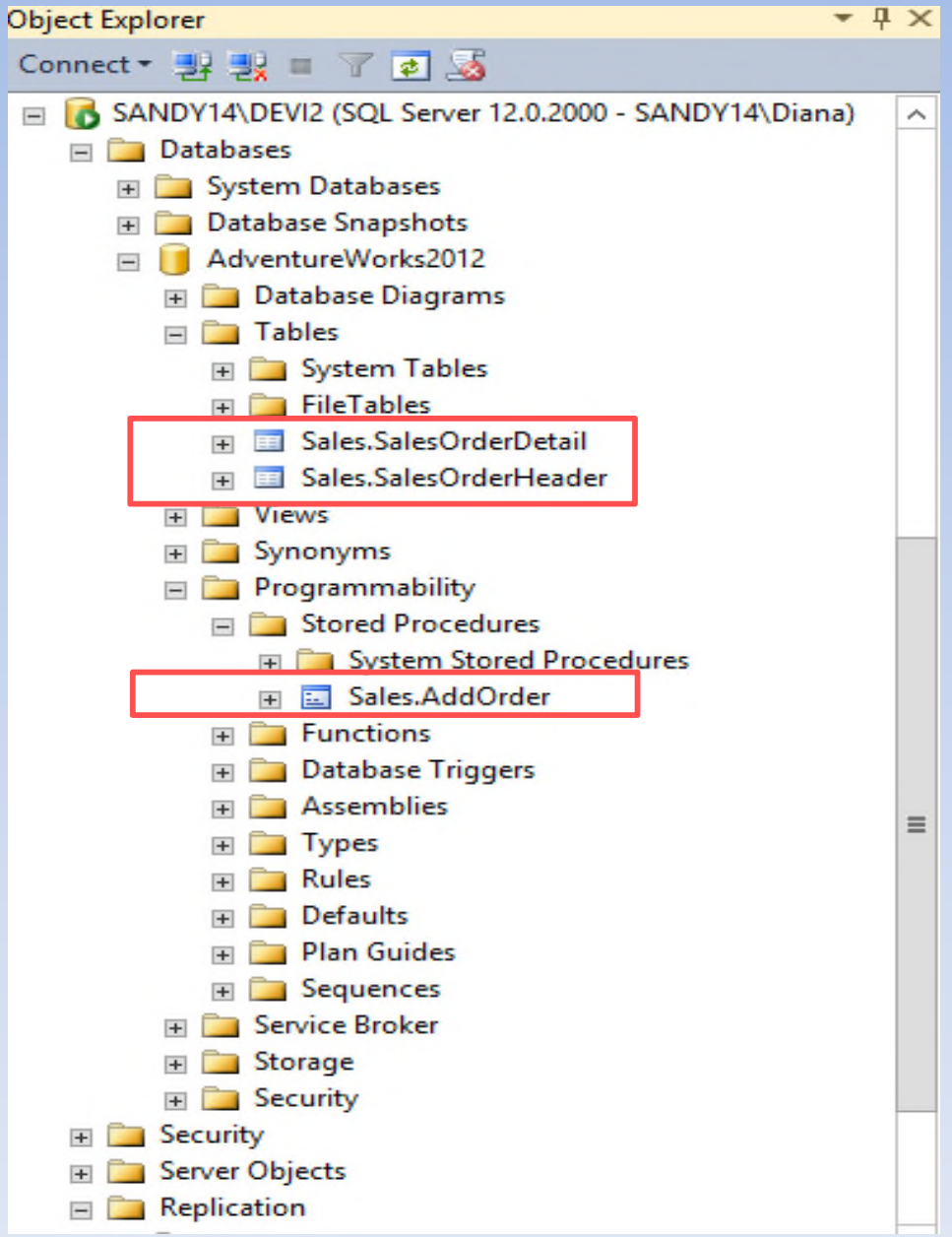




Demo: Creating the Subscription

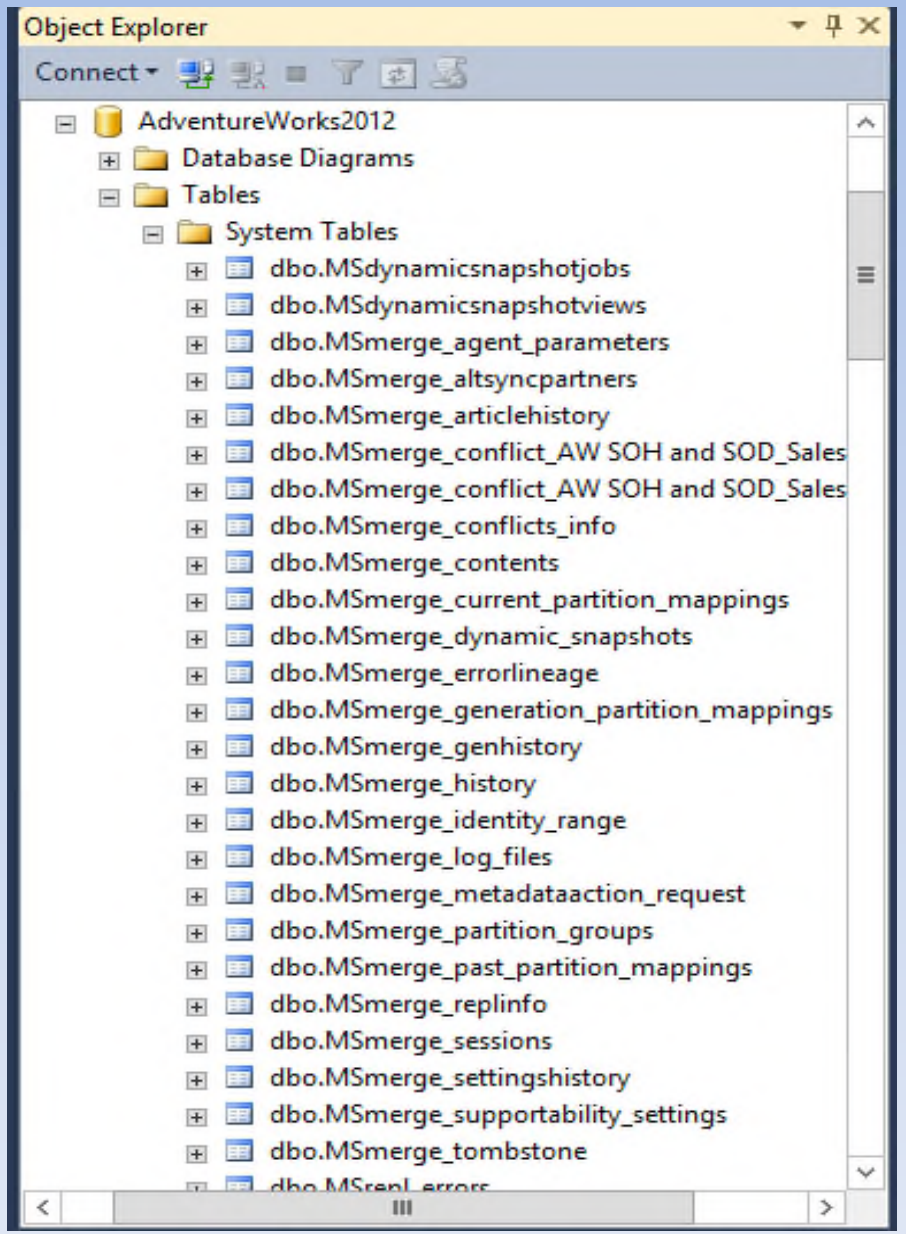
Object explorer shows that the Subscriber now has the replicated tables and stored procedure.

(User-defined data types used in SOH are also replicated.)



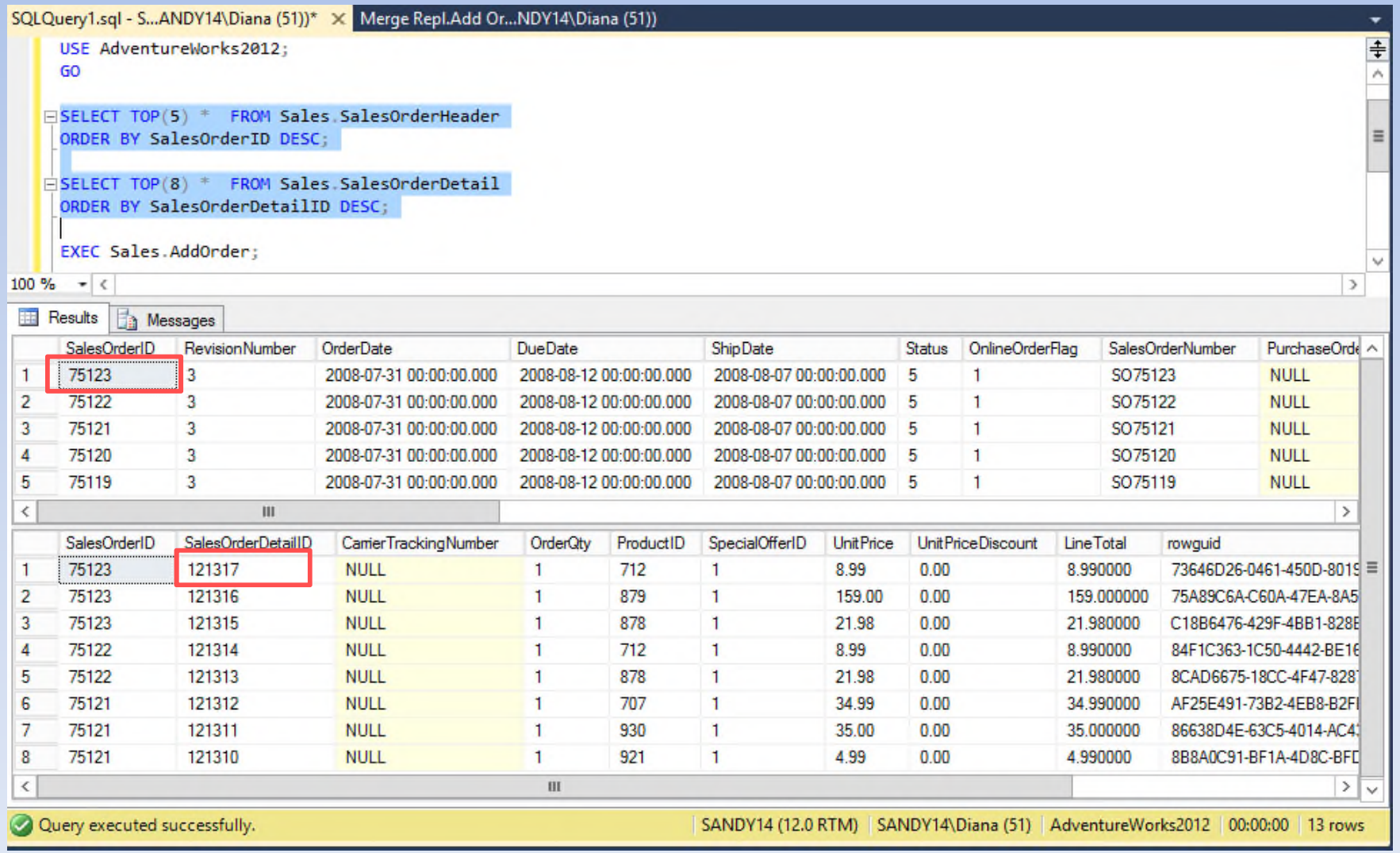
Demo: Creating the Subscription, concl.

Merge replication adds many system tables and triggers to track changes.



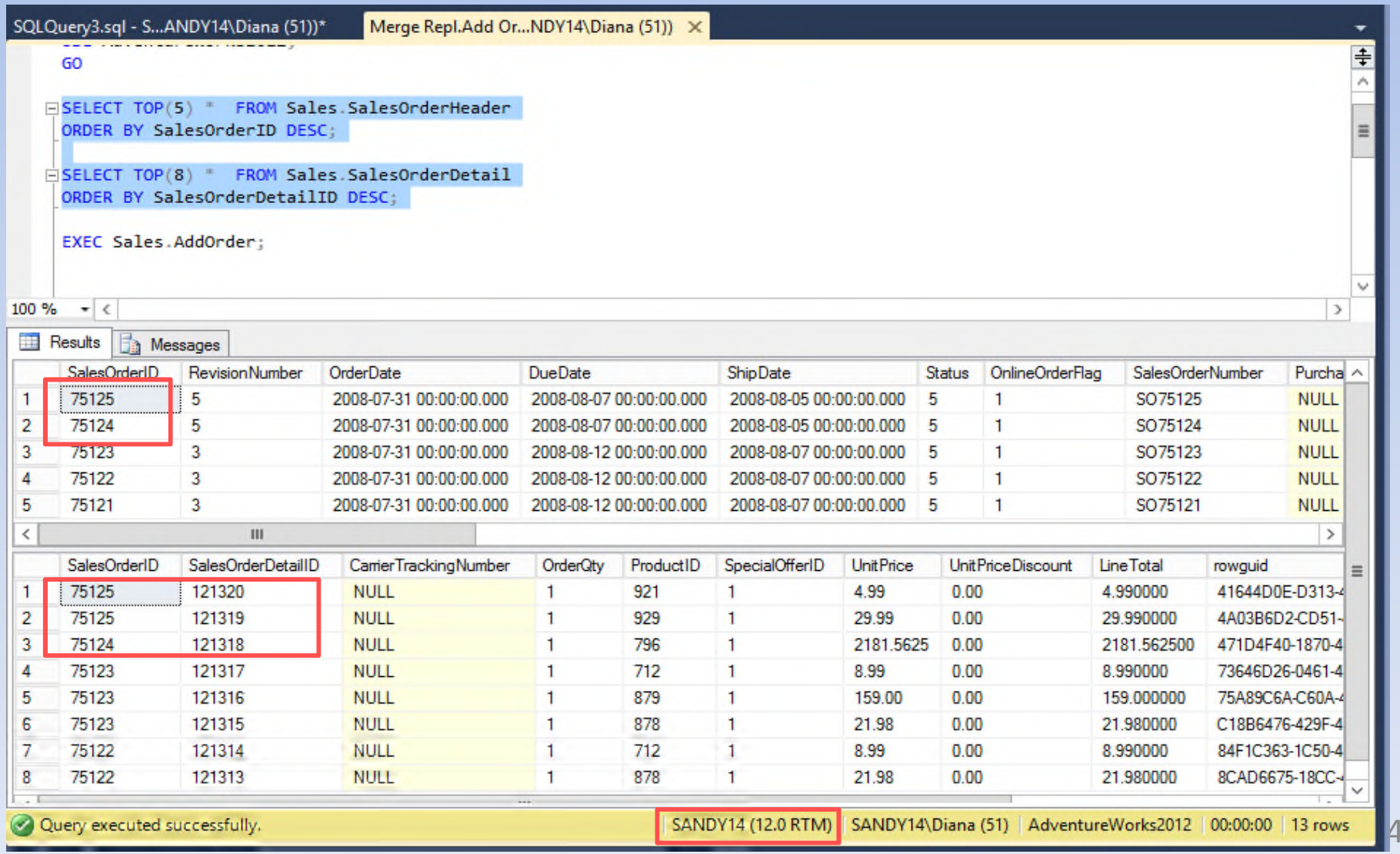
Demo: Replicating Data

Here are the latest rows in the SOH and SOD tables, before any additions are made. The max SalesOrderID IDENTITY value is 75123 and the max SalesOrderDetailID IDENTITY value is 121317.



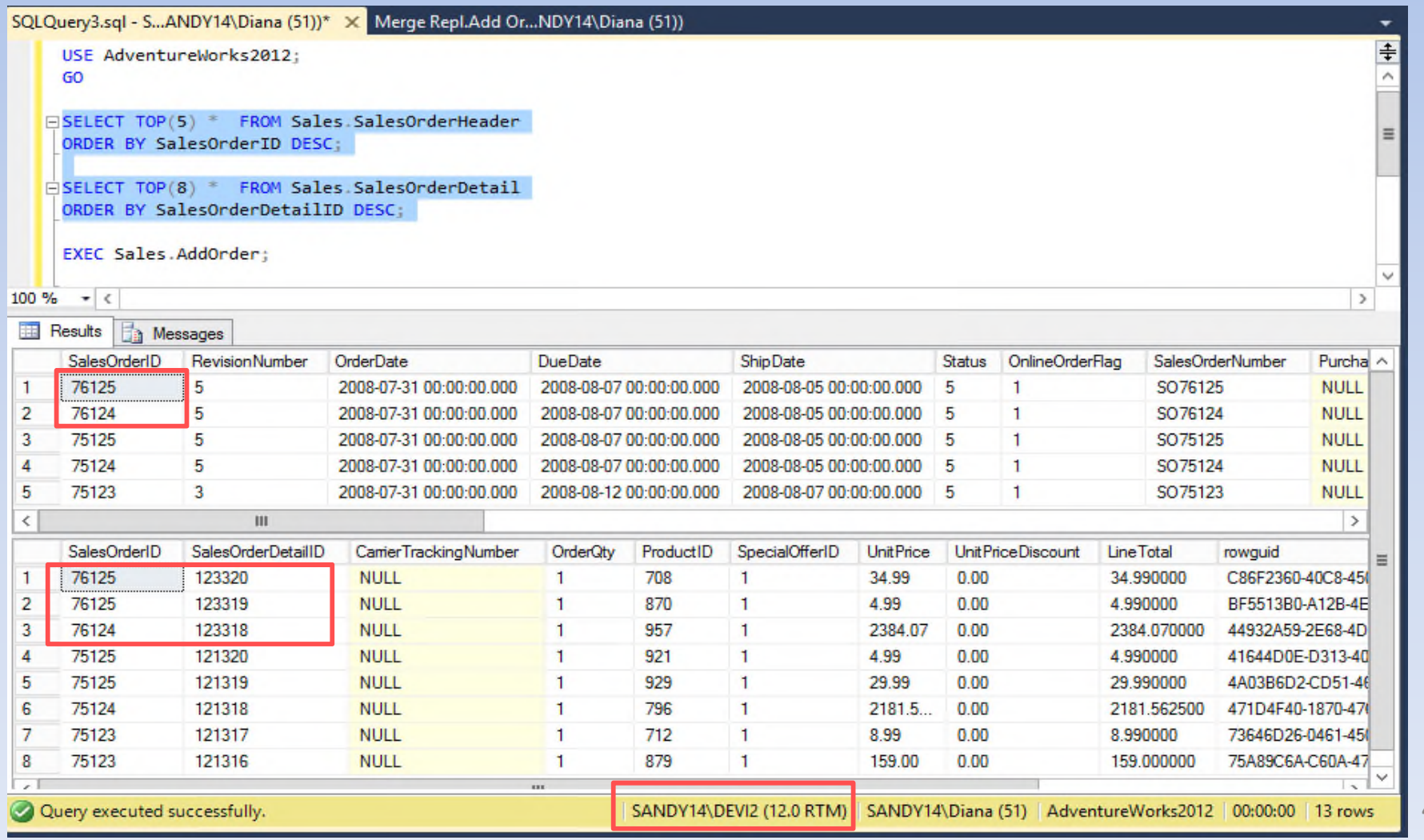
Demo: Replicating Data

After adding two orders on the Publisher, we see the SalesOrderID IDENTITY value of 75124 and 75125 for the added orders in SOH and some line items in the SOD table.



Demo: Replicating Data

On the Subscriber, we see the replicated orders. After adding two orders on the Subsciber, we see the SalesOrderID and SalesOrderHeaderID values assigned from the Subscriber's IDENTITY value ranges.

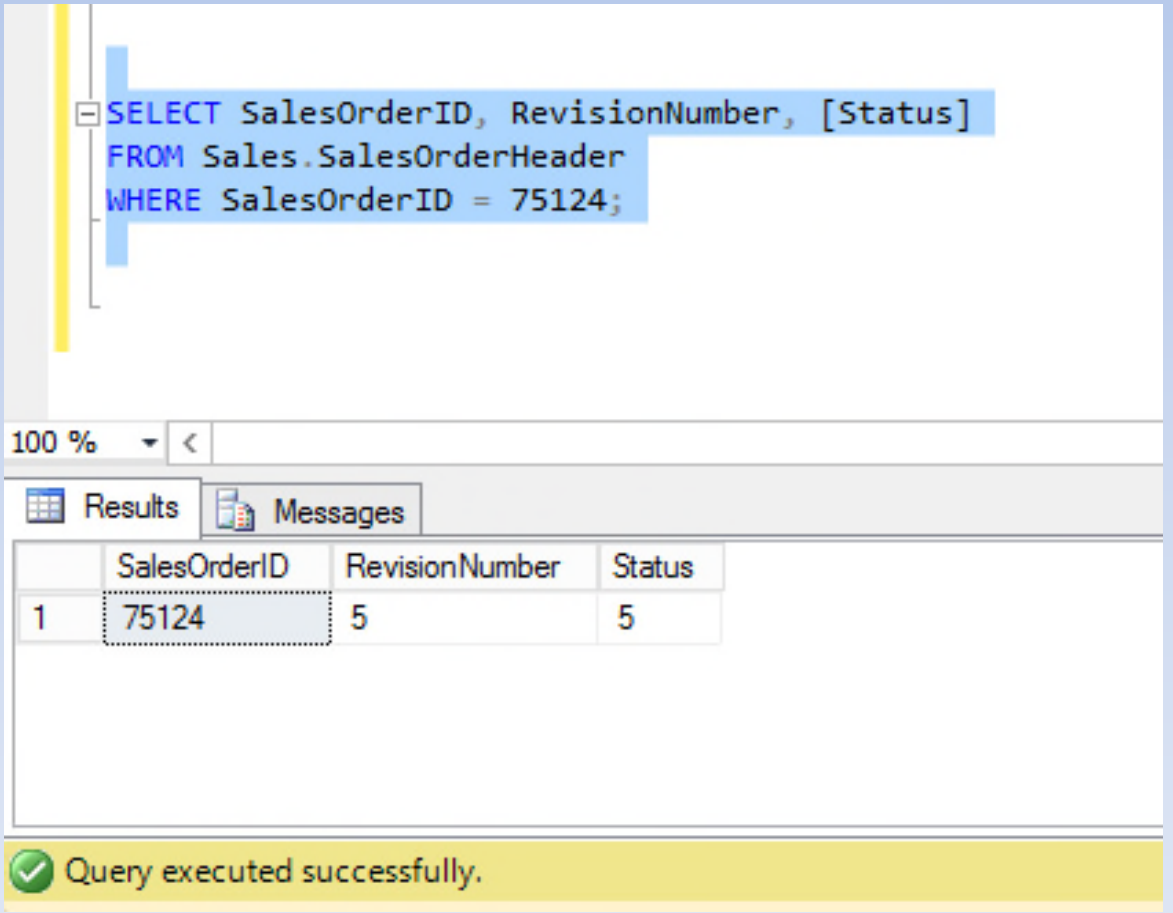


Demo: Conflict Resolution

- We have set conflict resolution to column-level tracking, which allows for update of different fields in the same row during the replication interval.
- If the replication is set to *immediate*, and there is no likelyhood of simultaneous update or delete targeting the same row on two instances, then row-level tracking should be chosen, as there will be fewer tracking tables required.
- We will show that with column-level tracking, updating two fields in the same row on different instances is not a conflict.
- Then we will update the same field on two instances, showing how the conflict is automatically resolved, and how the administrator can later change the conflict resolution.

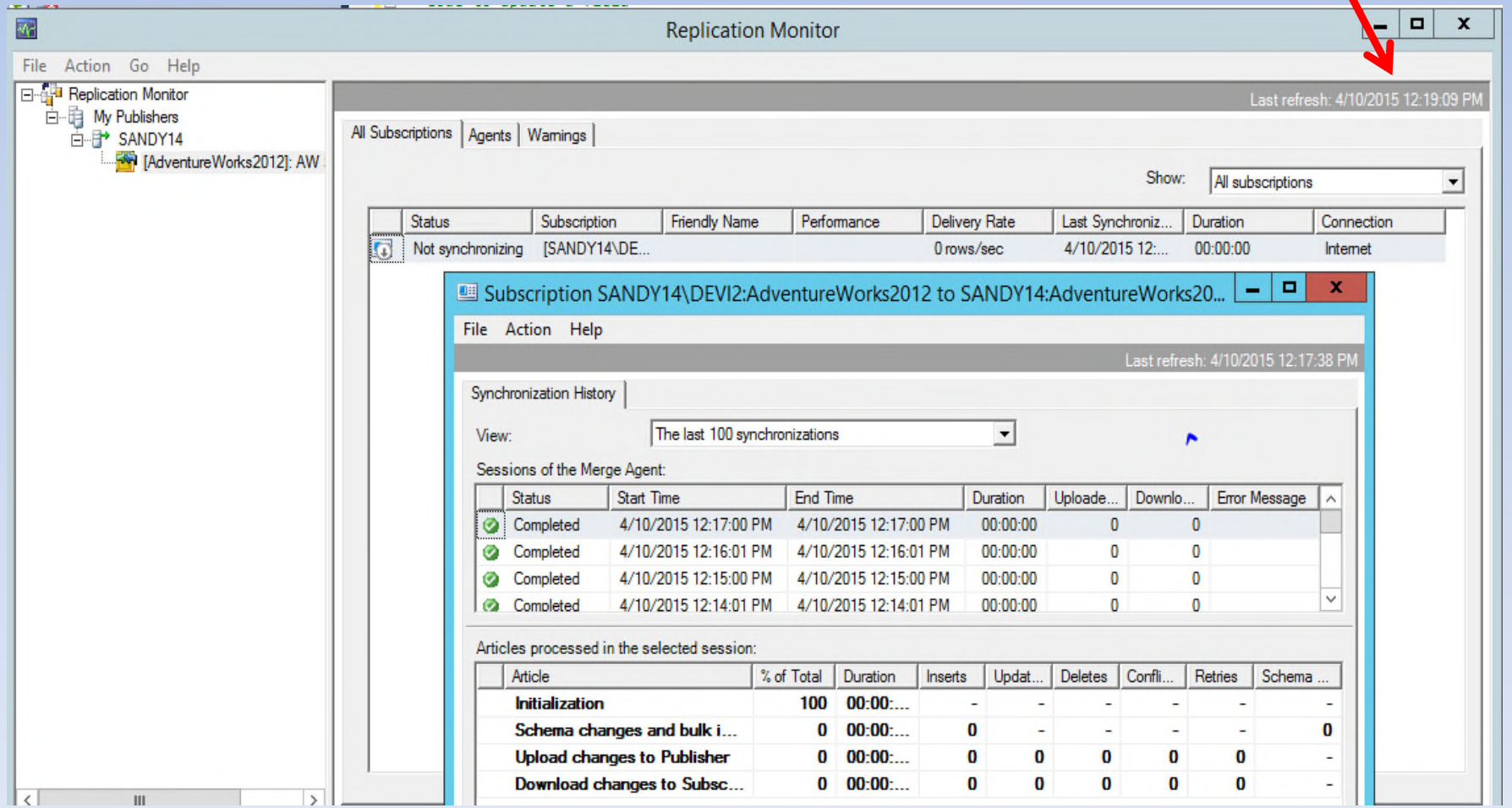
Demo: Conflict Resolution

Here are the values of RevisionNumber and Status in one of the rows in SOH:



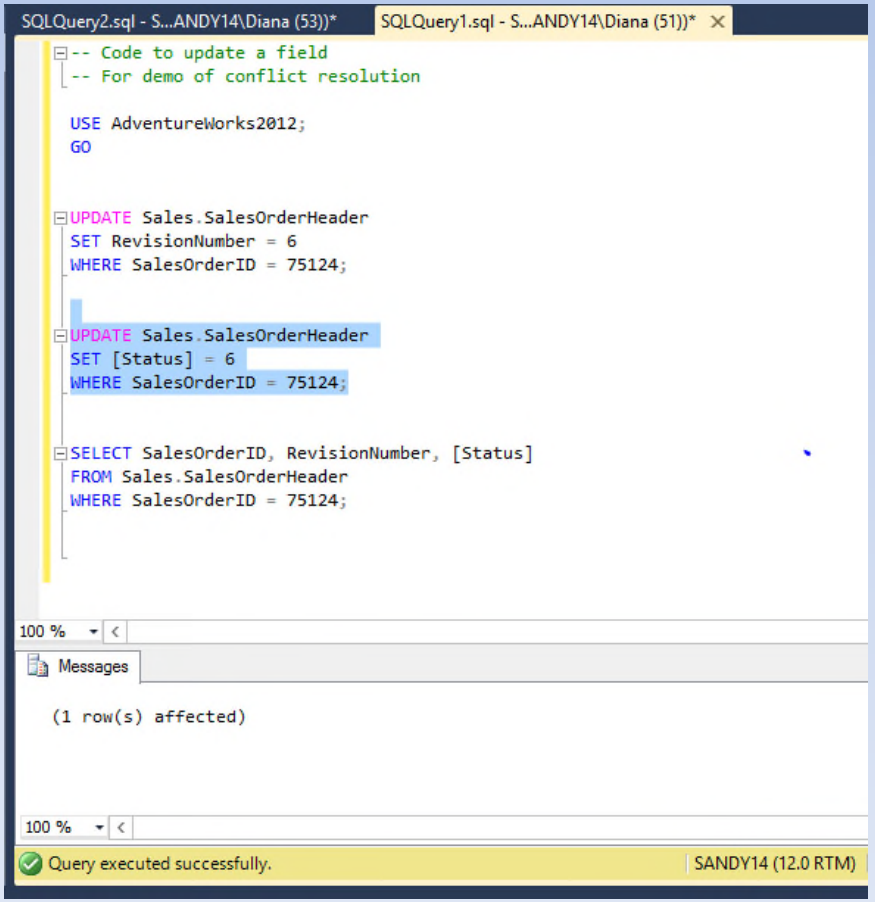
Demo: Conflict Resolution

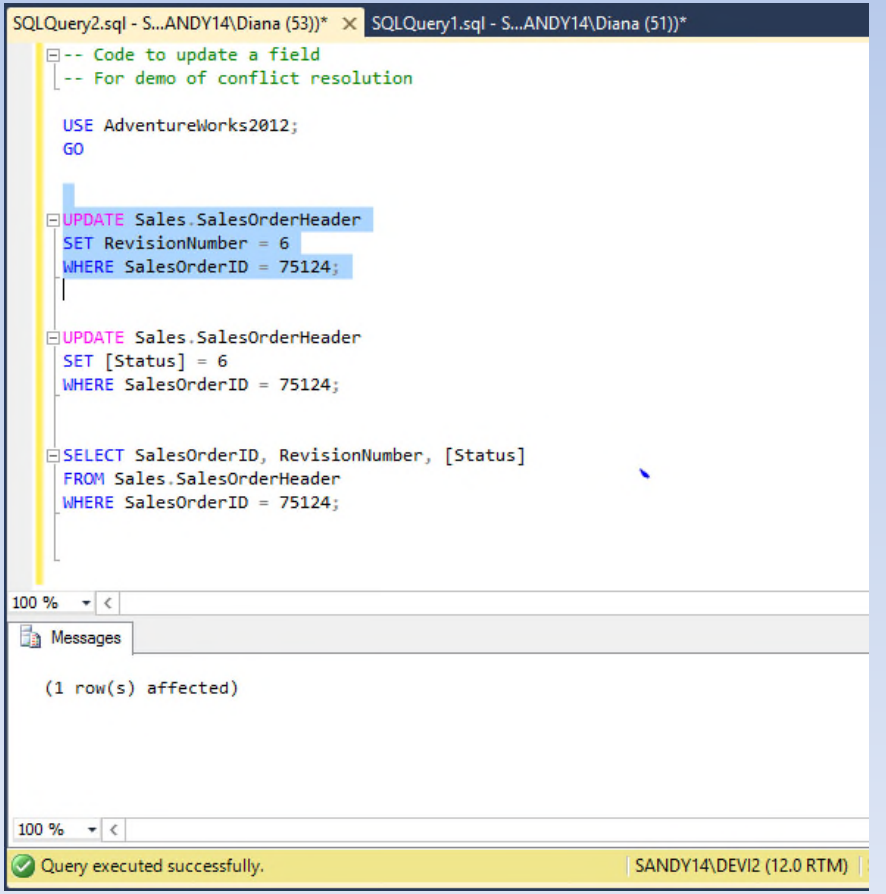
We wait for Replication Monitor to indicate that the one-minute replication interval is just beginning.



Demo: Conflict Resolution

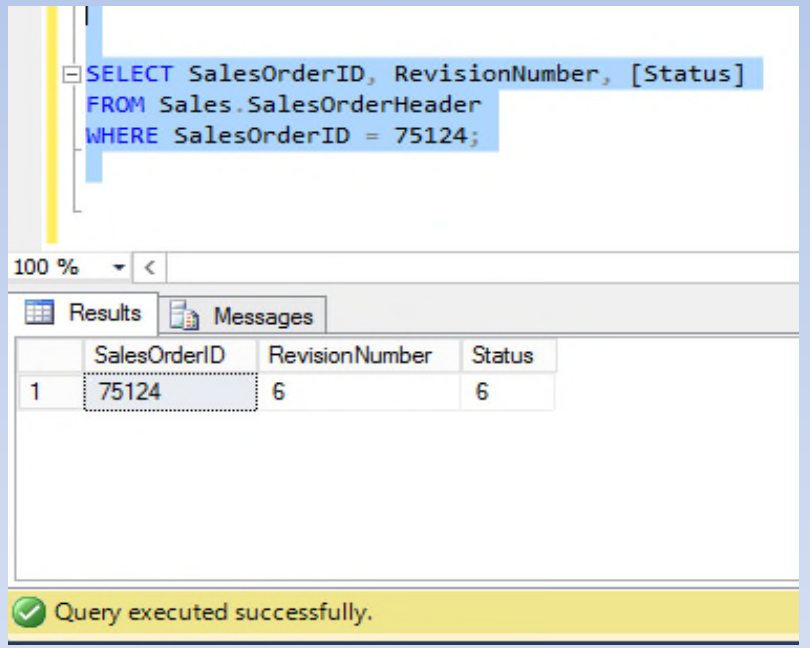
Within the next minute, we perform two updates, one on each instance. In the same row, the Publisher updates [Status]; the Subscriber updates RevisionNumber.

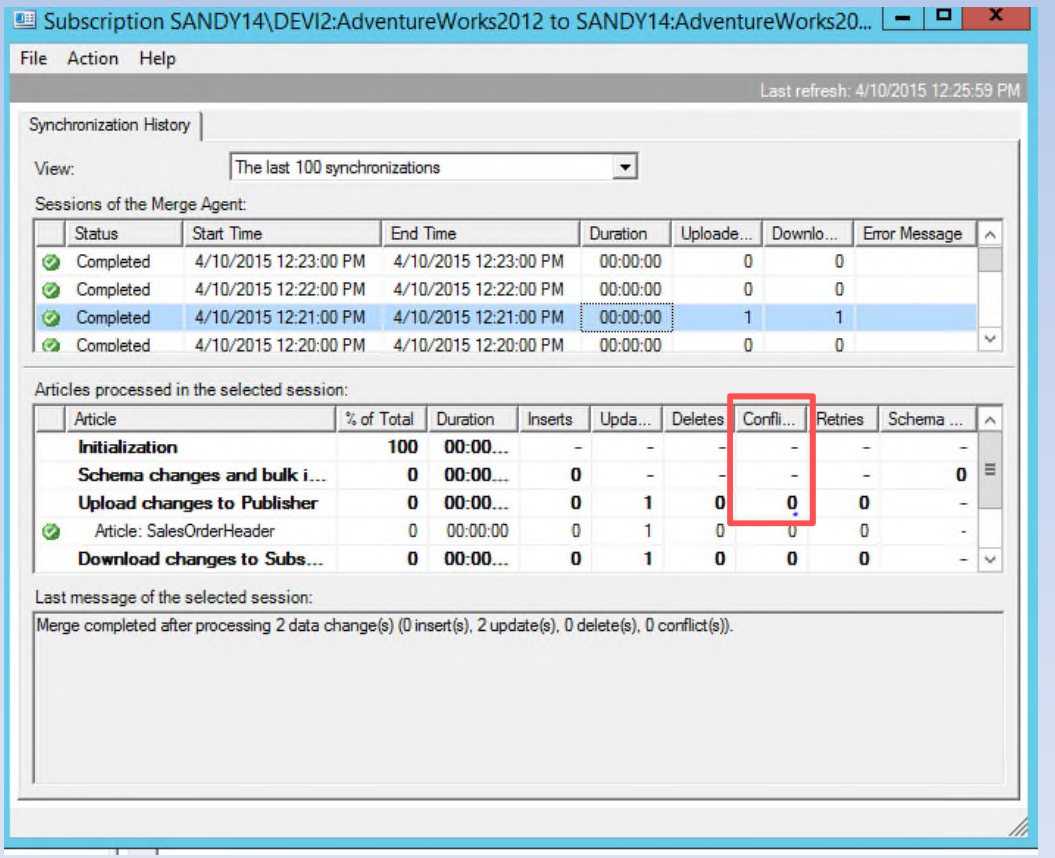




Demo: Conflict Resolution

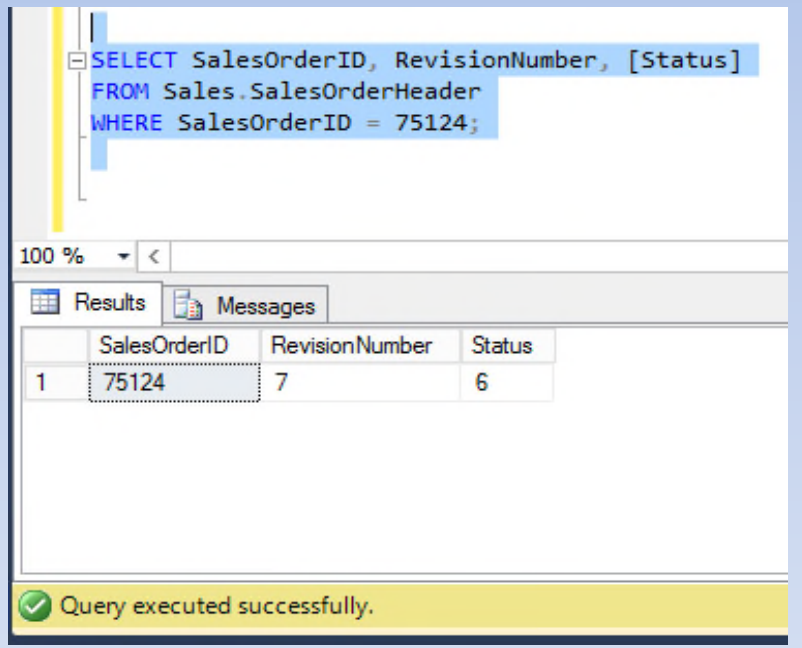
The results are merged, as column-level tracking is being used by merge replication. Replication Monitor shows no conflict.

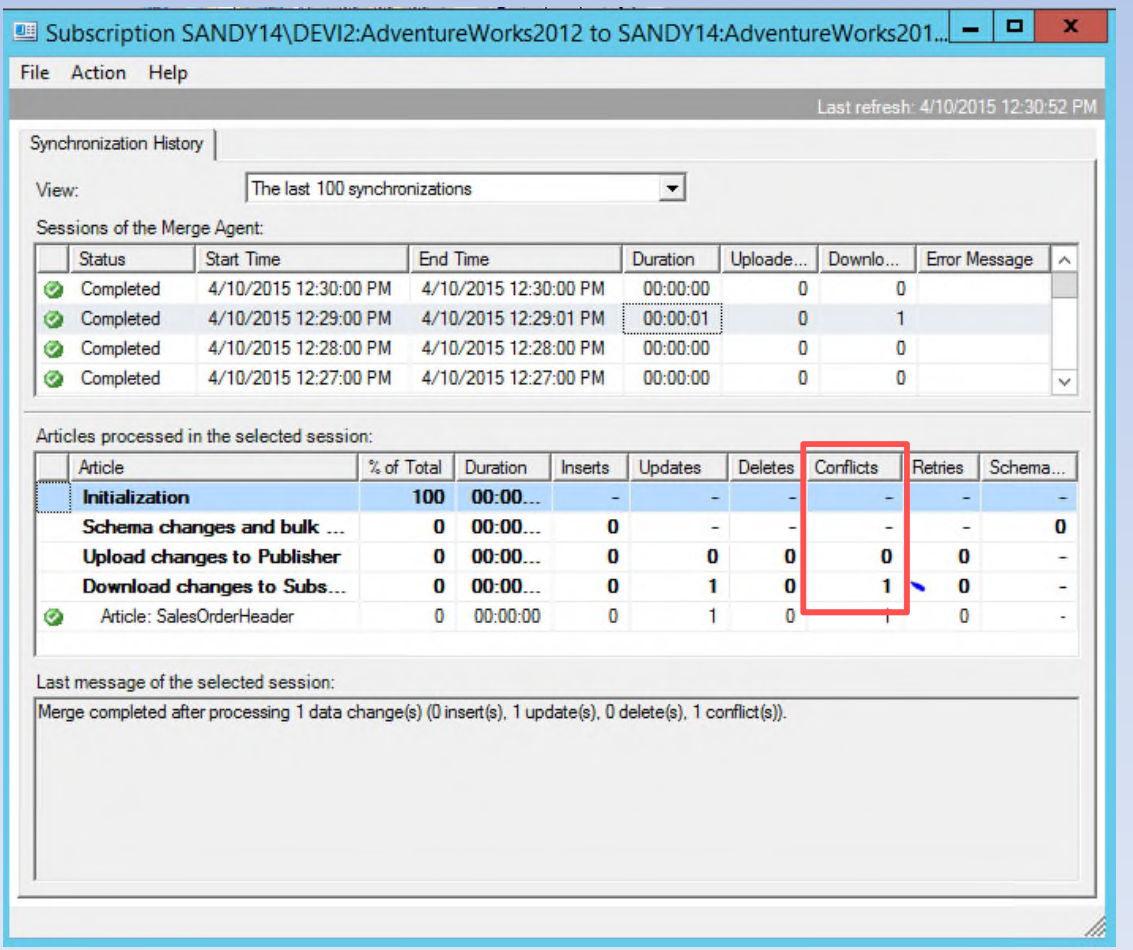




Demo: Conflict Resolution

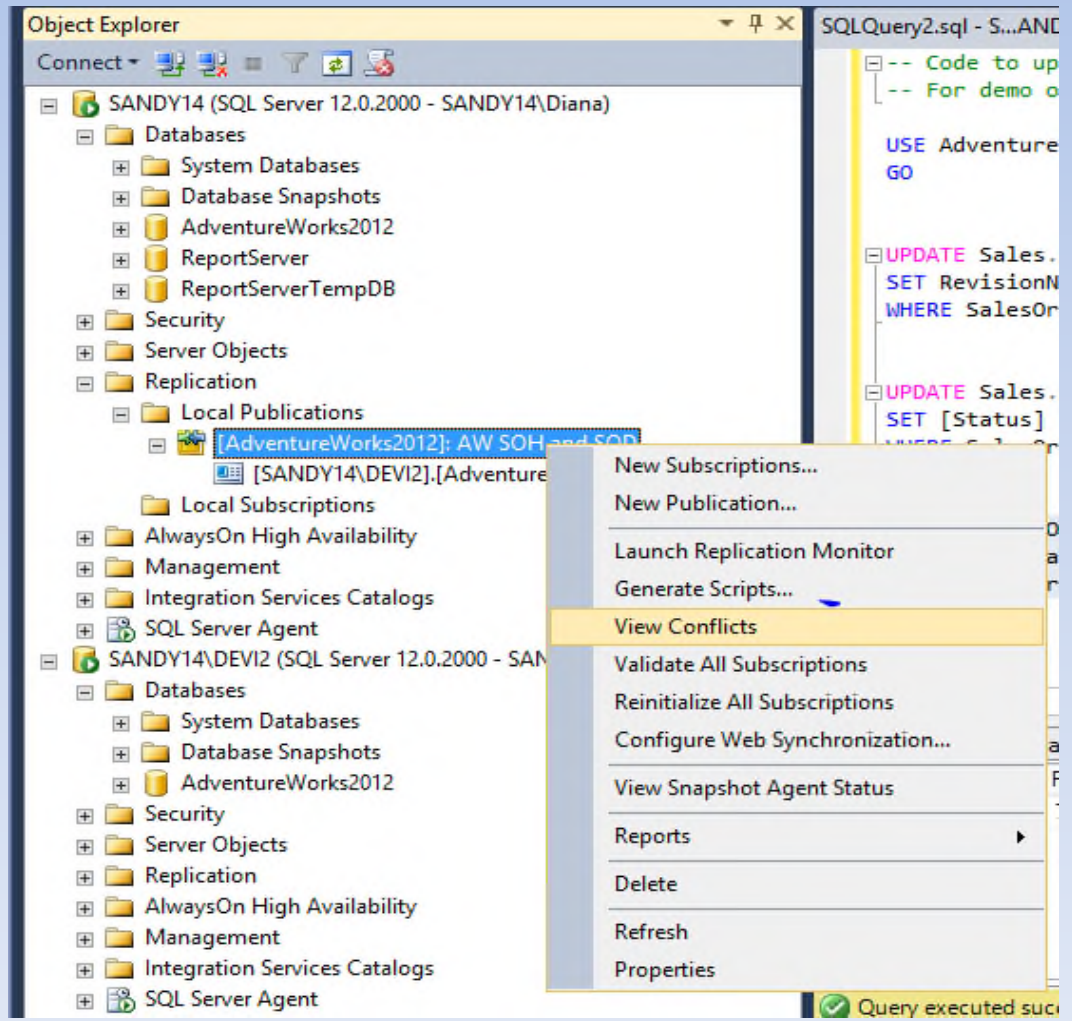
Next, during one replication interval, update RevisionNumber to 7 on the Publisher and to 5 on the Subscriber. The Publisher wins.

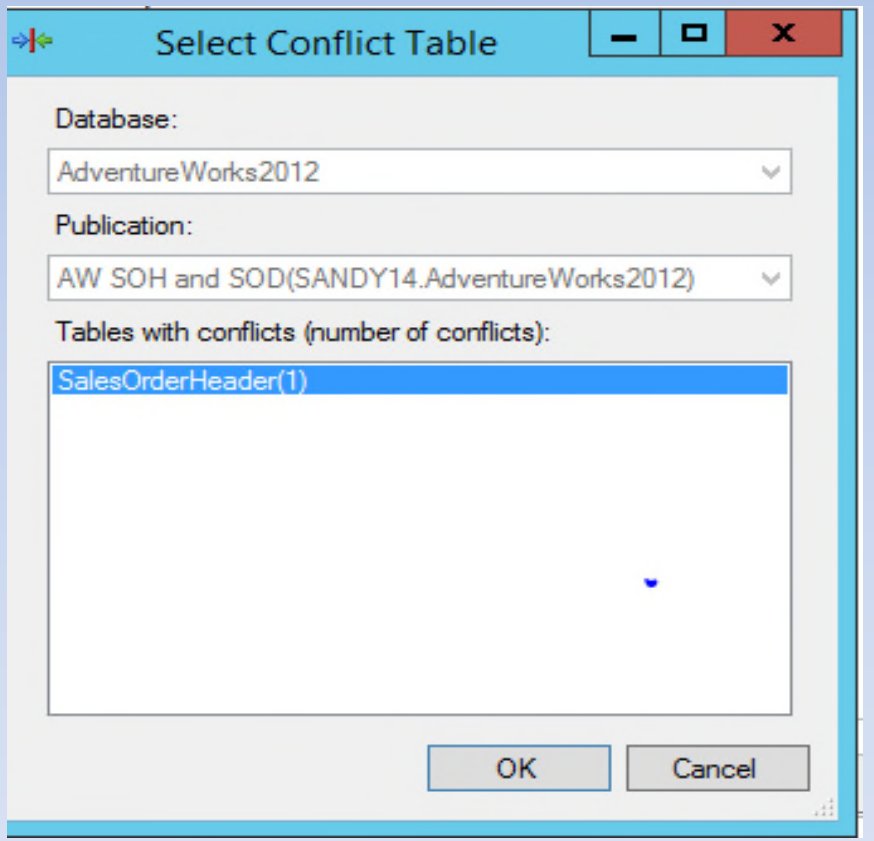




Demo: Conflict Resolution

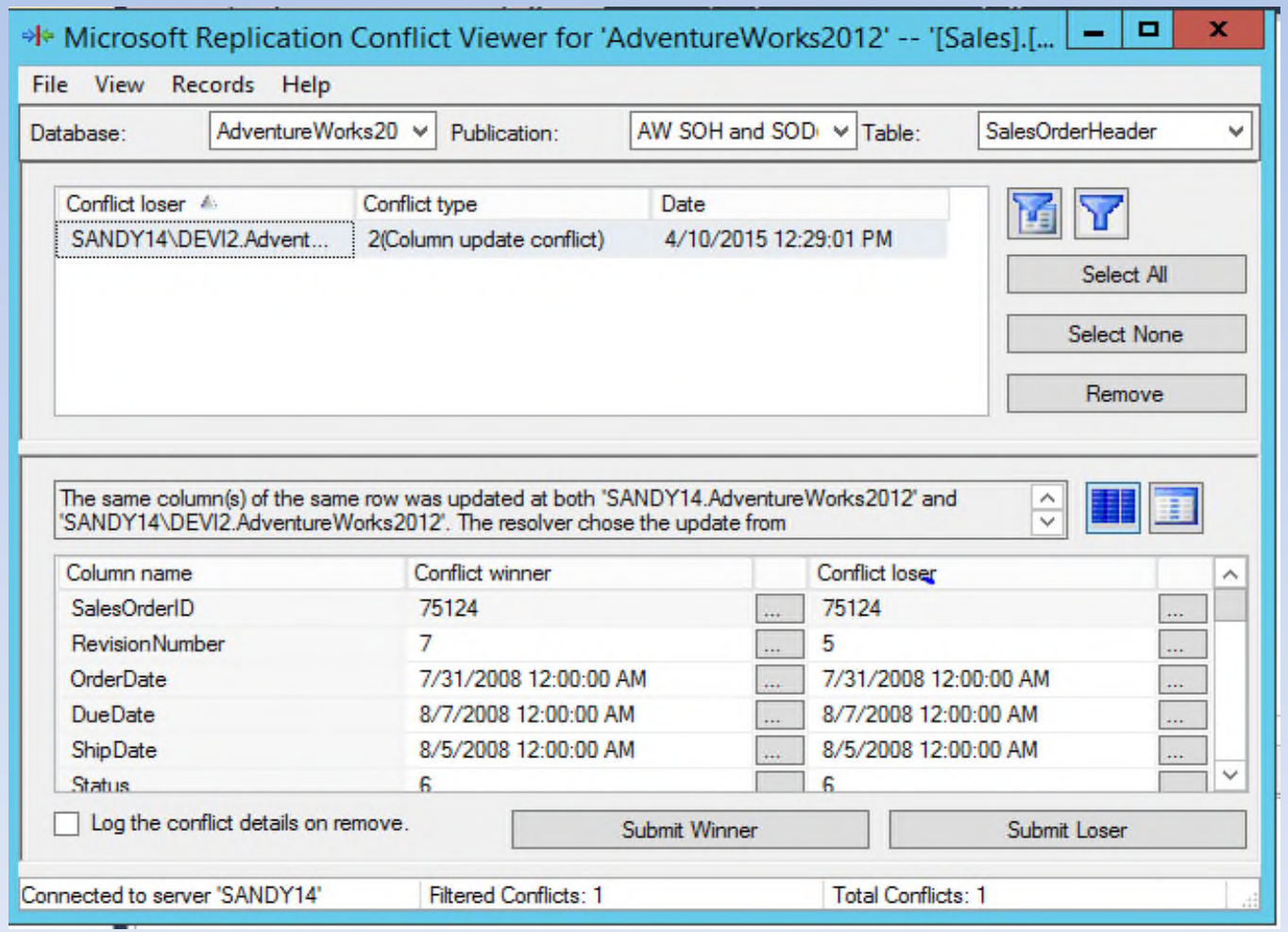
To view the conflict, right-click the publication and then click View Conflicts. Select the conflict.





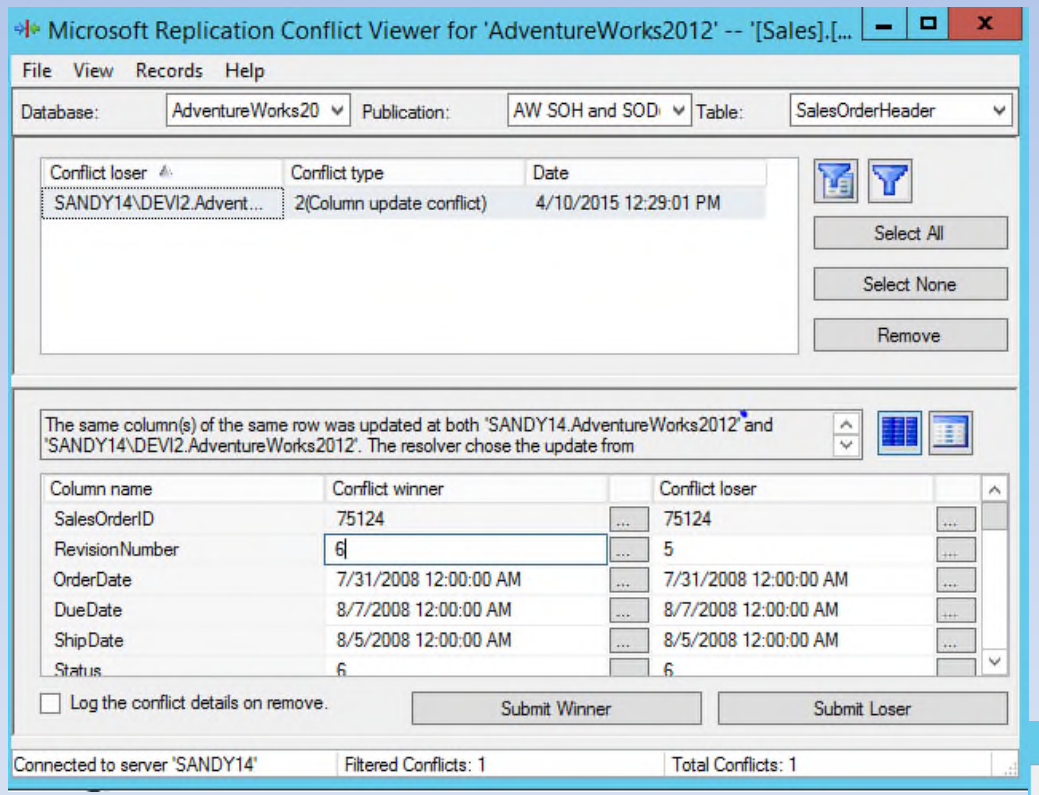
Demo: Conflict Resolution

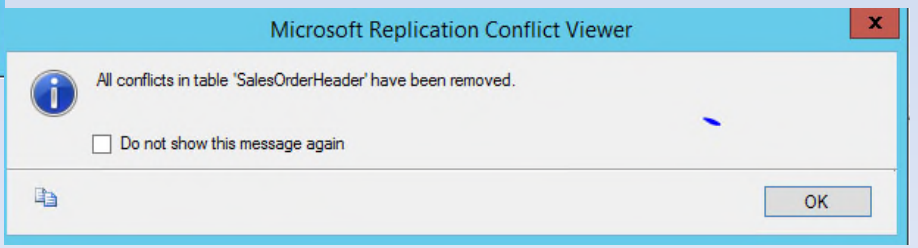
The Conflict Viewer will open, showing the winning and losing values.



Demo: Conflict Resolution

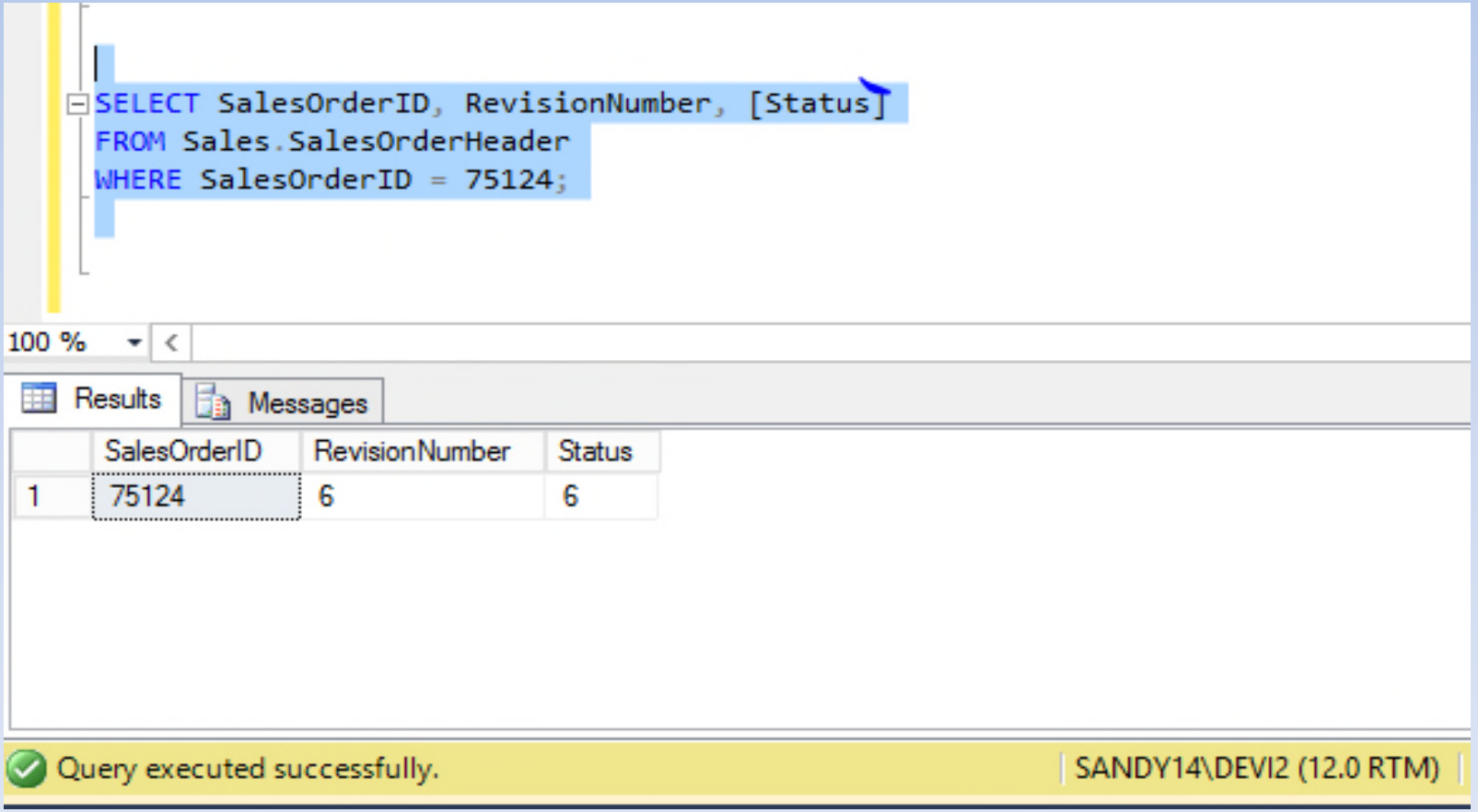
Change the value to 6 under Conflict winner, then click the Submit Winner button. The conflict will be marked as resolved.





Demo: Conflict Resolution

The result is propagated to the Publisher and Subscriber(s).



Why I Love Merge Replication

1. Merge replication is resilient. The publisher and/or one or more subscribers can be down for up to 14 days; when they come back online, all copies will synchronize.
2. Merge replication can provide high availability. If an instance goes down, point its clients to another replication partner instance.
3. The extra metadata required for merge replication will almost never be a “show stopper.”

Contact Info. and Further Reading

Presenter Diana Dee may be contacted at:

DrDianaDee@verizon.net

(Scripts and instructions for the demos will be provided on request.)

Further Reading:

SQL Server Books Online (BOL) (for whatever version you are using)

(Recommended BOL article: "Replication Agent Security Model")

Stairway to SQL Server Replication

http://www.sqlservercentral.com/stairway/72401/