

Configure SQL SERVER Transactional Replication with Always on

Hub Sub Techno

Kamran Hussain

Existing Configure

Domain: hst.com 192.168.179.141

Windows Cluster: sqlwcl 192.168.179.135

Always on Cluster: sqlcl 192.168.179.134

Always on Cluster Listener sqlli 192.168.179.136

3 Node Always on Cluster

sql1 192.168.179.131

sql2 192.168.179.132

sql3 192.168.179.133

DB1 = Primary DB for Transaction

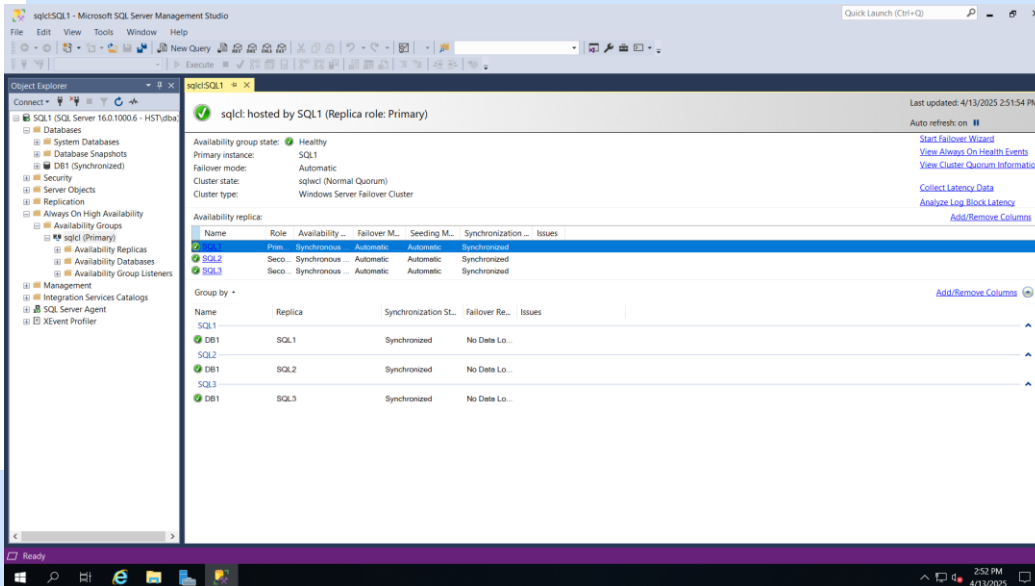
DB1SUB= Primary DB Copy for Reporting

S. No	IP	Name	Purpose	Publisher DB	Subscriber DB	Distributor DB
1	192.168.179.130	SQL1	Publisher, Subscriber	DB1	DB1SUB	
2	192.168.179.131	SQL2	Publisher, Subscriber	DB1	DB1SUB	
3	192.168.179.132	SQL3	Publisher, Subscriber	DB1	DB1SUB	
4	192.168.179.142	SQLDistributor	Distributor			Distributor
5	192.168.179.136	SQLLI	Listener			

High Level Steps

1. Create Distributor at Distributor Server
2. Add Primary Publisher Name at Distributor Server
3. Create Distributor at Primary Publisher Server
4. Create Publisher with Always on Listener connection
5. Create Subscriber Database and Add in Always on Cluster Group
6. Create Subscriber with Always on Listener connection
7. Add Secondary Node as Publisher Name at Distributor Server
8. Create Distributor at Secondary Servers and Create Linked Server
9. Define Publisher Re – Direct at Distributor
10. Monitor Replication

Current Configuration Health



The screenshot displays the 'sqlcl: hosted by SQL1 (Replica role: Primary)' configuration in SQL Server Enterprise Manager. The 'Availability group state' is 'Healthy'. The primary instance is 'SQL1'. The failover mode is 'Automatic', and the cluster state is 'sqlcl (Normal Quorum)'. The cluster type is 'Windows Server Failover Cluster'.

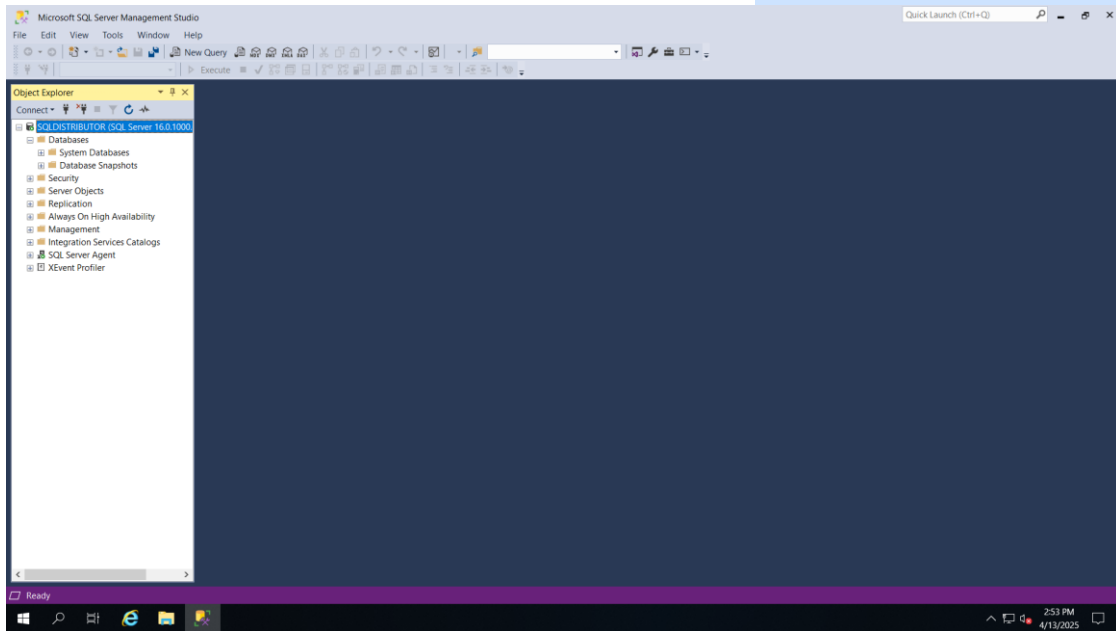
The 'Availability replica' table shows the following configuration:

Name	Role	Availability ...	Failover M...	Seeding M...	Synchronization ...	Issues
SQL1	Primary	Synchronous	Automatic	Automatic	Synchronized	
SQL2	Seco...	Synchronous	Automatic	Automatic	Synchronized	
SQL3	Seco...	Synchronous	Automatic	Automatic	Synchronized	

The 'Group by' table shows the replication status for each database across the replicas:

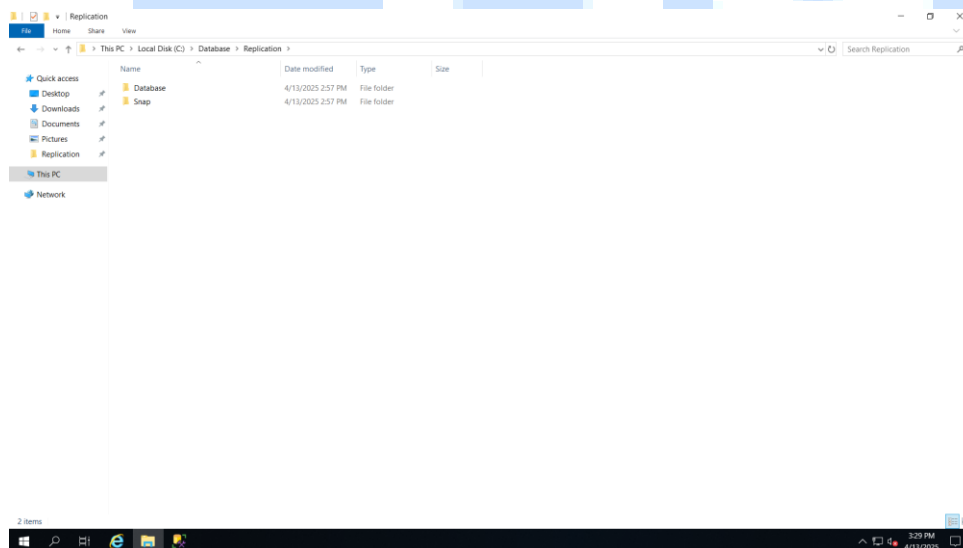
Name	Replica	Synchronization St...	Failover Re...	Issues
SQL1				
DB1	SQL1	Synchronized	No Data Lo...	
SQL2				
DB1	SQL2	Synchronized	No Data Lo...	
SQL3				
DB1	SQL3	Synchronized	No Data Lo...	

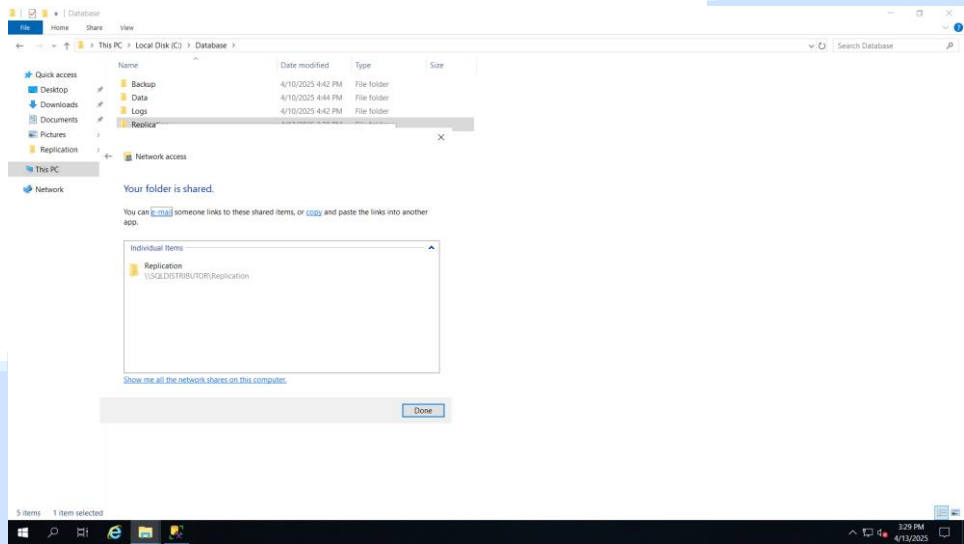
Create Distributor at Distributor Server



Create Share Folder "C:\Database\Replication\Snap"

\\sql distributor\Replication\Snap

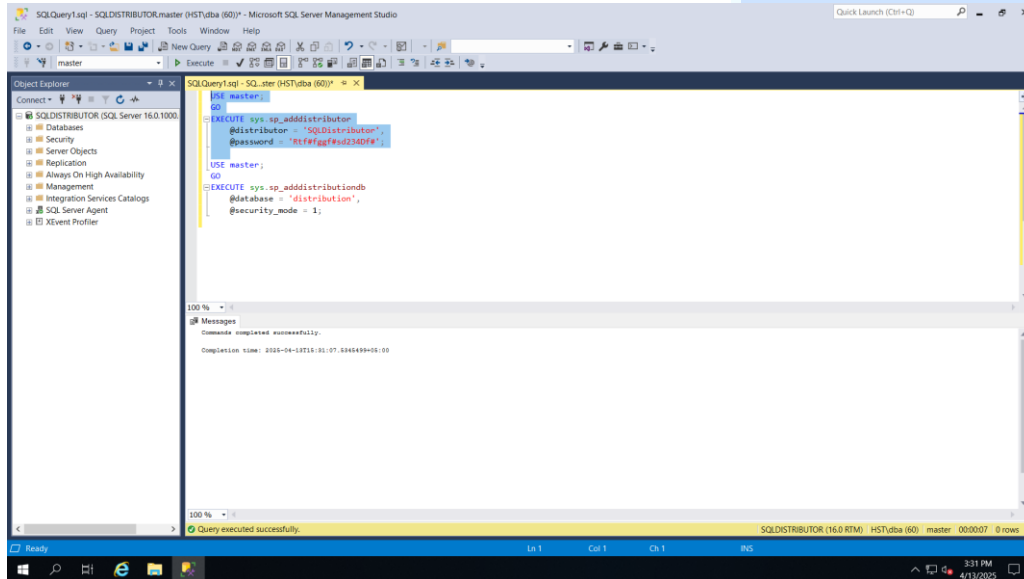




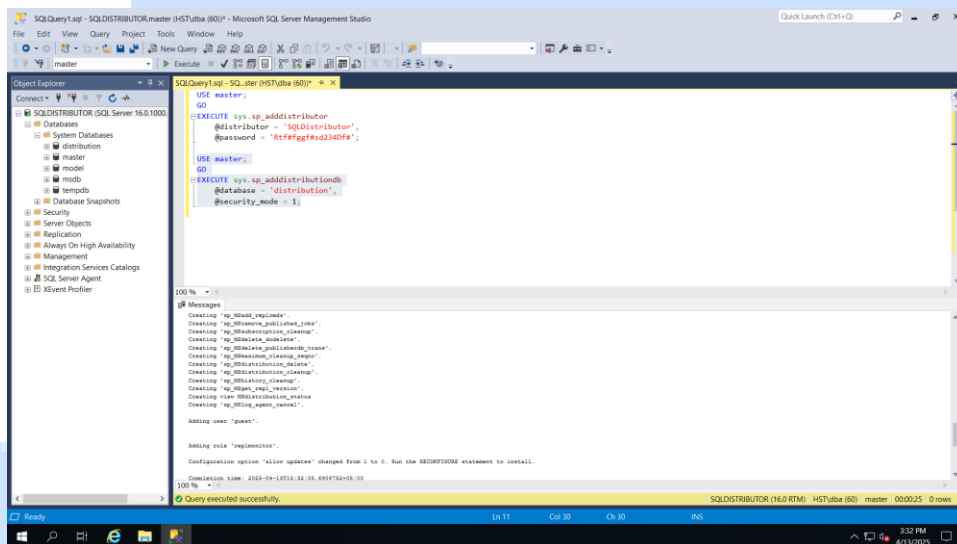
Hum

ECH

```
USE master;
GO
EXECUTE sys.sp_addistributor
    @distributor = 'SQLDistributor',
    @password = 'Rtf#fggf#sd234Df#';
```



```
USE master;
GO
EXECUTE sys.sp_adddistributiondb
    @database = 'distribution',
    @security_mode = 1;
```



Add Primary Publisher Name at Distributor Server

USE master;

GO

EXECUTE sys.sp_adddistpublisher

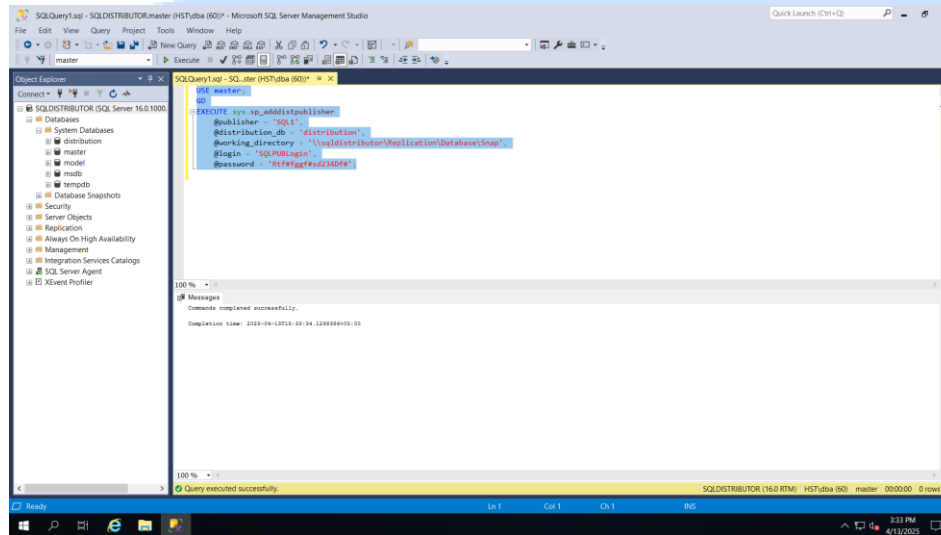
@publisher = 'SQL1',

@distribution_db = 'distribution',

@working_directory = '\\sqldistributor\Replication\Snap',

@login = 'SQLPUBLogin',

@password = 'Rtf#fggf#sd234Df#';



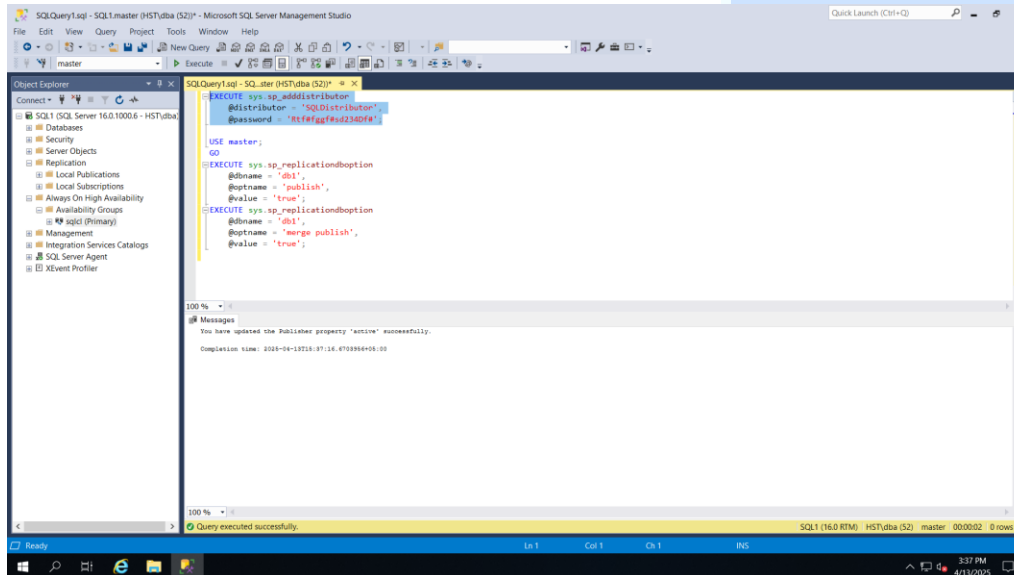
Humio ECH

Create Distributor at Primary Publisher Server

EXECUTE sys.sp_adddistributor

@distributor = 'SQLDistributor',

@password = 'Rtf#fggf#sd234Df#';



USE master;

GO

EXECUTE sys.sp_replicationdboption

@dbname = 'db1',

@optname = 'publish',

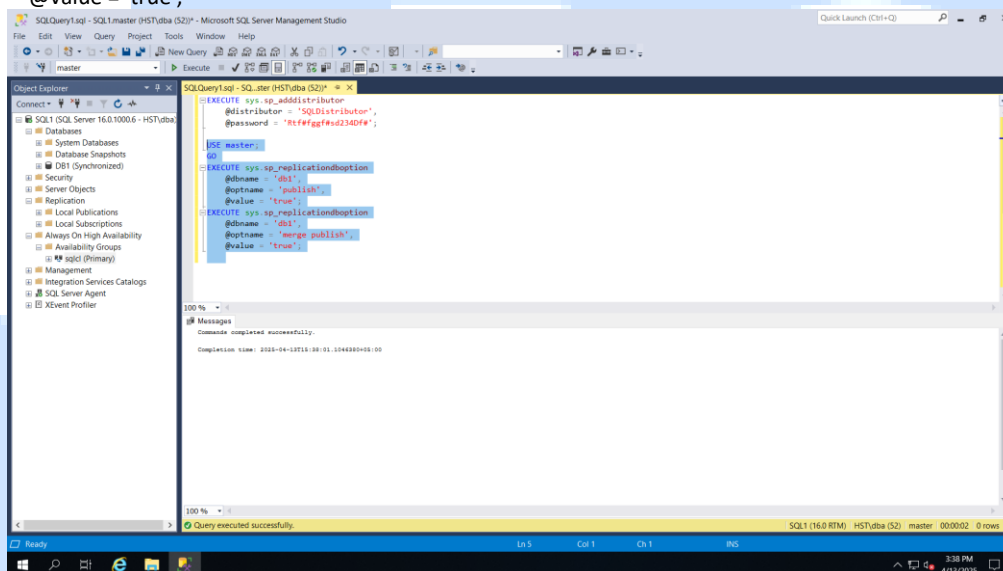
@value = 'true';

EXECUTE sys.sp_replicationdboption

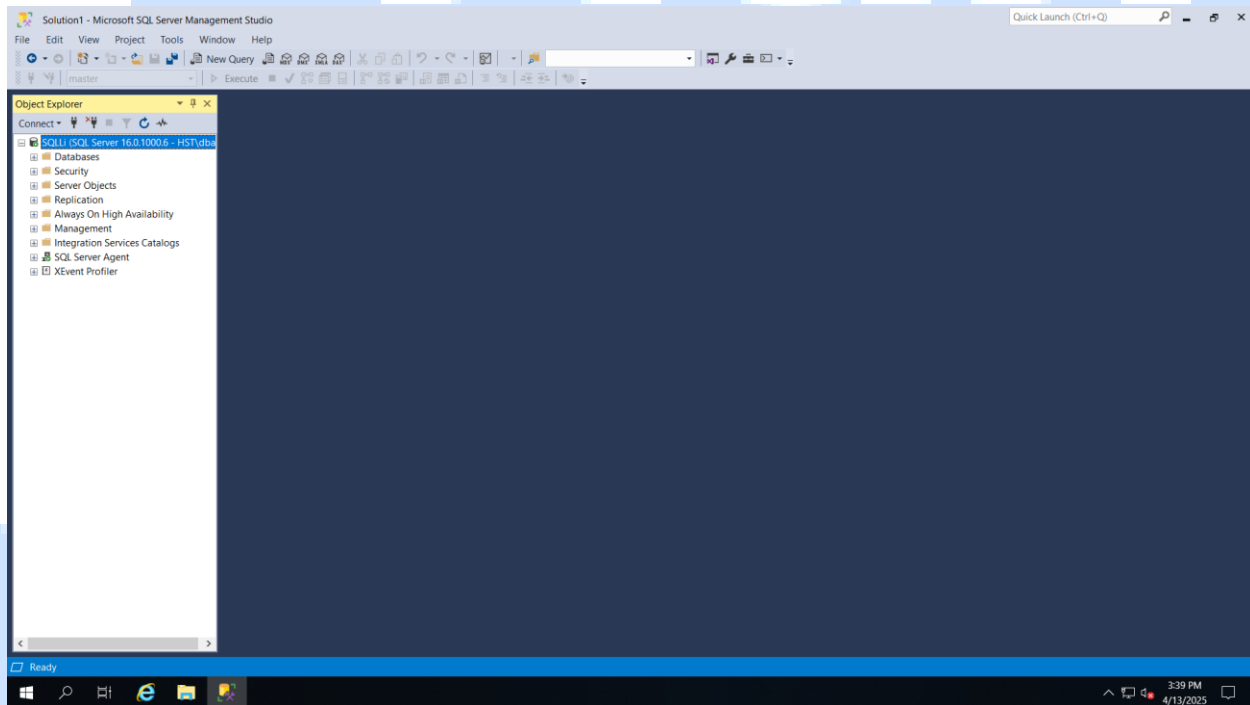
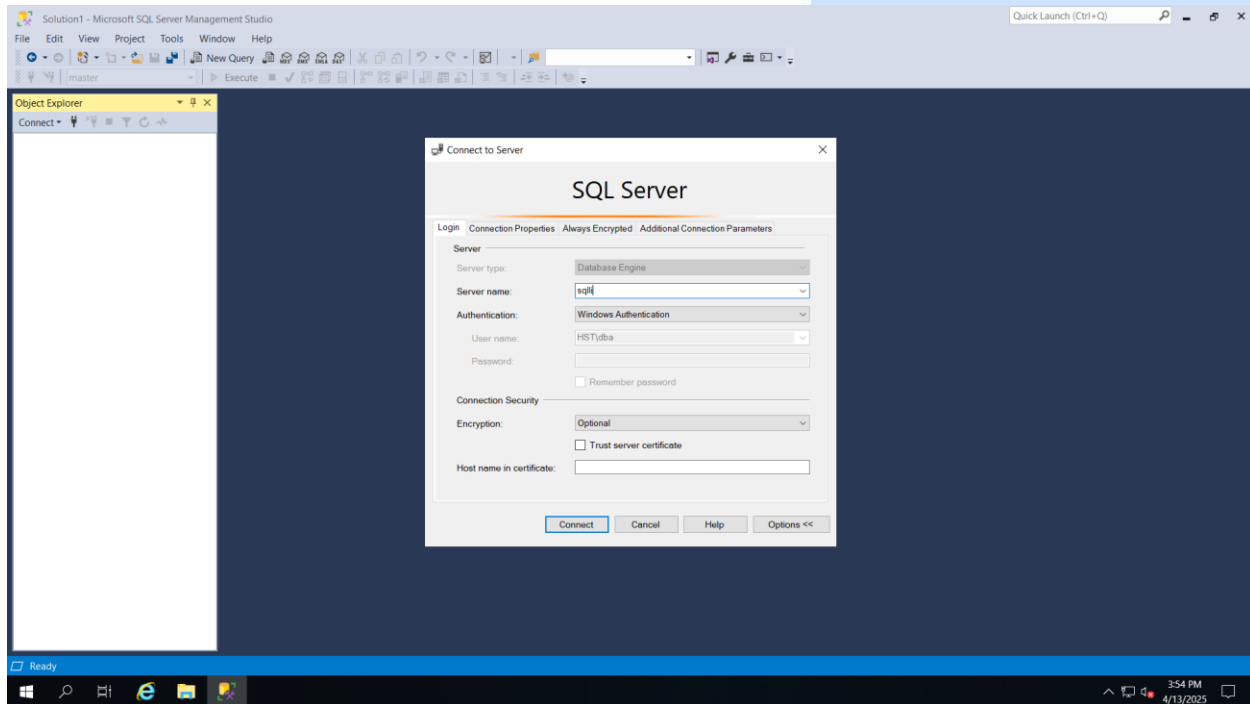
@dbname = 'db1',

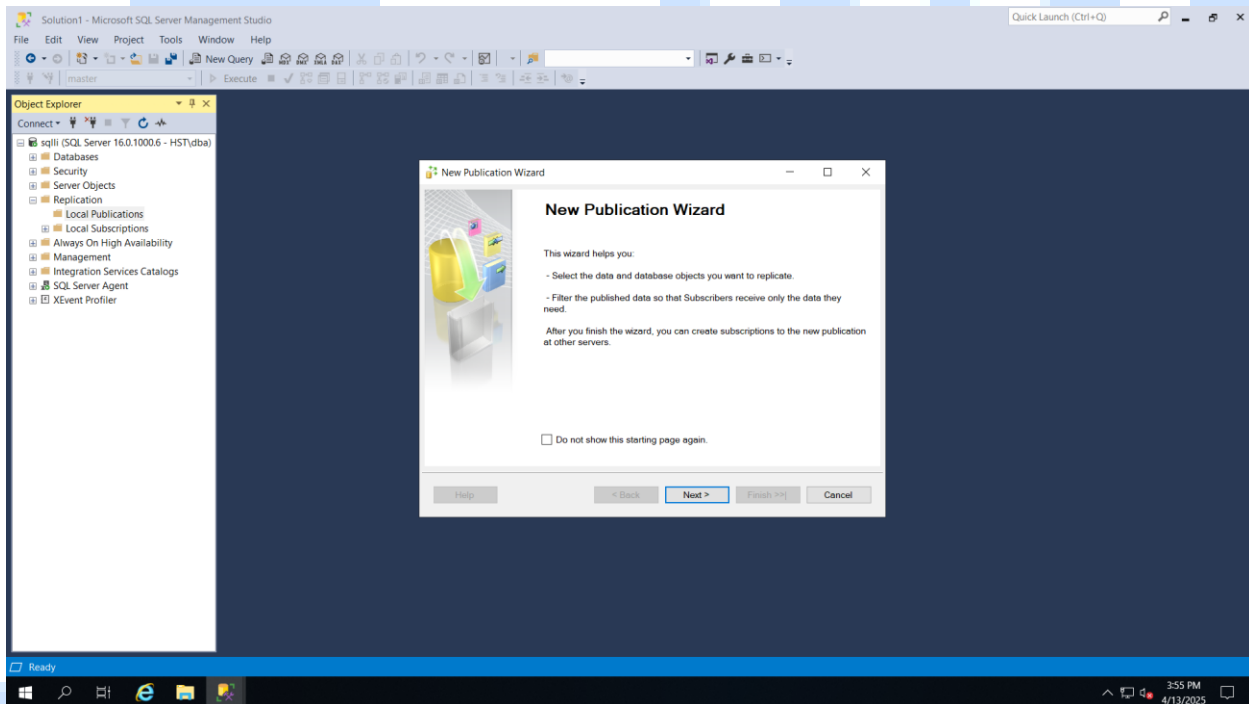
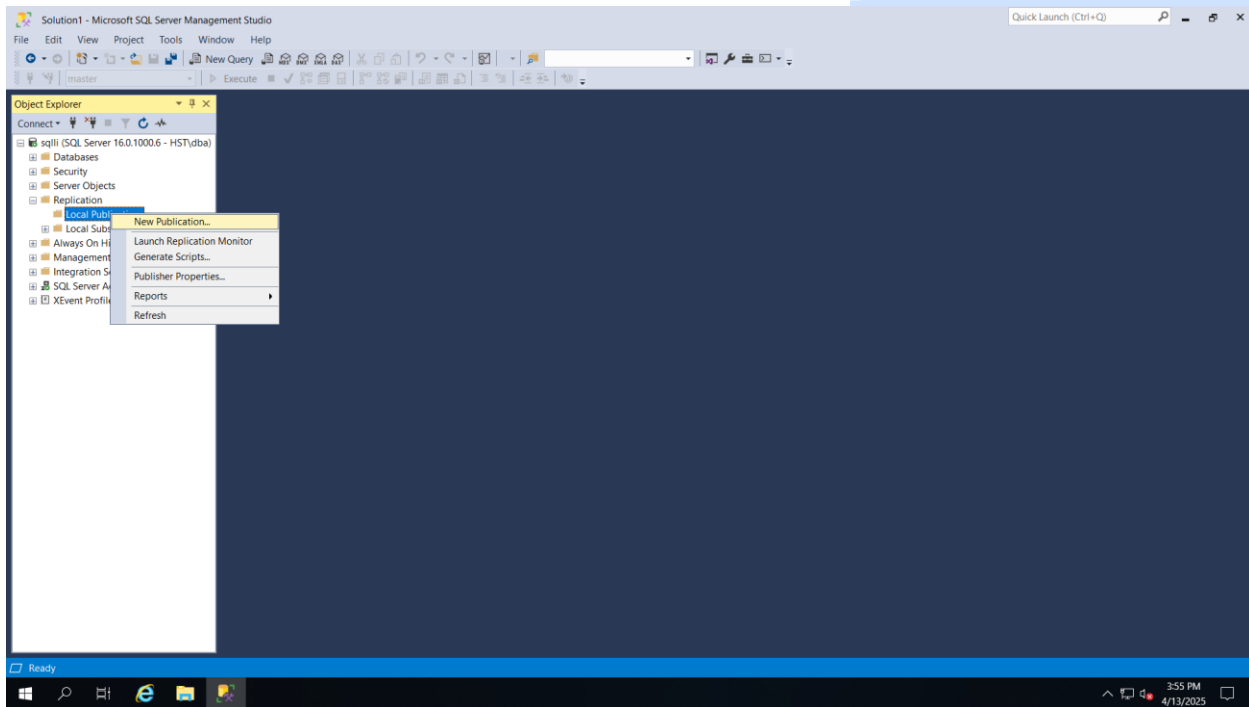
@optname = 'merge publish',

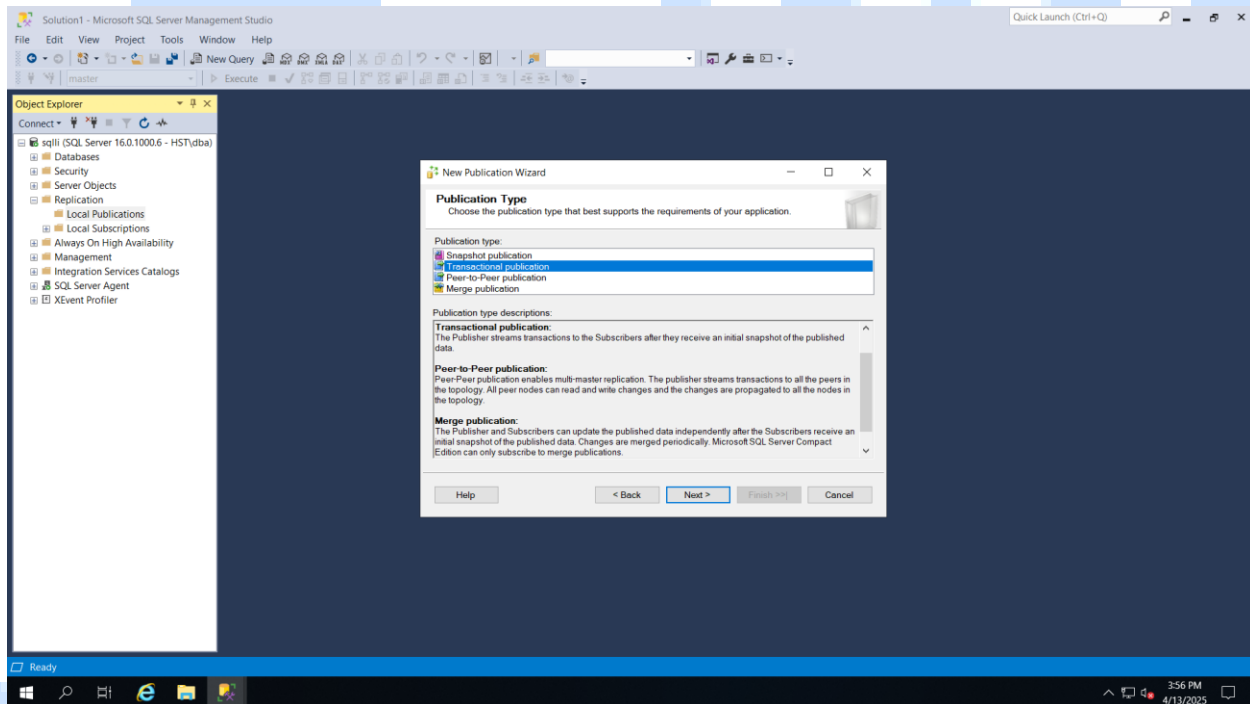
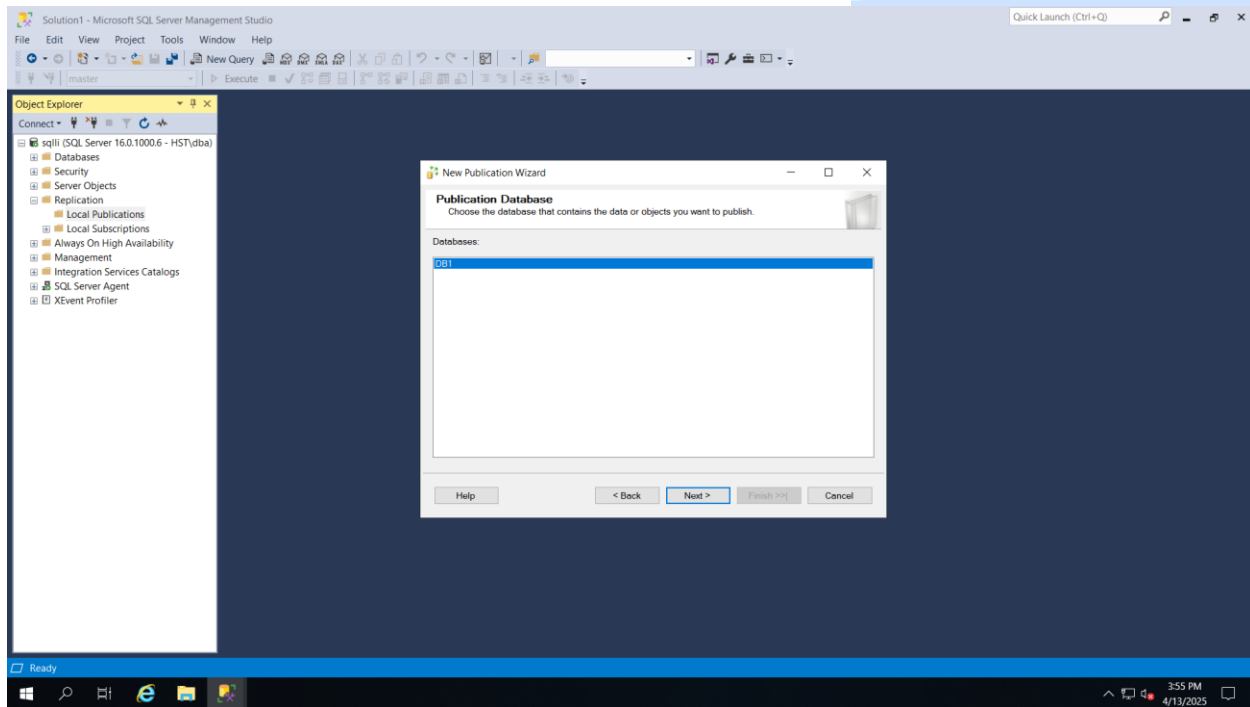
@value = 'true';

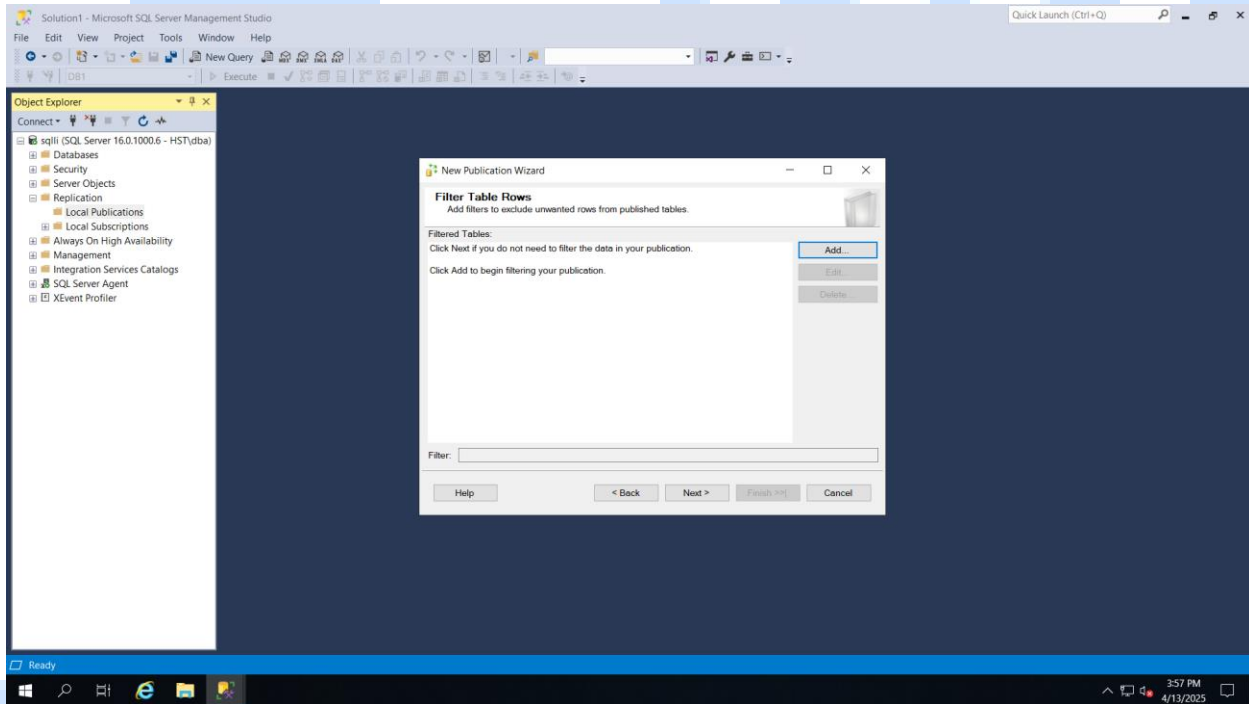
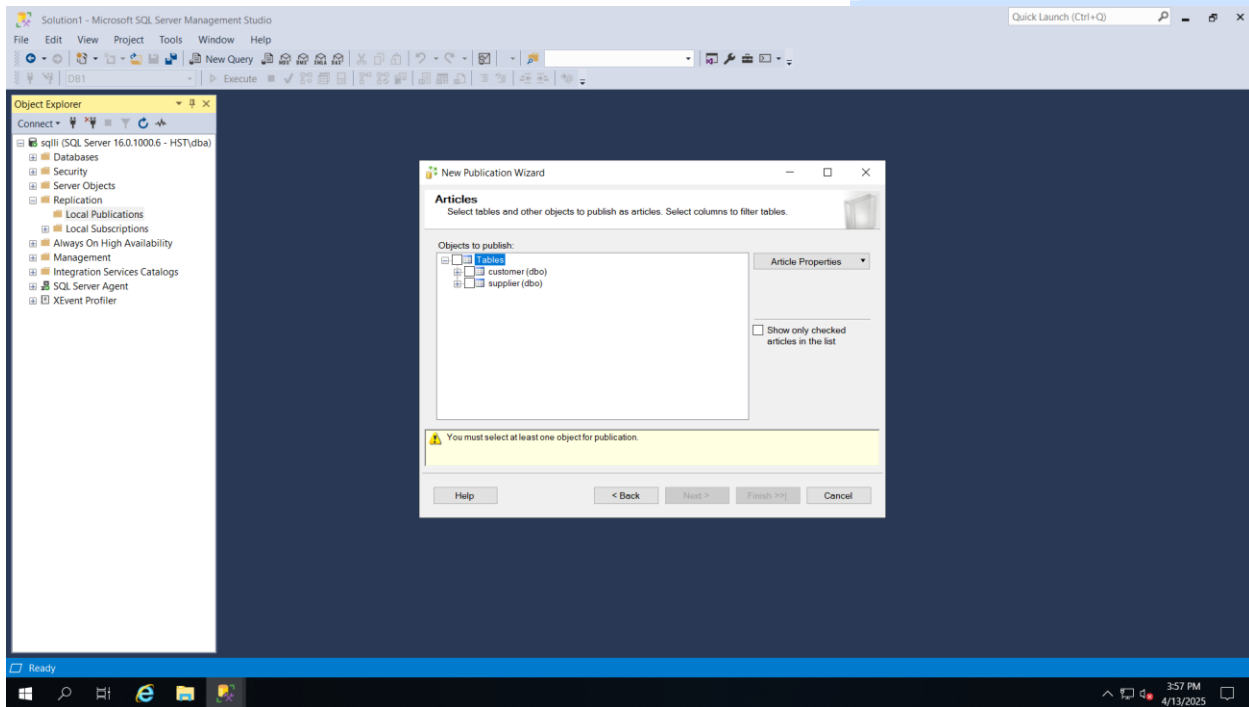


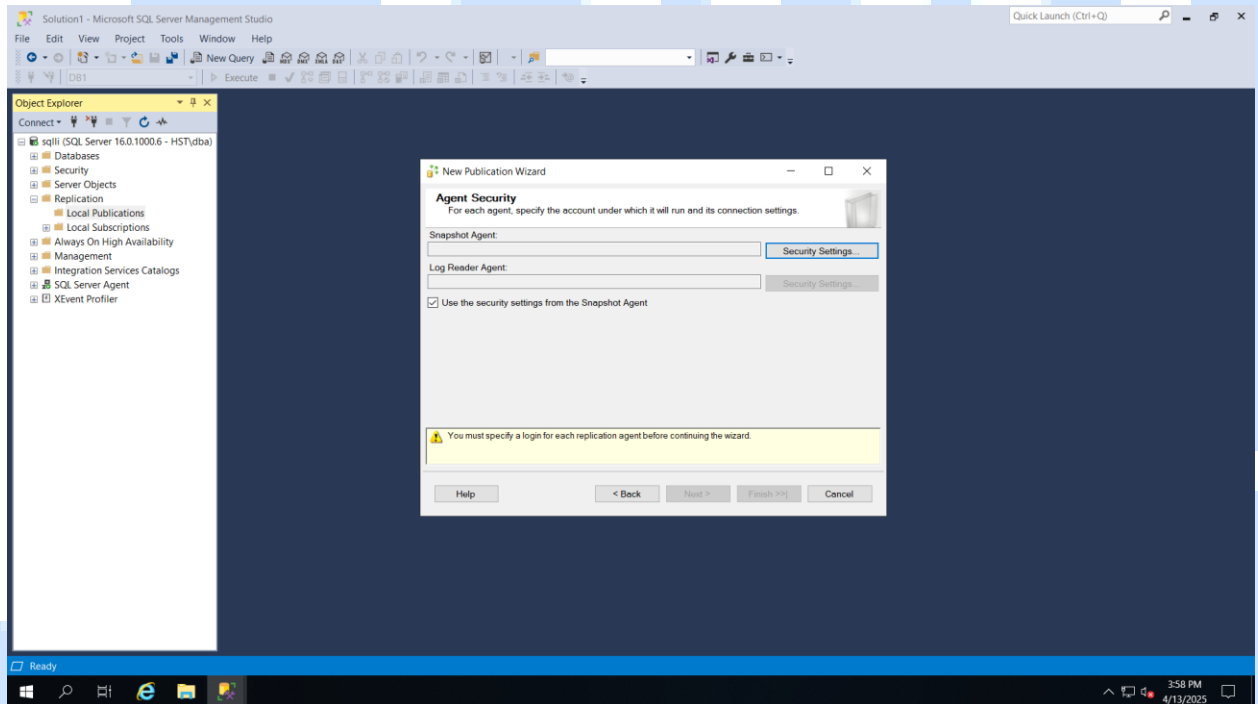
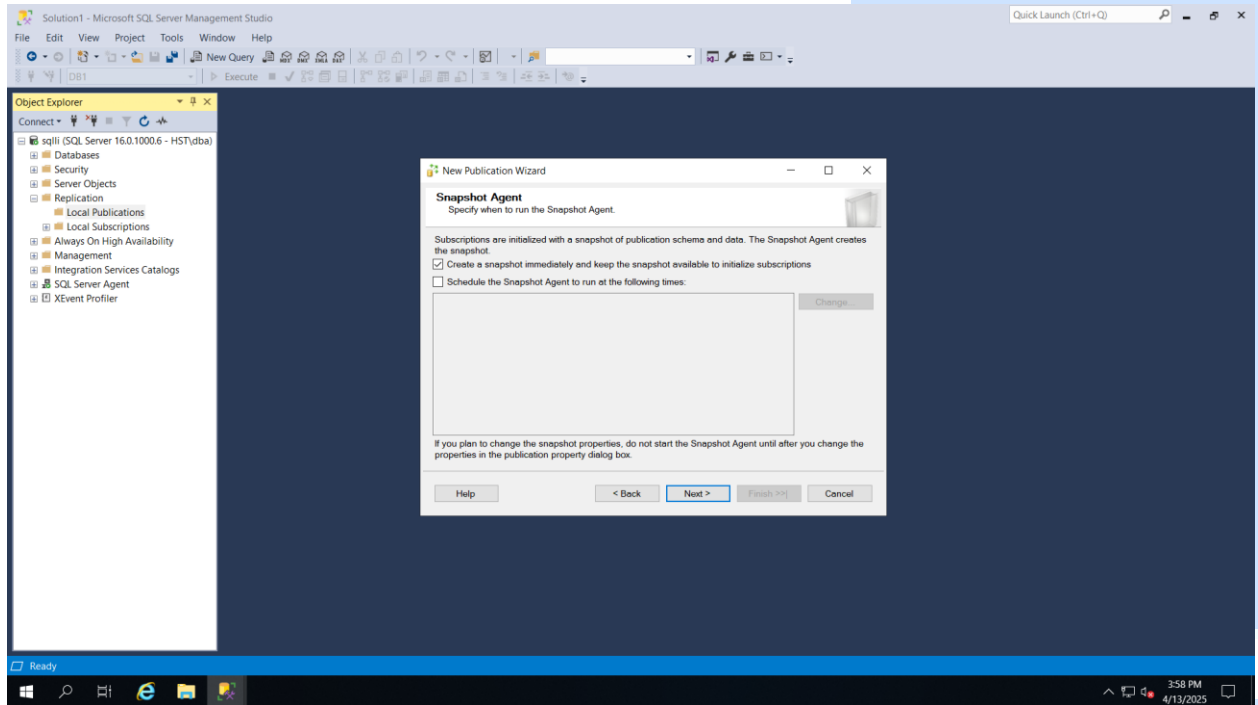
Create Publisher with Always on Listener connection

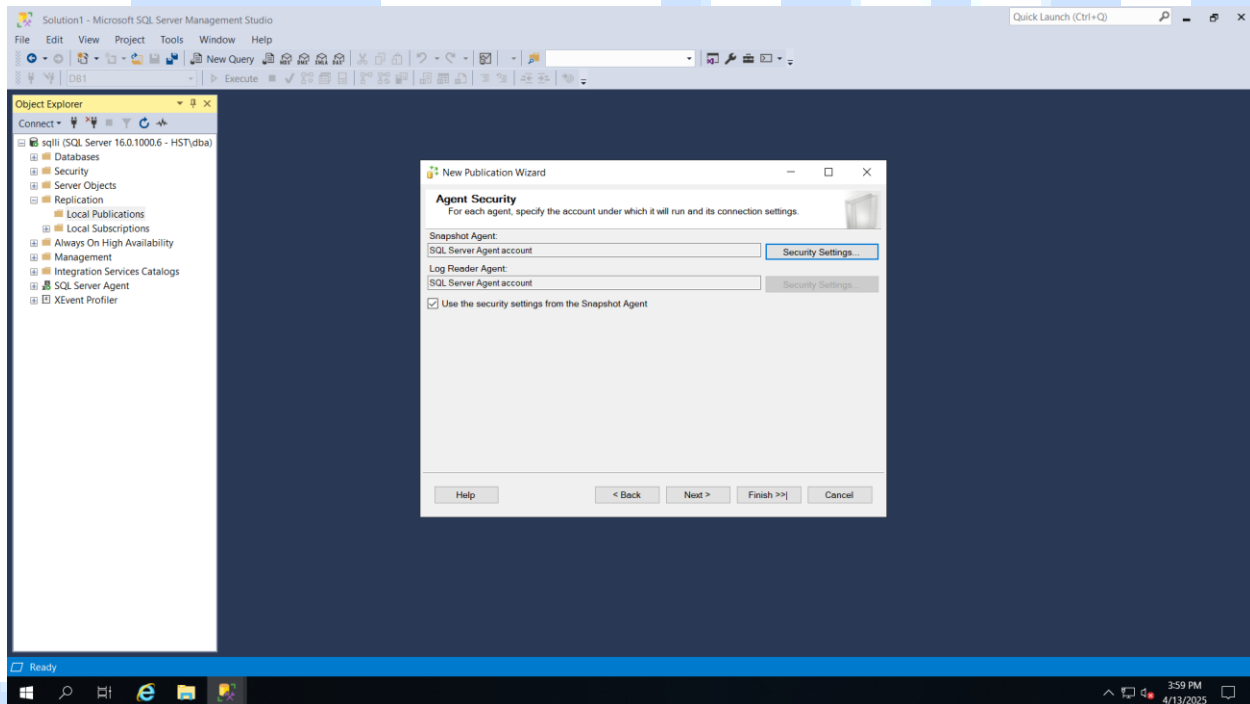
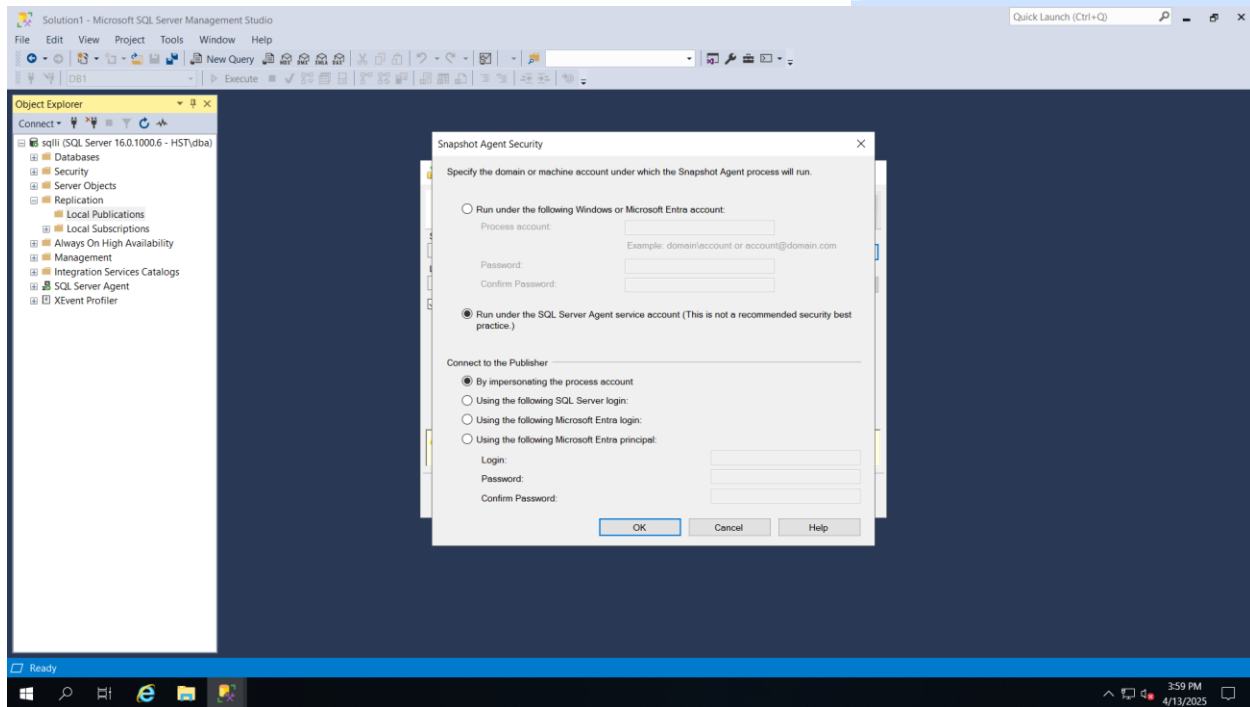


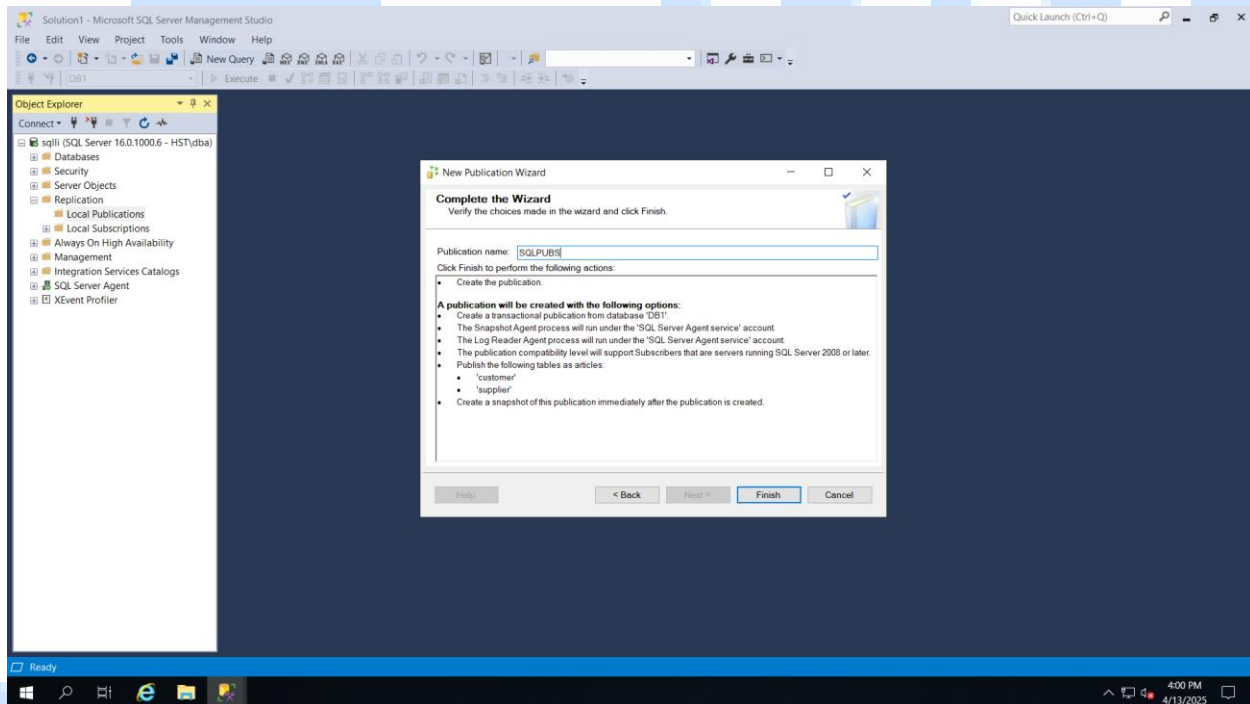
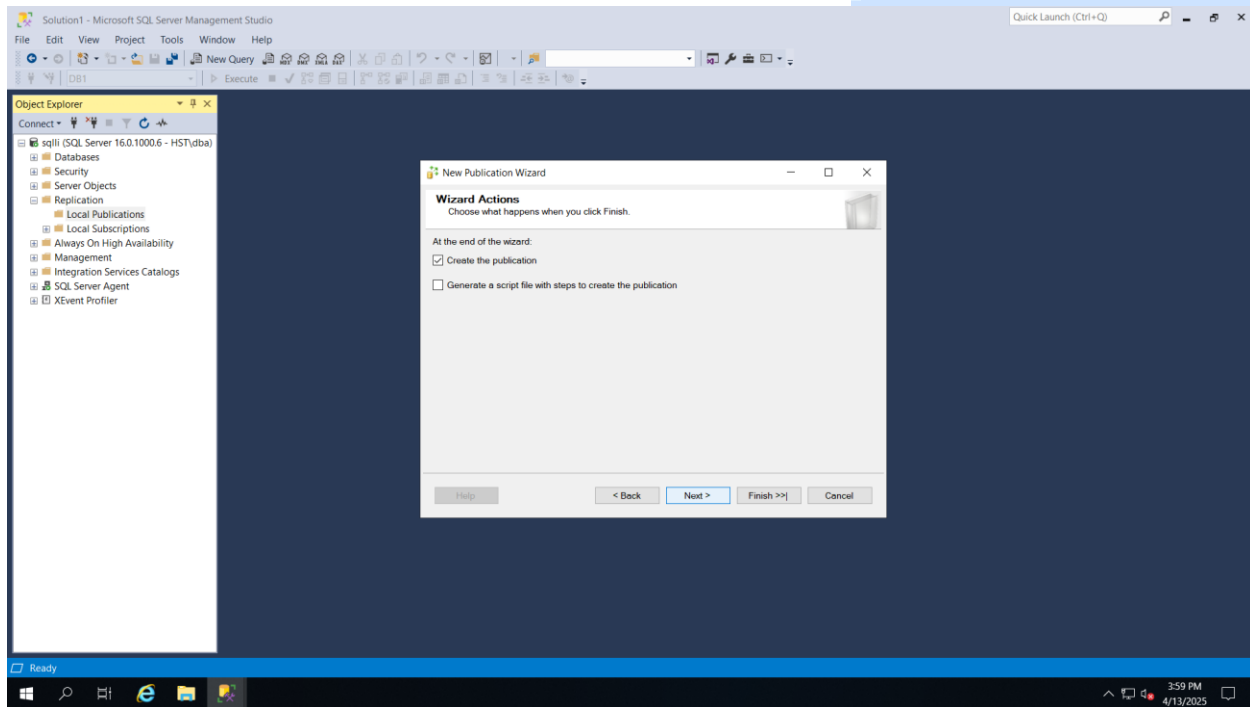














Solution1 - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Project Tools Window Help

New Query

Execute

Object Explorer

Connect

sqlcli (SQL Server 16.0.1000.6 - HST\dba)

Databases

Security

Server Objects

Replication

Local Publications

Local Subscriptions

Always On High Availability

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

New Publication Wizard

Creating Publication

Click Stop to interrupt the operation.

Success

3 Total 0 Error

3 Success 0 Warning

Details

Action	Status	Message
Creating Publication 'SQLPUBS'	Success	
Adding article 2 of 2	Success	
Starting the Snapshot Agent	Success	

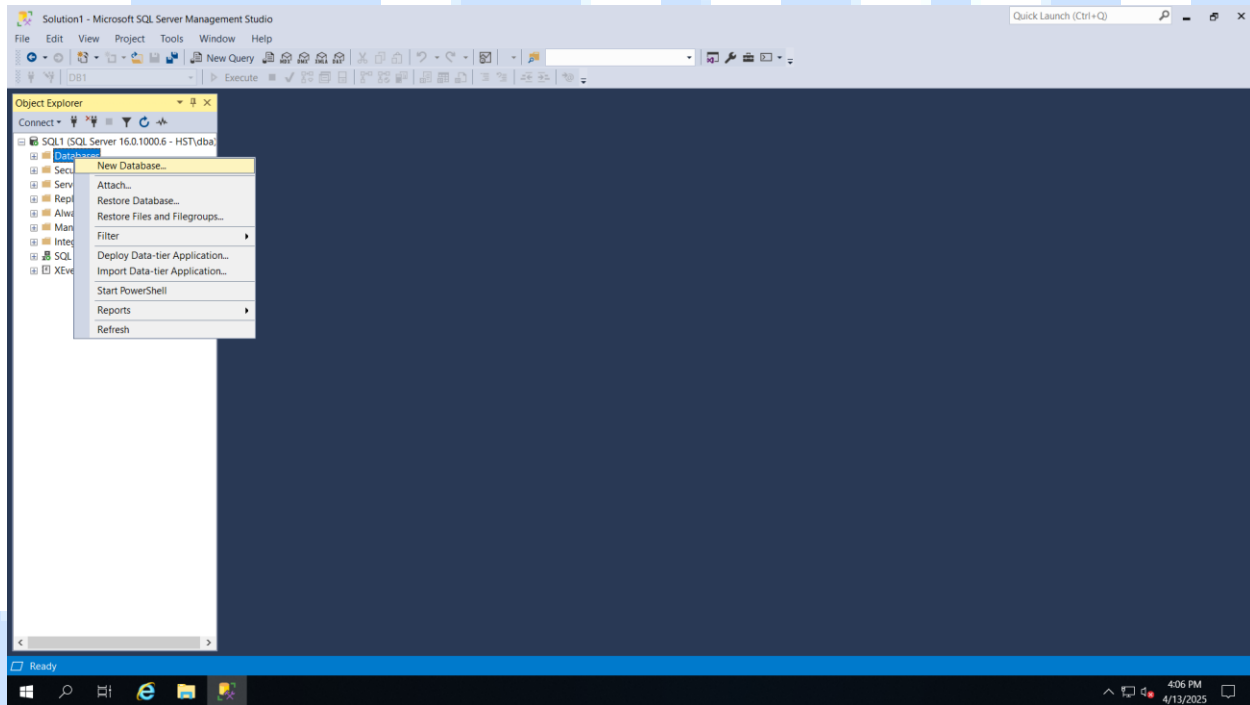
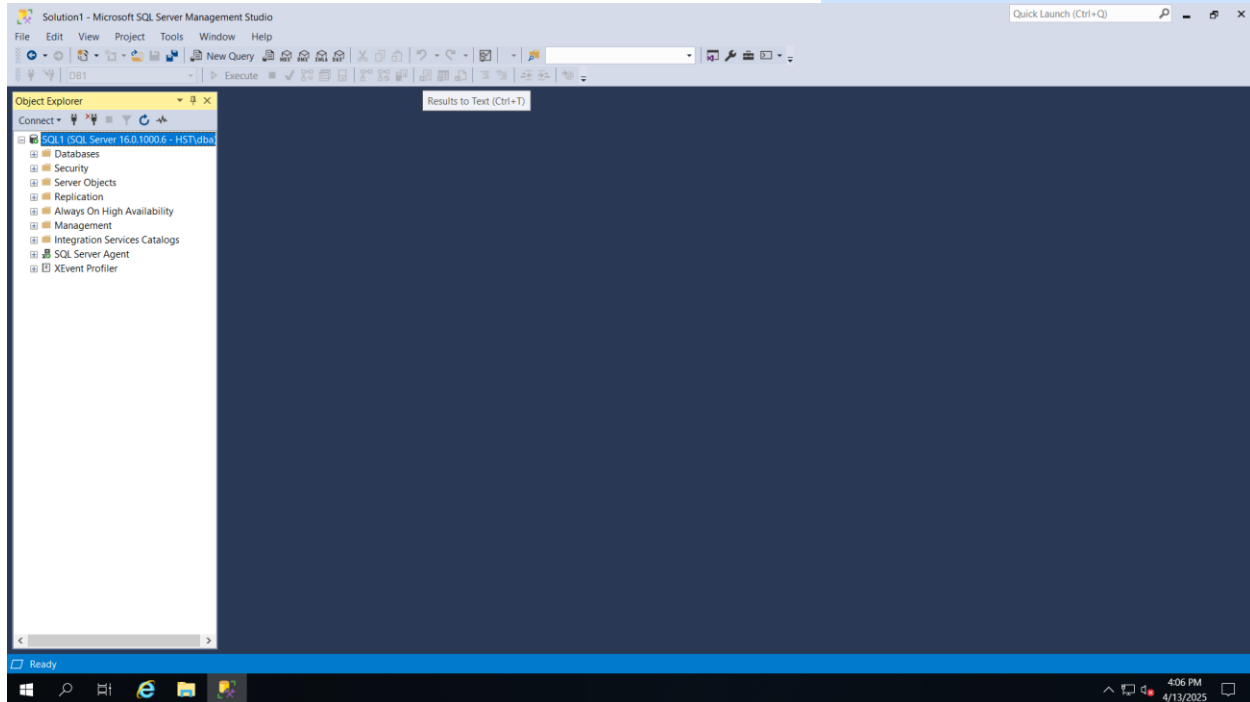
Stop Report

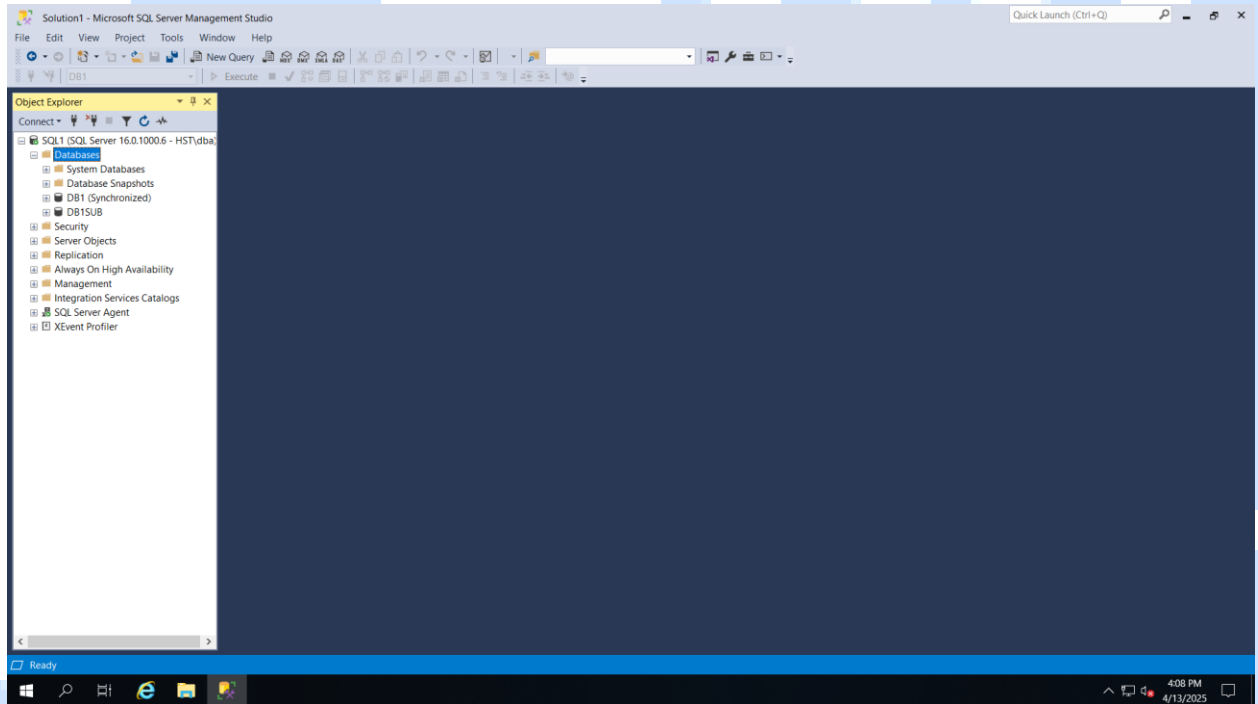
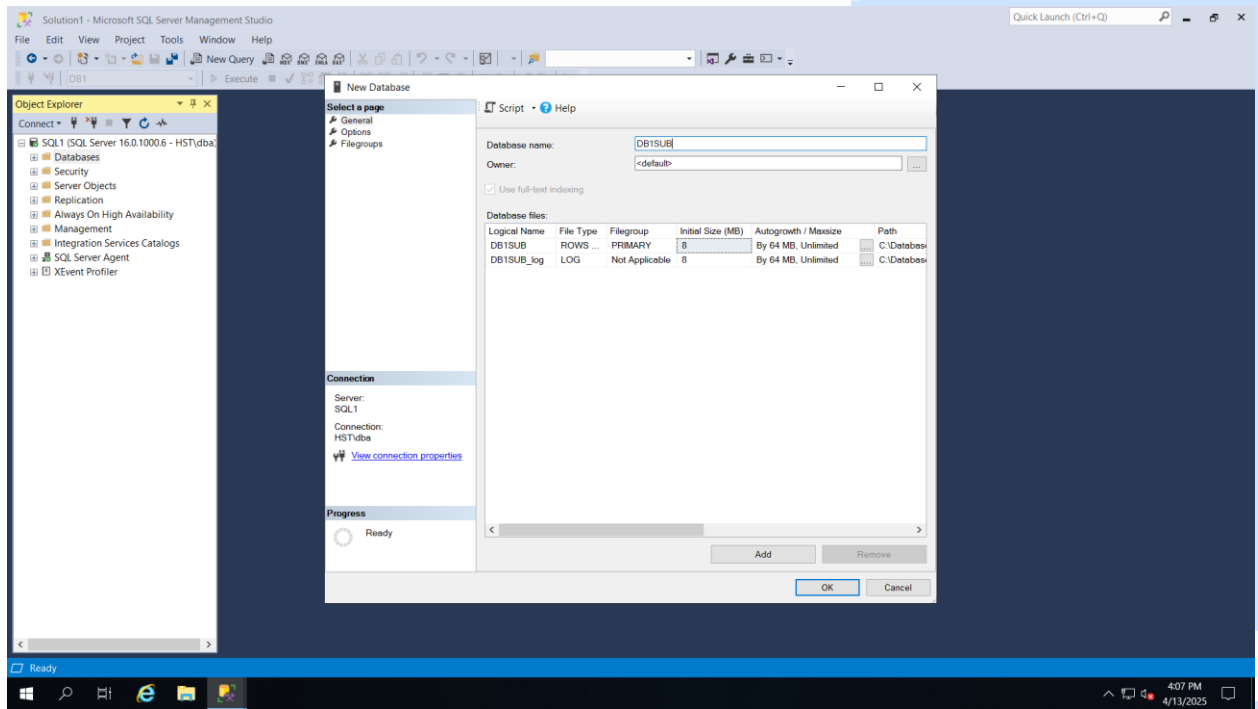
Close

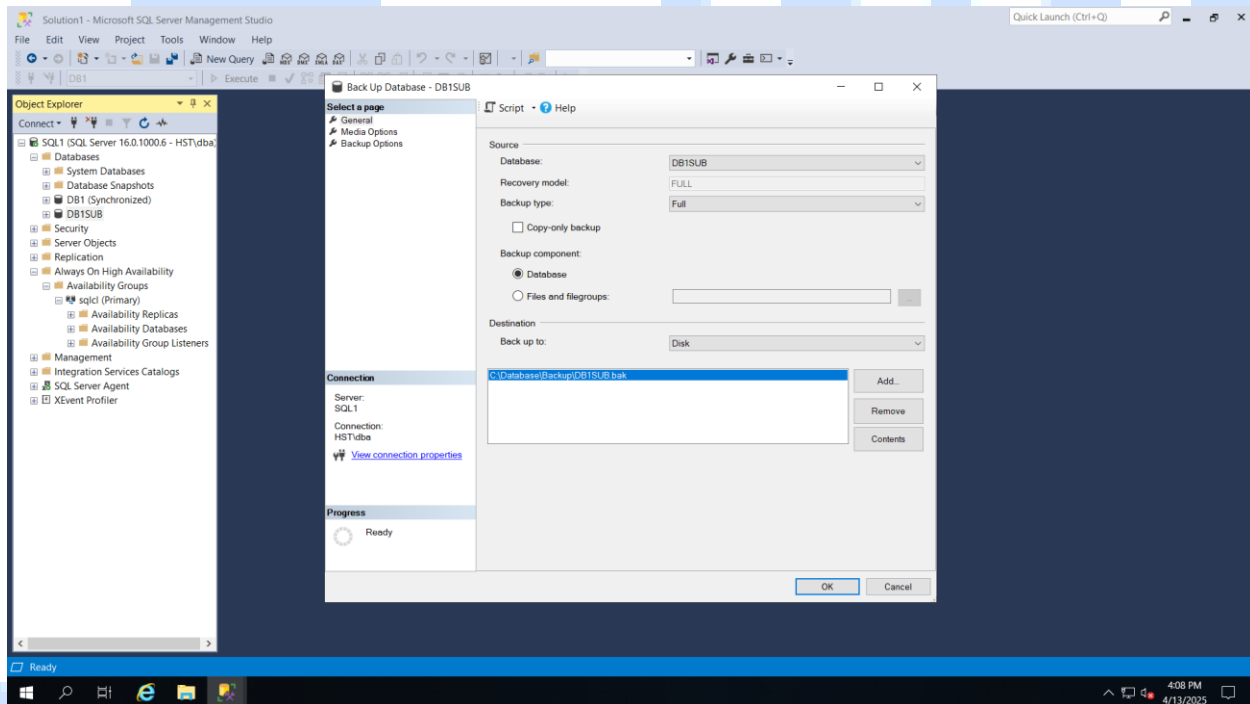
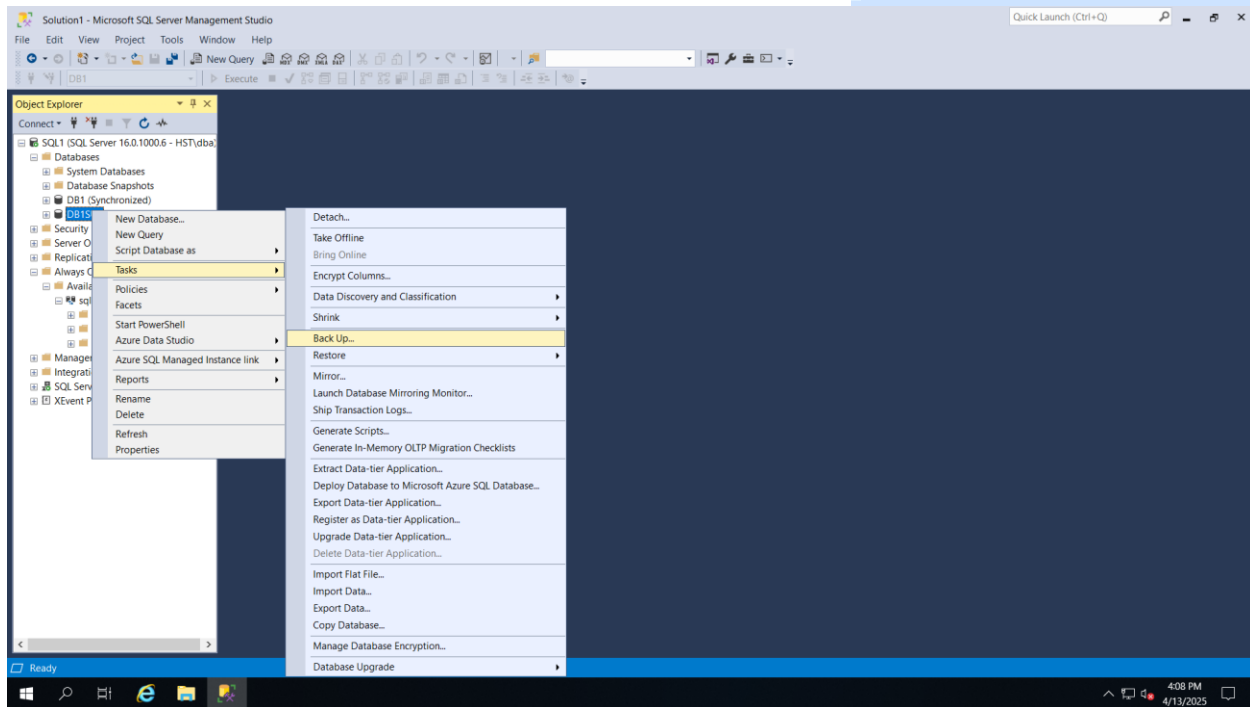
Ready

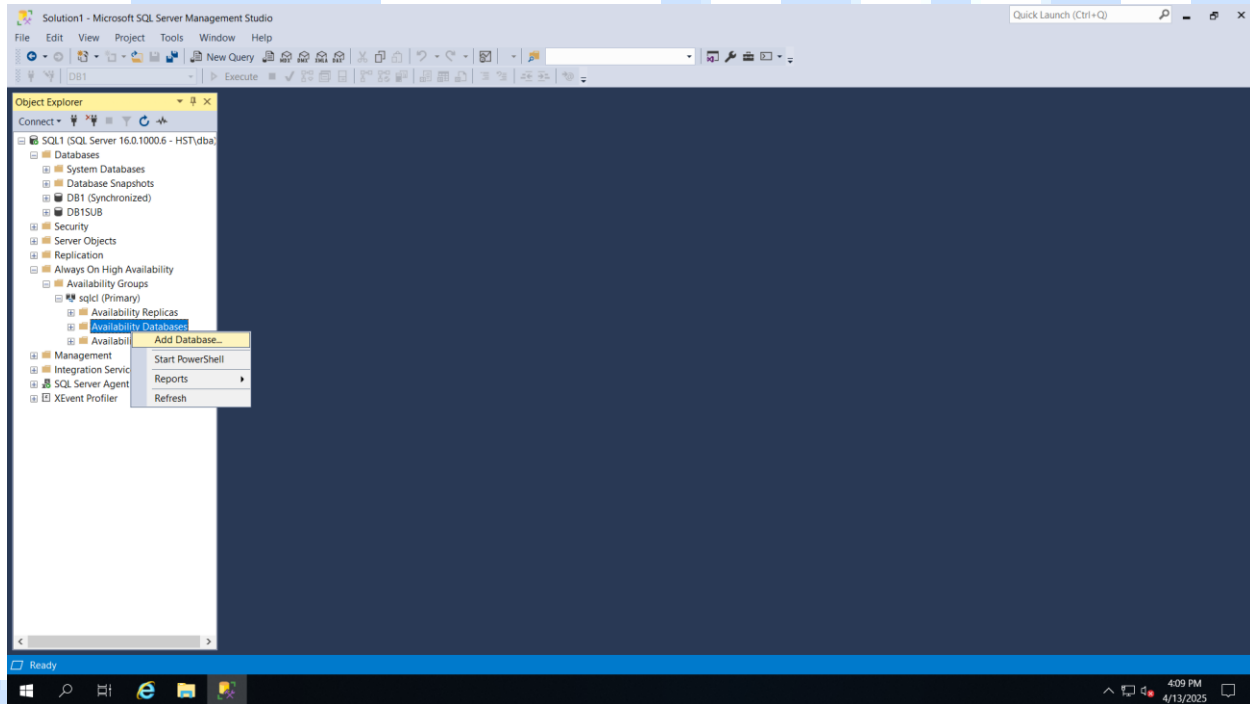
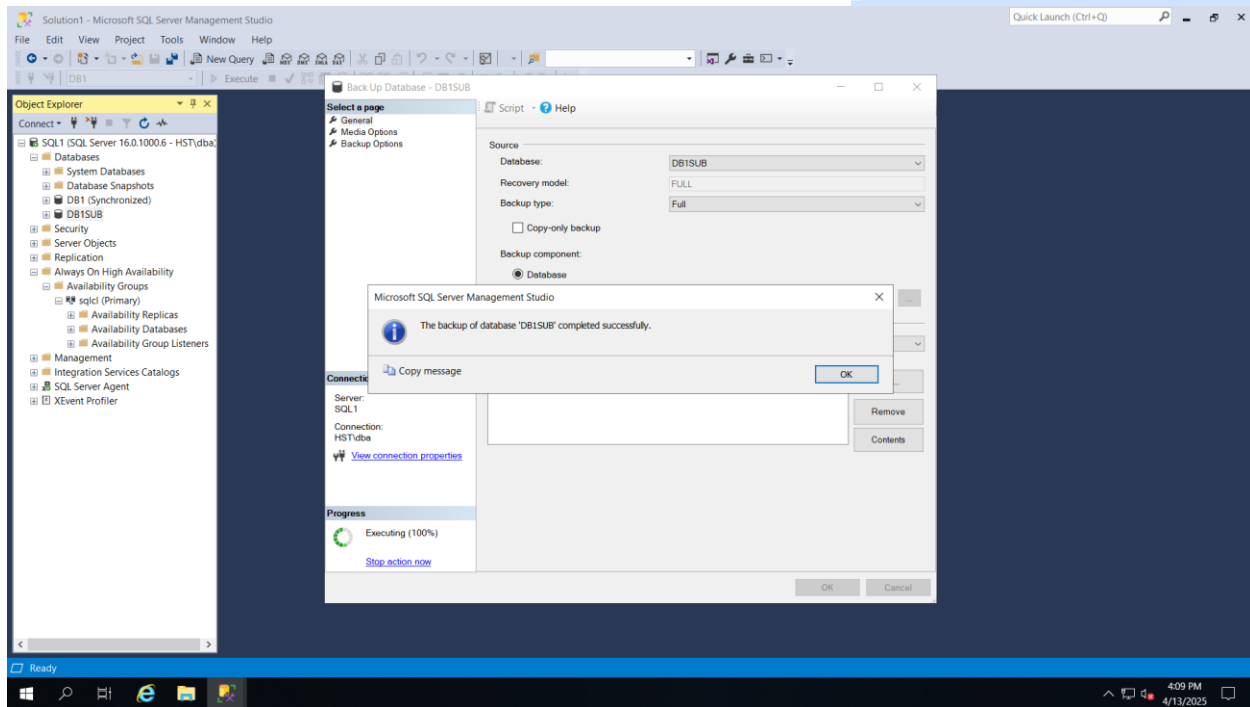
4:00 PM 4/13/2025

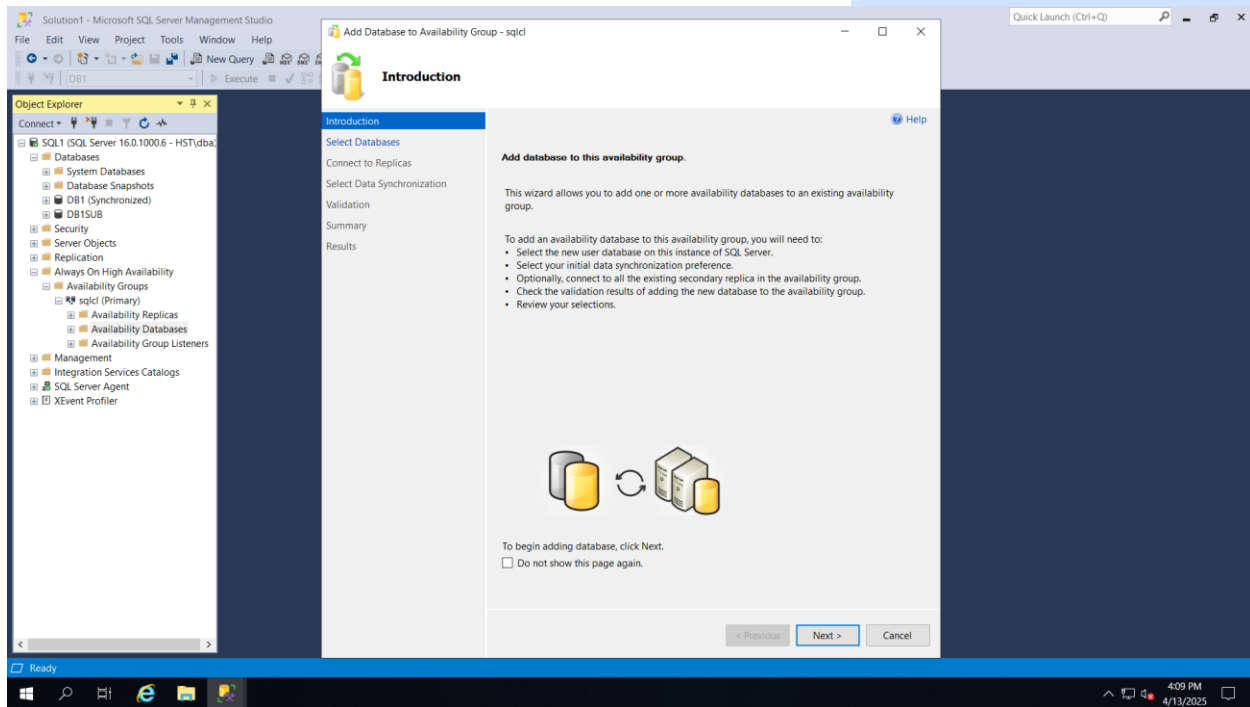
Create Subscriber Database and Add in Always on Cluster Group











Add Database to Availability Group - sqld

Introduction

Add database to this availability group.

This wizard allows you to add one or more availability databases to an existing availability group.

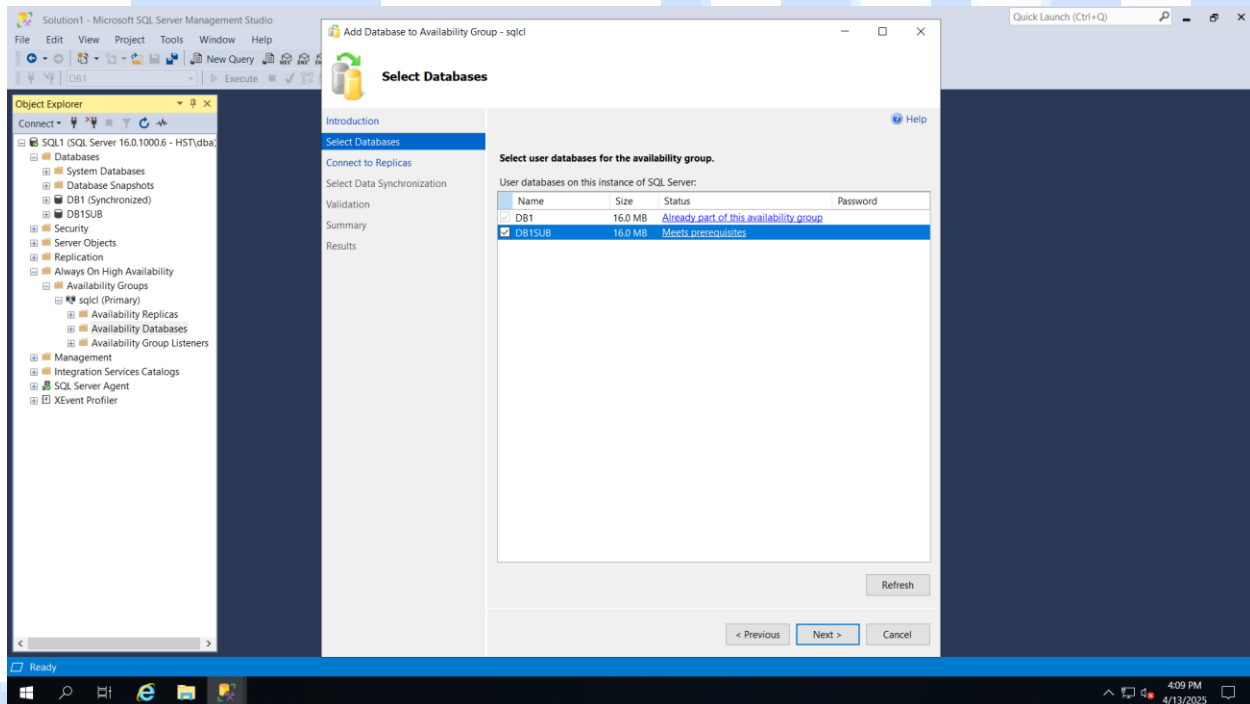
To add an availability database to this availability group, you will need to:

- Select the new user database on this instance of SQL Server.
- Select your initial data synchronization preference.
- Optionally, connect to all the existing secondary replica in the availability group.
- Check the validation results of adding the new database to the availability group.
- Review your selections.

To begin adding database, click Next.

☐ Do not show this page again.

< Previous Next > Cancel



Add Database to Availability Group - sqld

Select Databases

Select user databases for the availability group.

User databases on this instance of SQL Server:

Name	Size	Status	Password
DB1	16.0 MB	Already part of this availability group	
☒ DB1SUB	16.0 MB	Meets prerequisites	

Refresh

< Previous Next > Cancel

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Object Explorer

- Connect
- SQL1 (SQL Server 16.0.1000.6 - HST\ydba)
- Databases
 - System Databases
 - Database Snapshots
 - DB1 (Synchronized)
 - DB1SUB
- Security
- Server Objects
- Replication
- Always On High Availability
 - Availability Groups
 - sqlcl (Primary)
 - Availability Replicas
 - Availability Databases
 - Management
 - Integration Services Catalogs
 - SQL Server Agent
 - XEvent Profiler

Add Database to Availability Group - sqlcl

Connect to Existing Secondary Replicas

Introduction

Select Databases

Connect to Replicas

Select Data Synchronization

Validation

Summary

Results

Connect to all the existing secondary replicas.

Before the wizard can configure existing endpoints to grant them appropriate permissions, you must connect to all the existing secondary replicas.

Server Instance	Connected As	
SQL2	Not Connected	Connect...
SQL3	Not Connected	Connect...

Connect All...

< Previous Next > Cancel

Ready

4:10 PM 4/13/2025

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Object Explorer

- Connect
- SQL1 (SQL Server 16.0.1000.6 - HST\ydba)
- Databases
 - System Databases
 - Database Snapshots
 - DB1 (Synchronized)
 - DB1SUB
- Security
- Server Objects
- Replication
- Always On High Availability
 - Availability Groups
 - sqlcl (Primary)
 - Availability Replicas
 - Availability Databases
 - Management
 - Integration Services Catalogs
 - SQL Server Agent
 - XEvent Profiler

Add Database to Availability Group - sqlcl

Connect to Existing Secondary Replicas

Introduction

Select Databases

Connect to Replicas

Select Data Synchronization

Validation

Summary

Results

Connect to Server

SQL Server

Login Connection Properties Always Encrypted Additional Connection Parameters

Server

Server type: Database Engine

Server name: SQL2

Authentication: Windows Authentication

User name: HST\ydba

Password:

☐ Remember password

Connection Security

Encryption: Optional

☐ Trust server certificate

Host name in certificate:

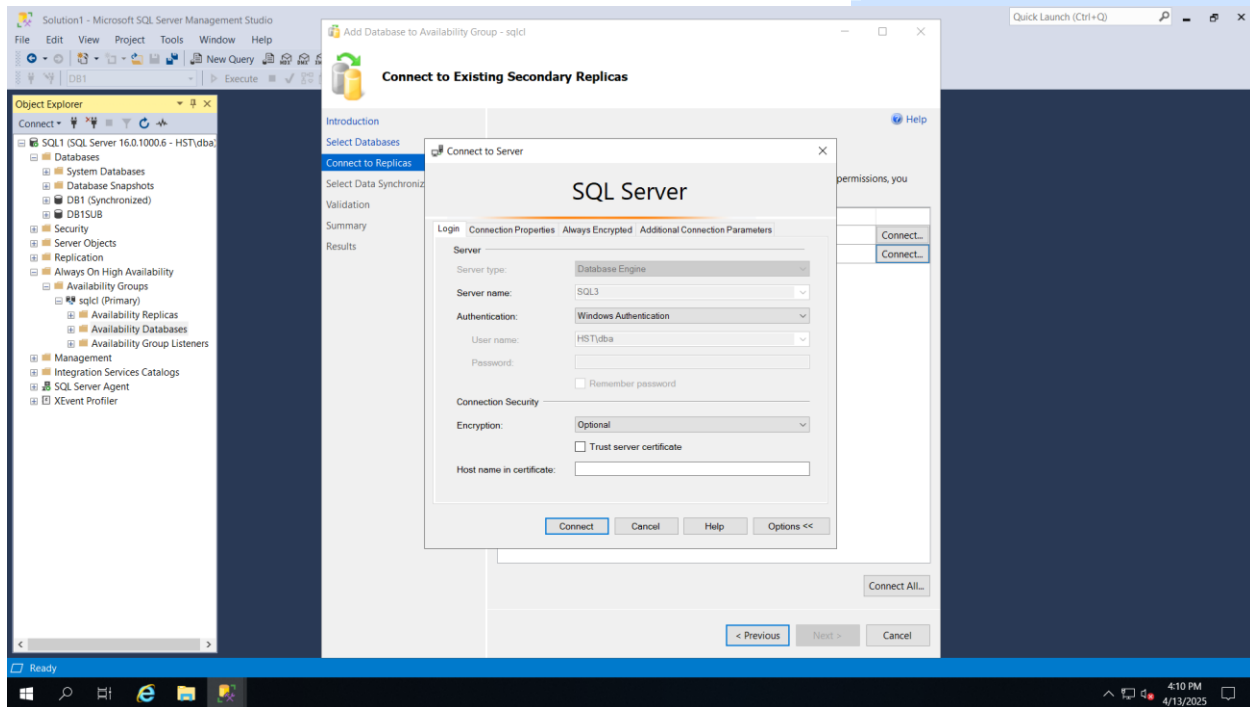
Connect Cancel Help Options <<

Connect All...

< Previous Next > Cancel

Ready

4:10 PM 4/13/2025



Connect to Existing Secondary Replicas

Introduction
Select Databases
Connect to Replicas
Select Data Synchronization
Validation
Summary
Results

Connect to Server

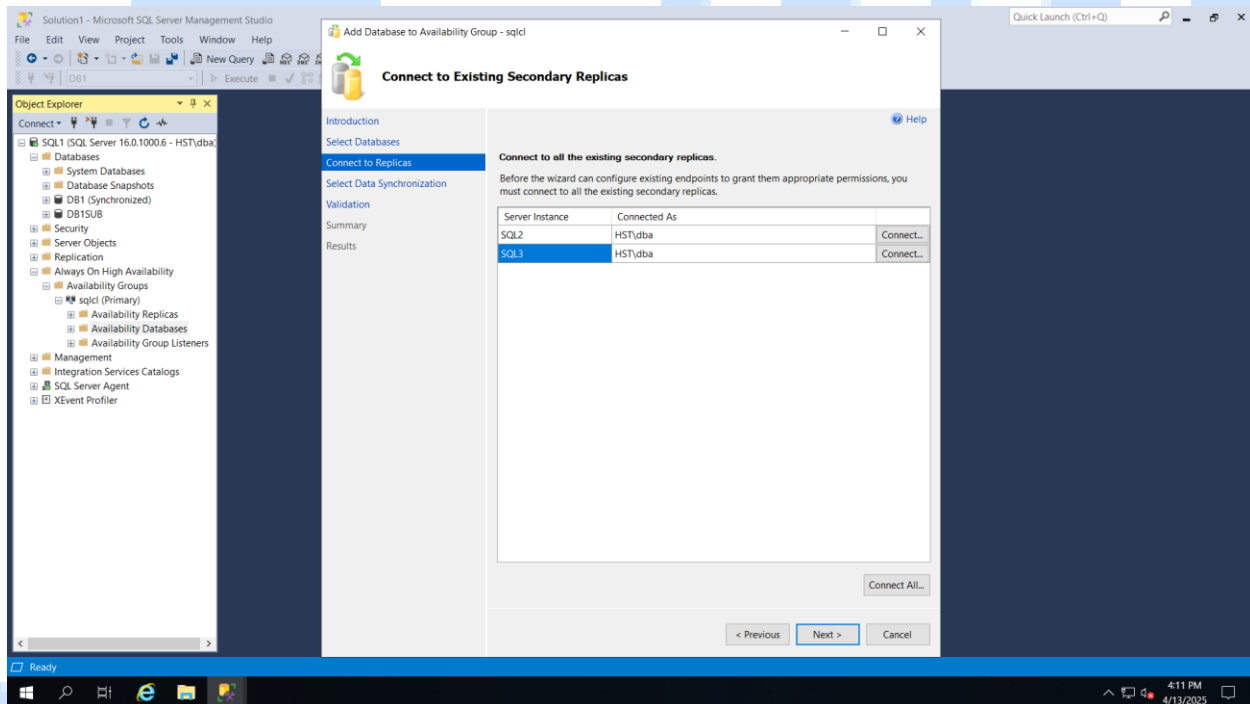
SQL Server

Server type: Database Engine
Server name: SQL3
Authentication: Windows Authentication
User name: HSTydba
Password:
☐ Remember password
Connection Security:
Encryption: Optional
☐ Trust server certificate
Host name in certificate:

Connect Cancel Help Options <<

Connect All...

< Previous Next > Cancel



Connect to Existing Secondary Replicas

Introduction
Select Databases
Connect to Replicas
Select Data Synchronization
Validation
Summary
Results

Connect to all the existing secondary replicas.
Before the wizard can configure existing endpoints to grant them appropriate permissions, you must connect to all the existing secondary replicas.

Server Instance	Connected As	Connect...
SQL2	HSTydba	Connect...
SQL3	HSTydba	Connect...

Connect All...

< Previous Next > Cancel

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect - SQL1 (SQL Server 16.0.1000.6 - HST\yba)

Databases

- System Databases
- Database Snapshots
- DB1 (Synchronized)
- DB1SUB

Security

Server Objects

Replication

Always On High Availability

- Availability Groups
 - sqlcl (Primary)
 - Availability Replicas
 - Availability Databases

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

Add Database to Availability Group - sqlcl

Select Initial Data Synchronization

Introduction

Select Databases

Connect to Replicas

Select Data Synchronization

Validation

Summary

Results

Select your data synchronization preference.

☒ **Automatic seeding**

SQL Server automatically creates databases for every selected secondary replica. Automatic seeding requires that the data and log file paths are the same on every SQL Server instance participating in the availability group.

☐ **Full database and log backup**

Starts data synchronization by performing full database and log backups for each selected database. These databases are restored to each secondary and joined to the availability group. Make sure the file share is accessible to all replicas and is mounted to the same directory on all Linux replicas.

Specify the file share path in Windows format:

Specify the file share location in Linux format:

☐ **Join only**

Starts data synchronization where you have already restored database and log backups to each secondary server. The selected databases are joined to the availability group on each secondary.

☐ **Skip initial data synchronization**

Choose this option if you want to perform your own database and log backups of each primary database.

< Previous Next > Cancel

Ready

4:11 PM 4/13/2025

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect - SQL1 (SQL Server 16.0.1000.6 - HST\yba)

Databases

- System Databases
- Database Snapshots
- DB1 (Synchronized)
- DB1SUB

Security

Server Objects

Replication

Always On High Availability

- Availability Groups
 - sqlcl (Primary)
 - Availability Replicas
 - Availability Databases

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

Add Database to Availability Group - sqlcl

Validation

Introduction

Select Databases

Connect to Replicas

Select Data Synchronization

Validation

Summary

Results

Results of availability group validation.

Name	Result
Checking for free disk space on the server instance that hosts secondary replica SQ...	Success
Checking if the selected databases already exist on the server instance that hosts s...	Success
Checking for the existence of the database files on the server instance that hosts s...	Success
Checking for compatibility of the database file locations on the server instance th...	Success
Checking for free disk space on the server instance that hosts secondary replica SQ...	Success
Checking if the selected databases already exist on the server instance that hosts s...	Success
Checking for the existence of the database files on the server instance that hosts s...	Success
Checking for compatibility of the database file locations on the server instance th...	Success

Re-run Validation

< Previous Next > Cancel

Ready

4:11 PM 4/13/2025

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

DB1

Object Explorer

- SQL1 (SQL Server 16.0.1000.6 - HST\dba)
 - Databases
 - System Databases
 - Database Snapshots
 - DB1 (Synchronized)
 - DB1SUB
 - Security
 - Server Objects
 - Replication
 - Always On High Availability
 - Availability Groups
 - sqlcl (Primary)
 - Availability Replicas
 - Availability Databases
 - Availability Group Listeners
 - Management
 - Integration Services Catalogs
 - SQL Server Agent
 - XEvent Profiler

Add Database to Availability Group - sqlcl

Summary

Introduction
Select Databases
Connect to Replicas
Select Data Synchronization
Validation
Summary
Results

Verify the choices made in this wizard.
Click Finish to perform the following actions:

- Availability group: sqlcl
- Databases
 - DB1SUB (16.0 MB)
- Initial data synchronization: Automatic Seeding

Script

< Previous Finish Cancel

Ready

4:12 PM 4/13/2025

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

DB1

Object Explorer

- SQL1 (SQL Server 16.0.1000.6 - HST\dba)
 - Databases
 - System Databases
 - Database Snapshots
 - DB1 (Synchronized)
 - DB1SUB
 - Security
 - Server Objects
 - Replication
 - Always On High Availability
 - Availability Groups
 - sqlcl (Primary)
 - Availability Replicas
 - Availability Databases
 - Availability Group Listeners
 - Management
 - Integration Services Catalogs
 - SQL Server Agent
 - XEvent Profiler

Add Database to Availability Group - sqlcl

Results

Introduction
Select Databases
Connect to Replicas
Select Data Synchronization
Validation
Summary
Results

The wizard completed successfully.

Summary:

Name	Result
Adding databases to availability group 'sqlcl'.	Success

< Previous Next > Close

Ready

4:12 PM 4/13/2025



sqlci:SQL1 - Microsoft SQL Server Management Studio

Quick Launch (Ctrl+Q)

File Edit View Project Tools Window Help

Object Explorer

Connect

SQL1 (SQL Server 16.0.1000.6 - HST\ydba)

Databases

System Databases

Database Snapshots

DB1 (Synchronized)

DB1SUB

Security

Server Objects

Replication

Always On High Availability

Availability Groups

sqlci (Primary)

Availability Replicas

Availability Databases

Availability Group Listeners

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

sqlci:SQL1

sqlci: hosted by SQL1 (Replica role: Primary)

Last updated: 4/13/2025 4:12:40 PM

Auto refresh: on

Start Failover Wizard

View Always On Health Events

View Cluster Quorum Information

Collect Latency Data

Analyze Log Block Latency

Add/Remove Columns

Availability group state: Healthy

Primary instance: SQL1

Failover mode: Automatic

Cluster state: sqlwcl (Normal Quorum)

Cluster type: Windows Server Failover Cluster

Availability replica:

Name	Role	Availability ...	Failover M...	Seeding M...	Synchronization ...	Issues
SQL1	Prim	Synchronous	Automatic	Automatic	Synchronized	
SQL2	Seco...	Synchronous ...	Automatic	Automatic	Synchronized	
SQL3	Seco...	Synchronous ...	Automatic	Automatic	Synchronized	

Group by

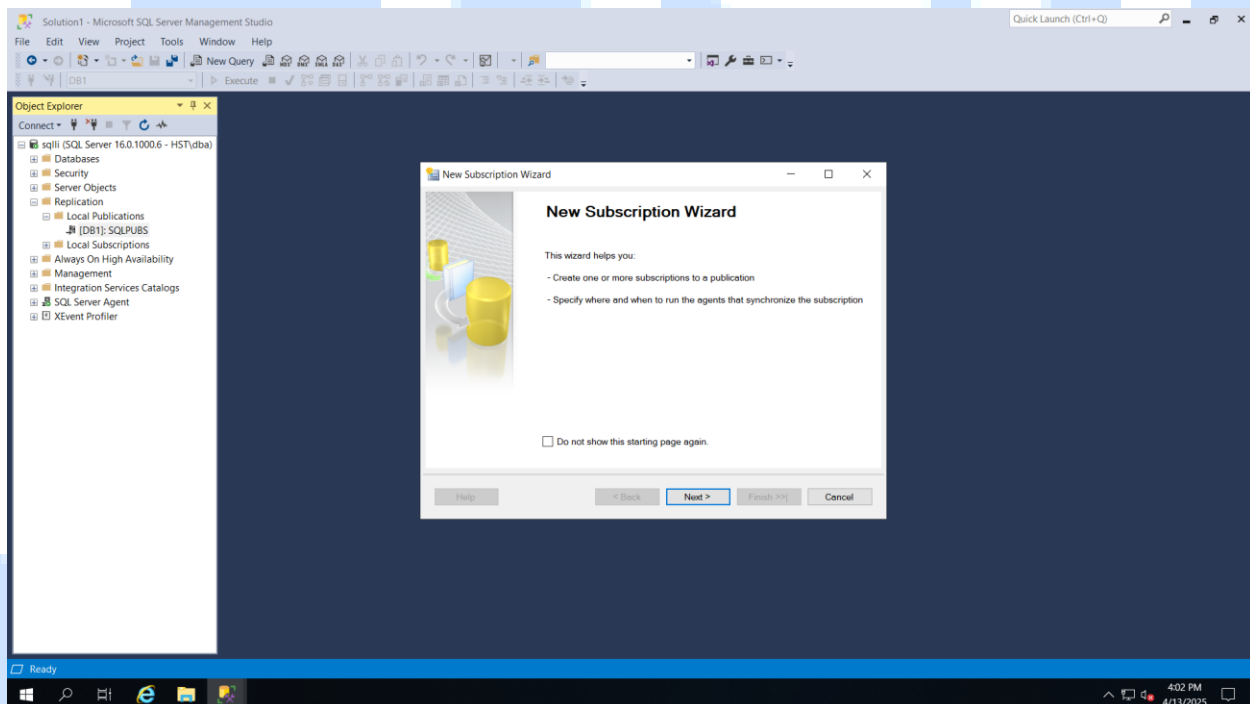
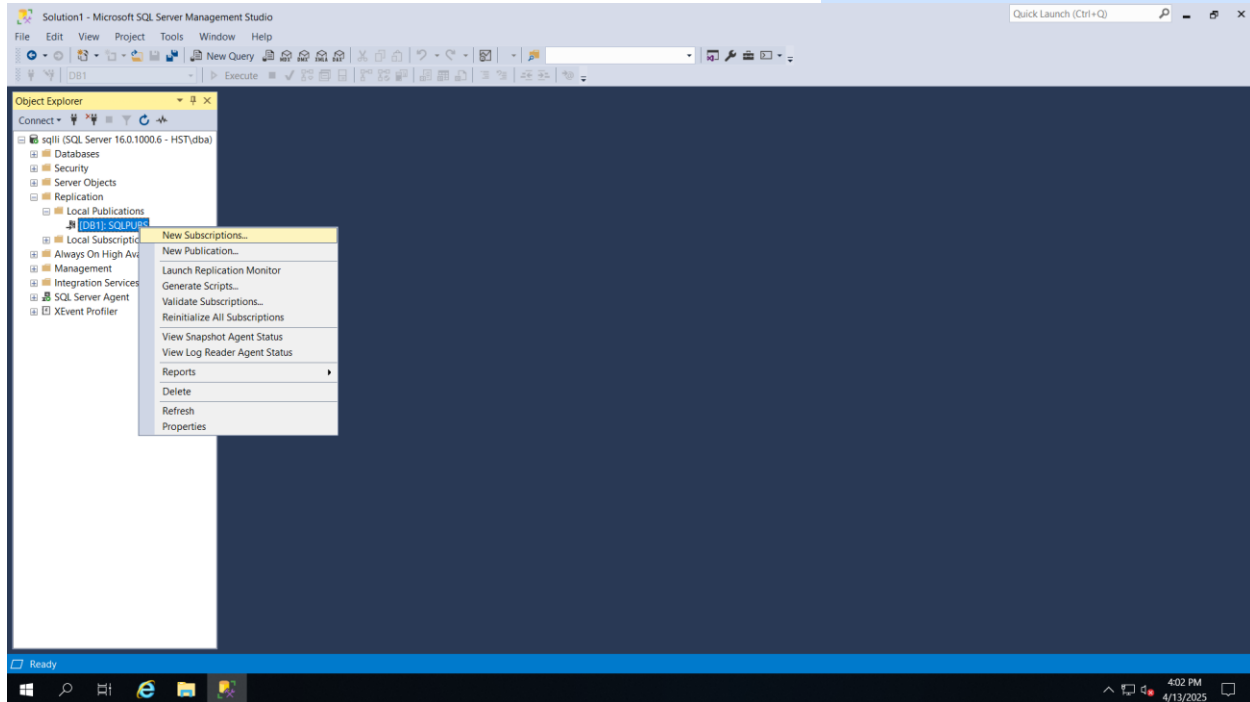
Add/Remove Columns

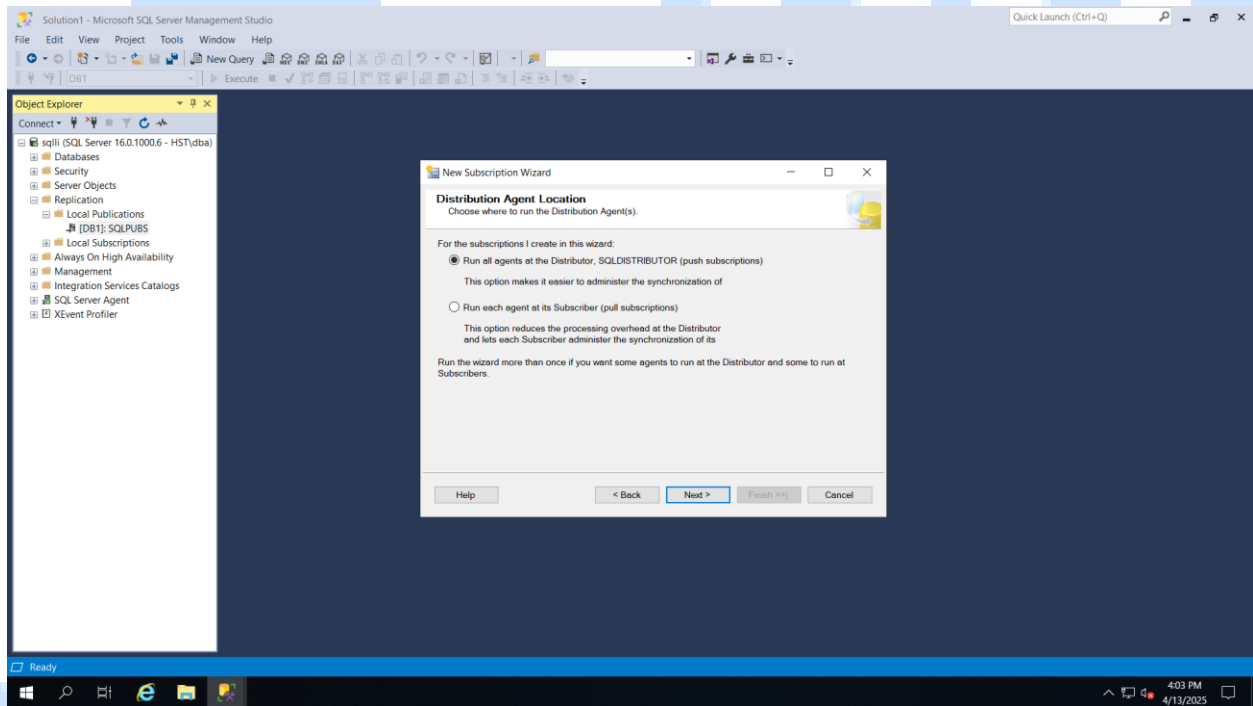
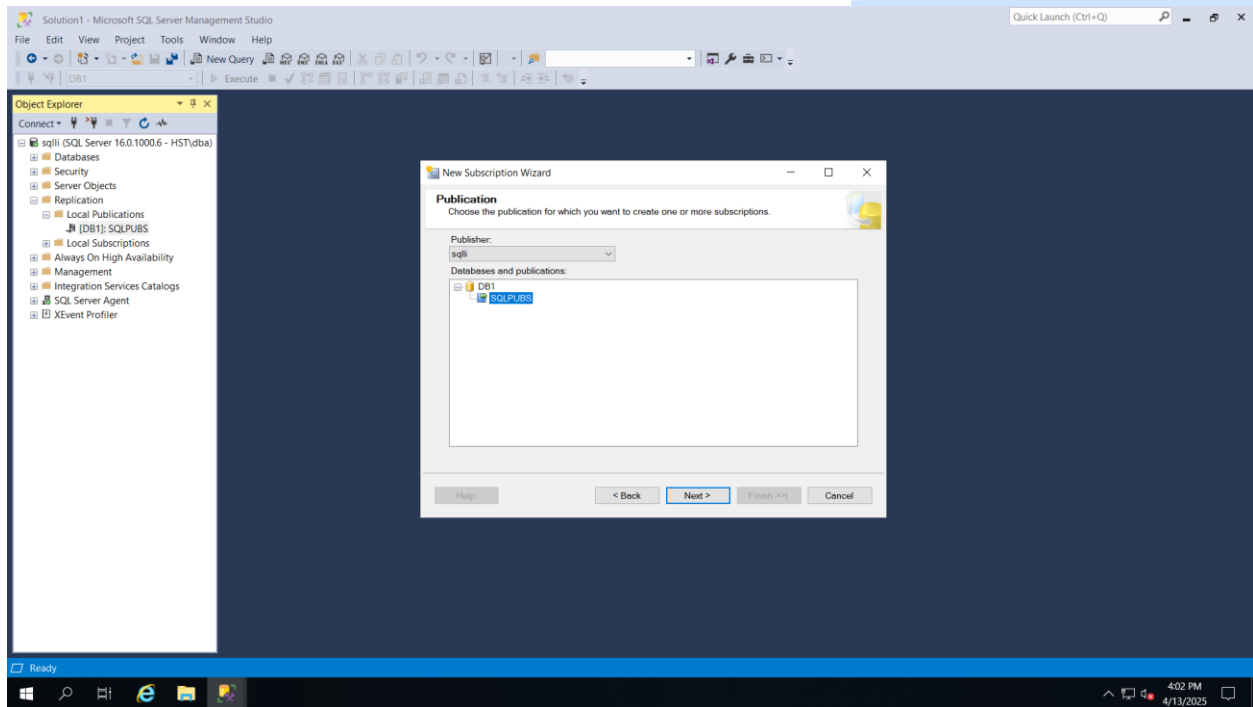
Name	Replica	Synchronization St...	Failover Re...	Issues
SQL1				
DB1	SQL1	Synchronized	No Data Lo...	
DB1SUB	SQL1	Synchronized	No Data Lo...	
SQL2				
DB1	SQL2	Synchronized	No Data Lo...	
DB1SUB	SQL2	Synchronized	No Data Lo...	
SQL3				
DB1	SQL3	Synchronized	No Data Lo...	
DB1SUB	SQL3	Synchronized	No Data Lo...	

Ready

4:12 PM 4/13/2025

Create Subscriber with Always on Listener connection





Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect

SQLLI (SQL Server 16.0.1000.6 - HST\dba)

Databases

Security

Server Objects

Replication

Local Publications

Local Subscriptions

Always On High Availability

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

New Subscription Wizard

Subscribers

Choose one or more Subscribers and specify each subscription database.

Subscribers and subscription databases:

Subscriber	Subscription Database
SQLLI	

Add Subscriber

You must select a Subscriber and specify the subscription database to continue the wizard.

Help < Back Next > Finish >> Cancel

Ready

4:03 PM 4/13/2025

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect

SQLLI (SQL Server 16.0.1000.6 - HST\dba)

Databases

Security

Server Objects

Replication

Local Publications

Local Subscriptions

Always On High Availability

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

New Subscription Wizard

Subscribers

Choose one or more Subscribers and specify each subscription database.

Subscribers and subscription databases:

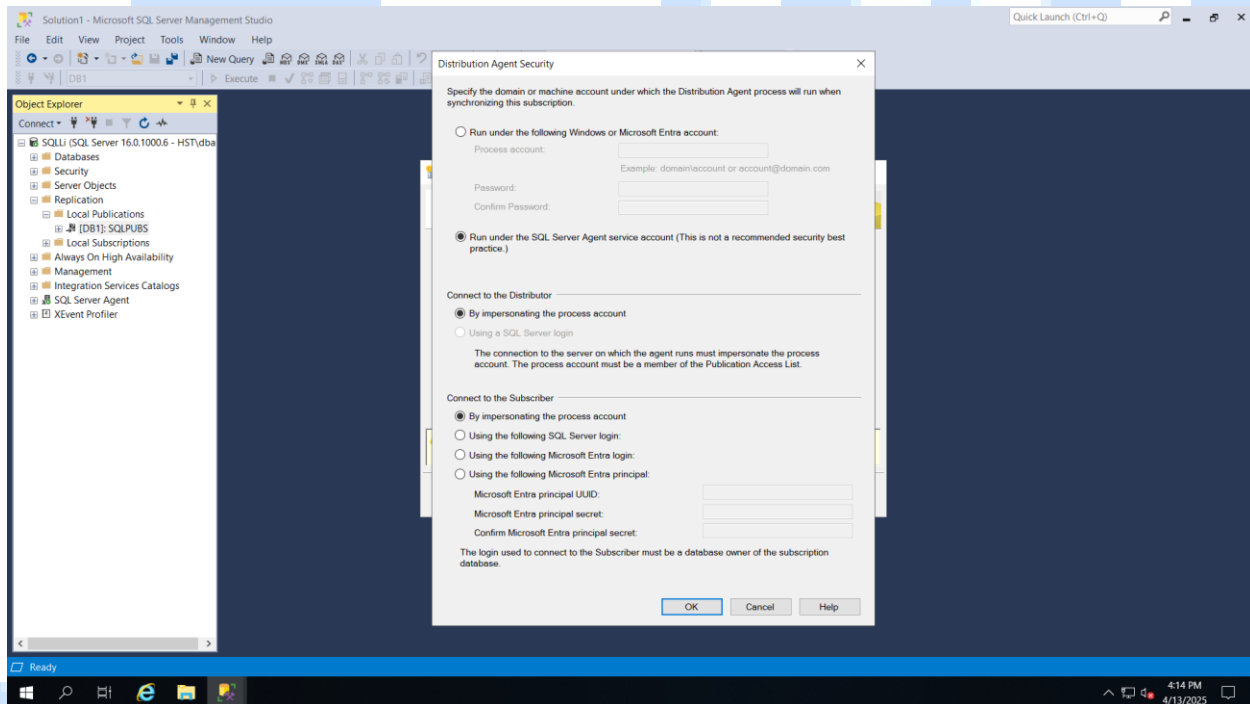
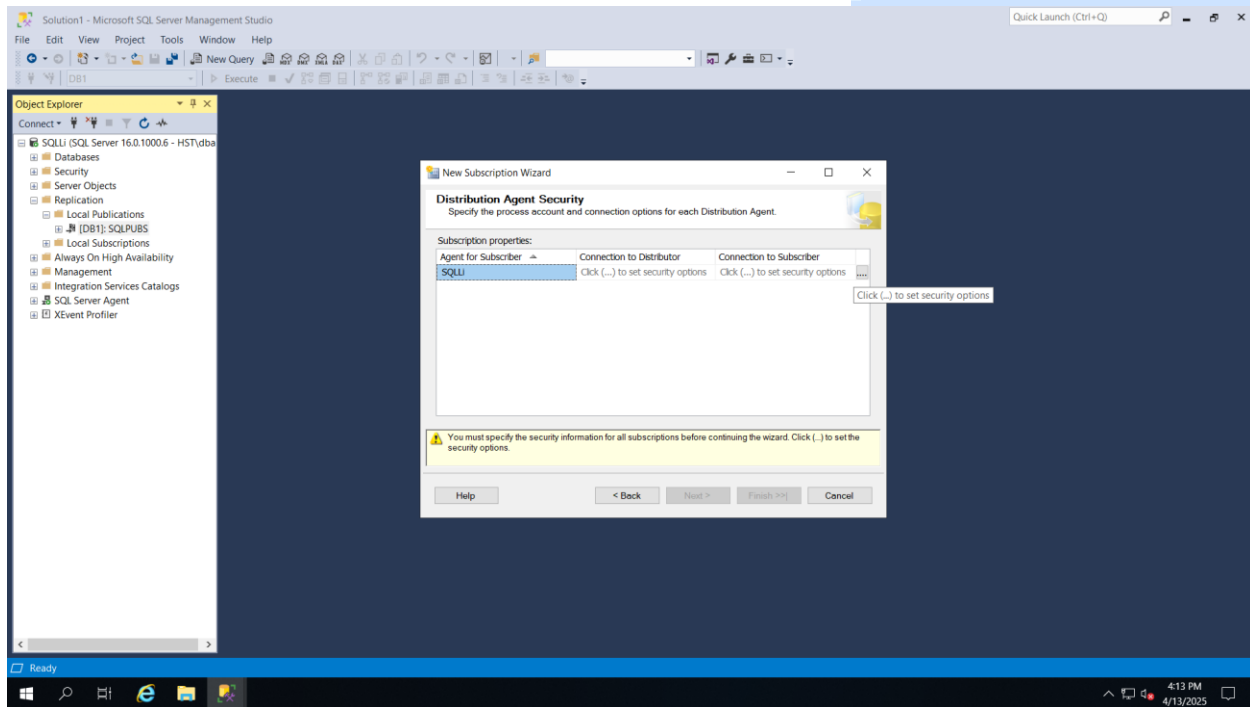
Subscriber	Subscription Database
☒ SQLLI	GOLESUB

Add Subscriber

Help < Back Next > Finish >> Cancel

Ready

4:13 PM 4/13/2025





Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect

SQLU (SQL Server 16.0.1000.6 - HST\tdba)

Databases

Security

Server Objects

Replication

Local Publications

Local Subscriptions

Always On High Availability

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

New Subscription Wizard

Distribution Agent Security

Specify the process account and connection options for each Distribution Agent.

Subscription properties:

Agent for Subscriber	Connection to Distributor	Connection to Subscriber
SQLU	Impersonate process account	Impersonate process account

Help < Back Next > Finish >> Cancel

Ready

4:14 PM 4/13/2025

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect

SQLU (SQL Server 16.0.1000.6 - HST\tdba)

Databases

Security

Server Objects

Replication

Local Publications

Local Subscriptions

Always On High Availability

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

New Subscription Wizard

Synchronization Schedule

Specify the synchronization schedule for each agent.

Agent schedule:

Subscriber	Agent Location	Agent Schedule
SQLU	Distributor	Run continuously

Help < Back Next > Finish >> Cancel

Ready

4:14 PM 4/13/2025



Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect

SQLLI (SQL Server 16.0.1000.6 - HSTYdba)

Databases

Security

Server Objects

Replication

Local Publications

[DB1]: SQLPUBS

Local Subscriptions

Always On High Availability

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

New Subscription Wizard

Initialize Subscriptions

Specify whether to initialize each subscription with a snapshot of the publication data and

Subscription properties:

Subscriber	Memory Optimized	Initialize	Initialize When
SQLLI	☐	☒	Immediately

A subscription database needs to be initialized with a snapshot of the publication data and schema unless it has already been specially prepared for the subscription.

Help < Back Next > Finish >> Cancel

Ready

4:14 PM 4/13/2025

Solution1 - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connect

SQLLI (SQL Server 16.0.1000.6 - HSTYdba)

Databases

Security

Server Objects

Replication

Local Publications

[DB1]: SQLPUBS

Local Subscriptions

Always On High Availability

Management

Integration Services Catalogs

SQL Server Agent

XEvent Profiler

New Subscription Wizard

Wizard Actions

Choose what happens when you click Finish.

At the end of the wizard:

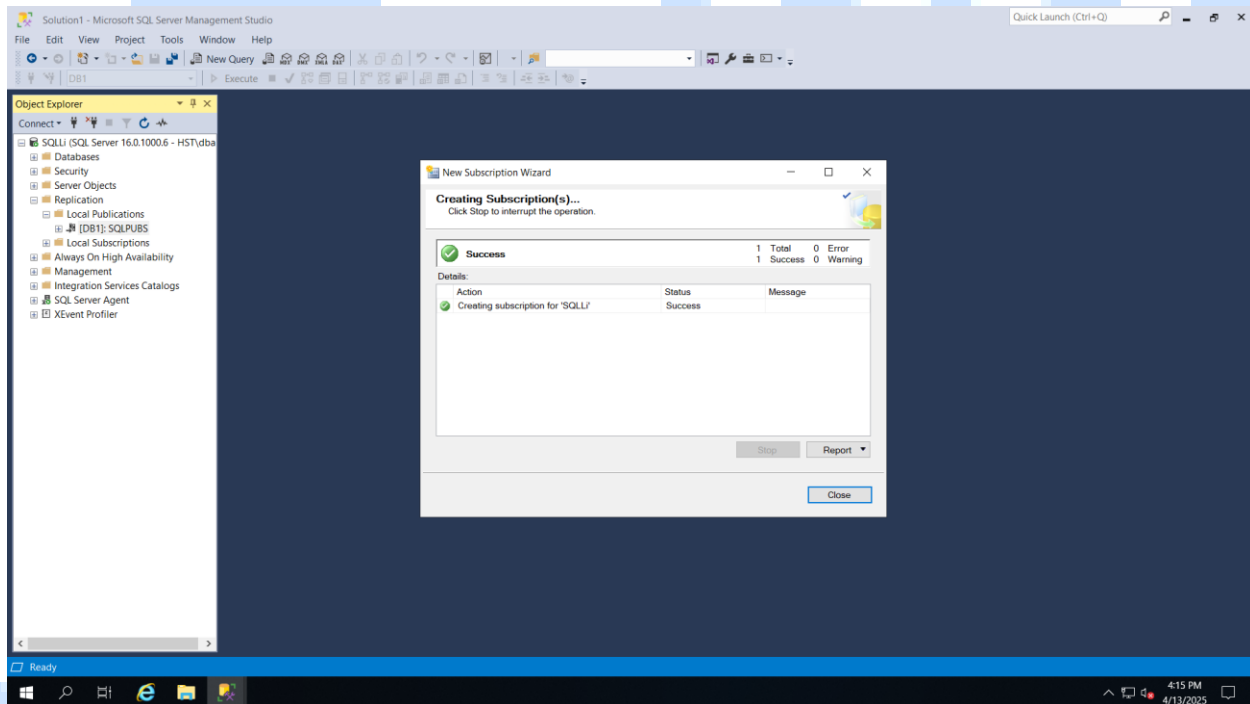
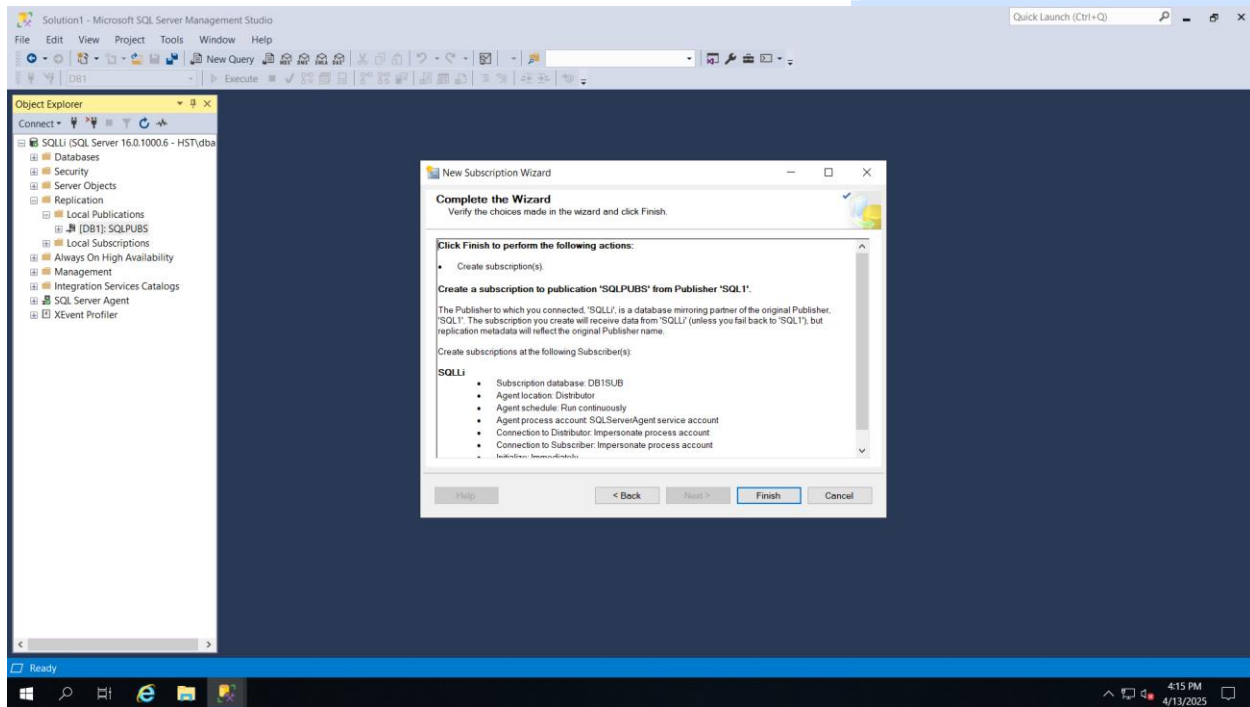
☒ Create the subscription(s)

☐ Generate a script file with steps to create the subscription(s)

Help < Back Next > Finish >> Cancel

Ready

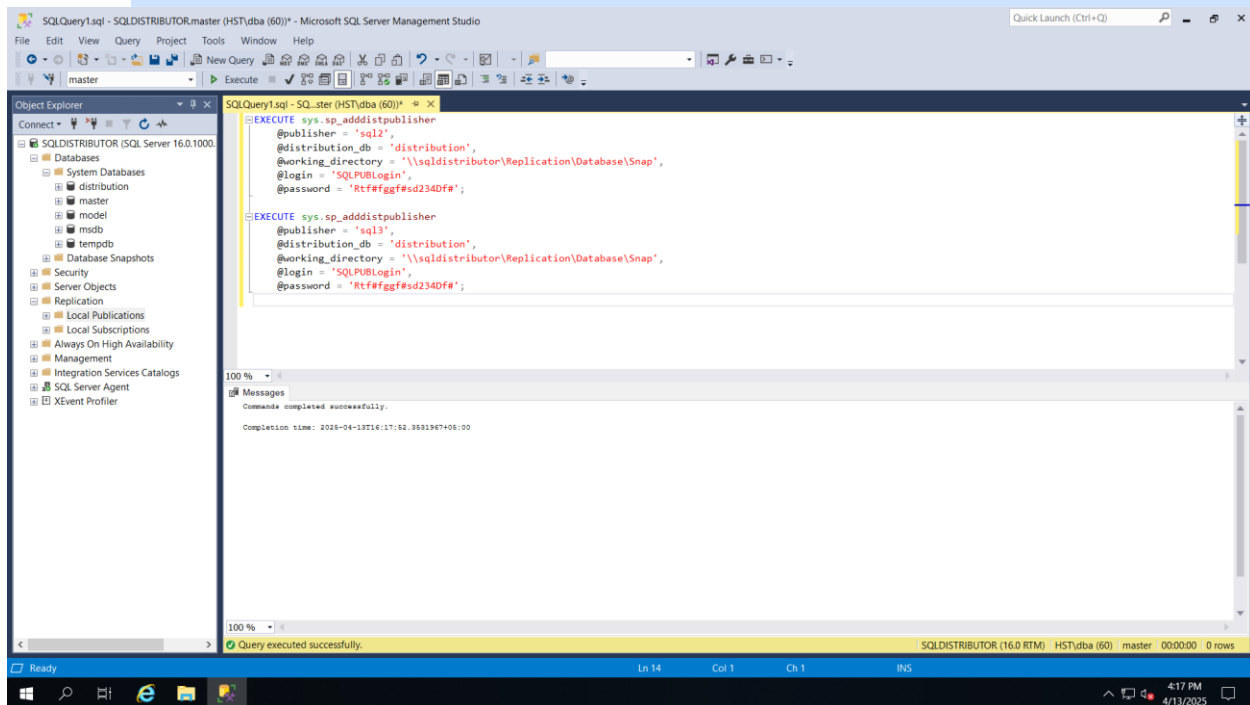
4:15 PM 4/13/2025



Add Secondary Node as Publisher Name at Distributor Server

```
EXECUTE sys.sp_adddistpublisher
    @publisher = 'sql2',
    @distribution_db = 'distribution',
    @working_directory = '\\sql distributor\Replication\Snap',
    @login = 'SQLPUBLogin',
    @password = 'Rtf#fggf#sd234Df#';
```

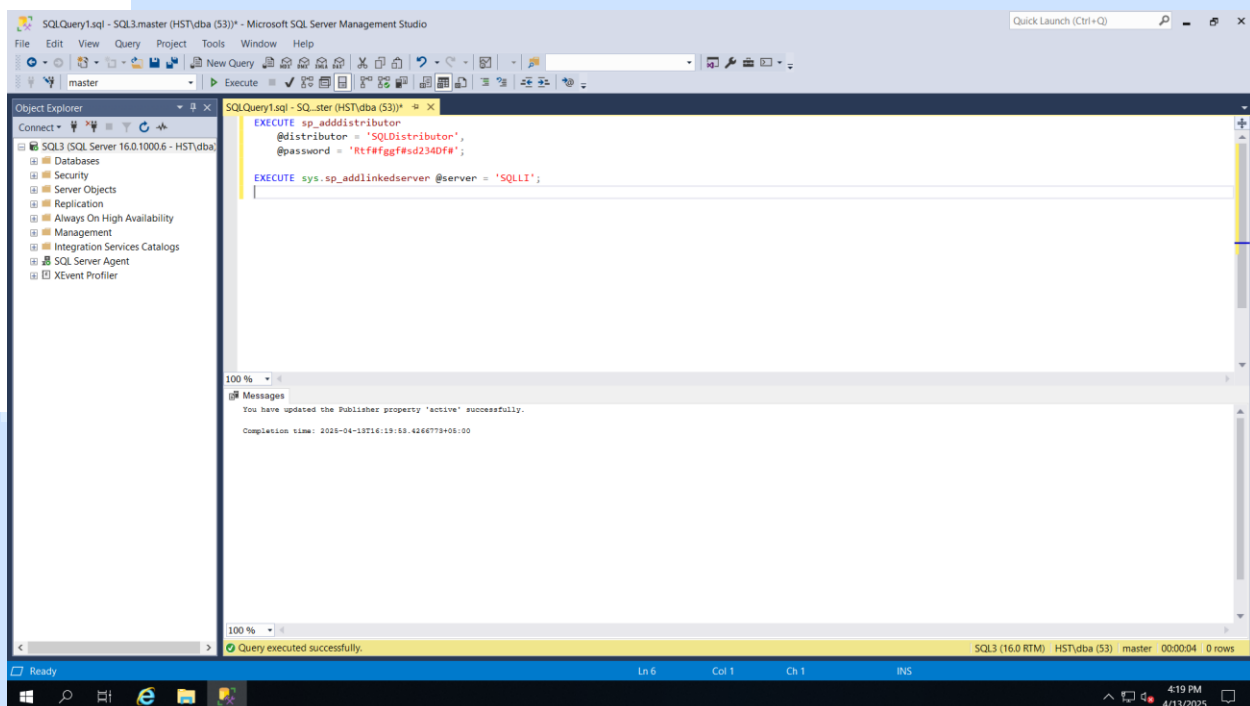
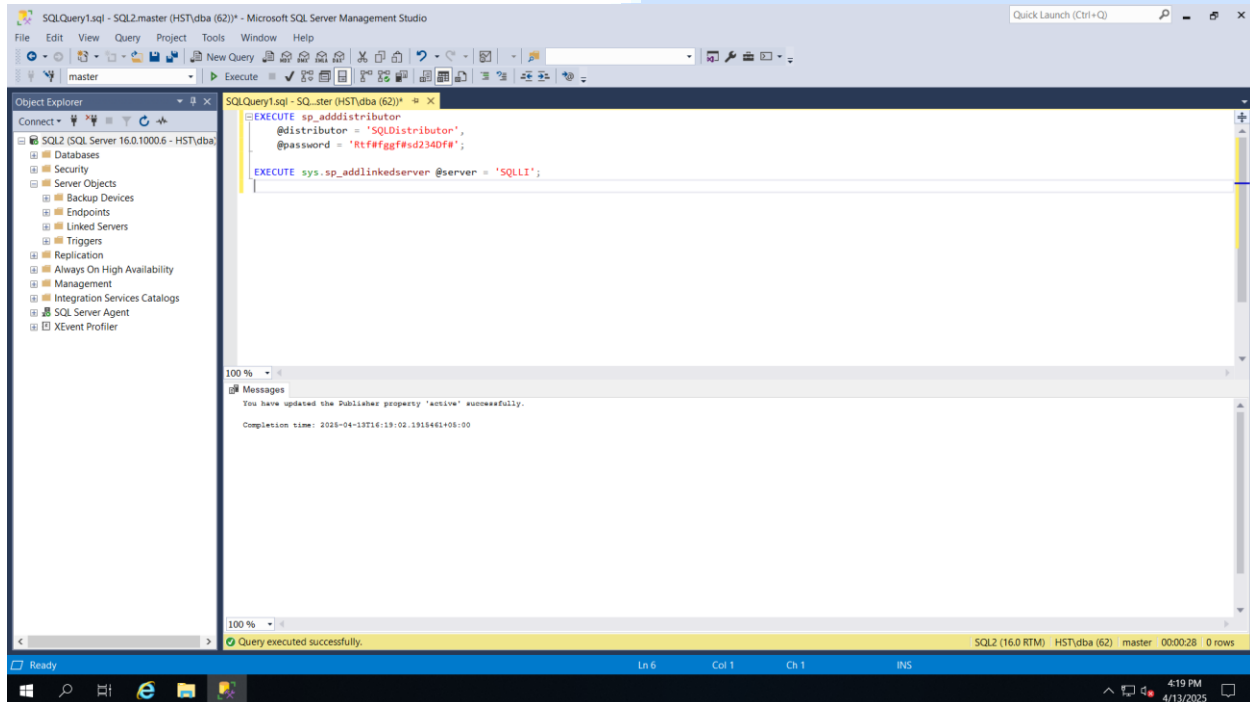
```
EXECUTE sys.sp_adddistpublisher
    @publisher = 'sql3',
    @distribution_db = 'distribution',
    @working_directory = '\\sql distributor\Replication\Snap',
    @login = 'SQLPUBLogin',
    @password = 'Rtf#fggf#sd234Df#';
```



Create Distributor at all Secondary Servers and Create Linked Server

```
EXECUTE sp_adddistributor
    @distributor = 'SQLDistributor',
    @password = 'Rtf#fggf#sd234Df#';
```

```
EXECUTE sys.sp_addlinkedserver @server = 'SQLLI';
```



Define Publisher Re – Direct at Distributor

USE distribution;

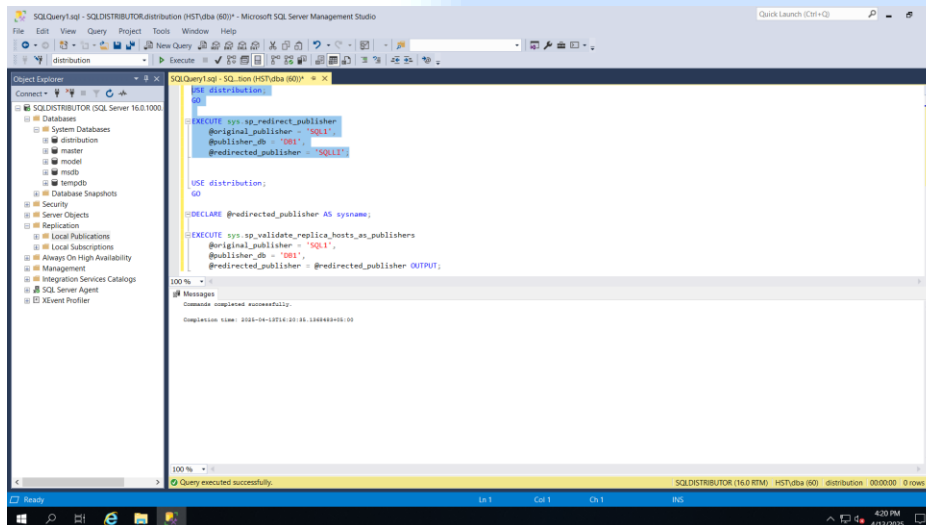
GO

EXECUTE sys.sp_redirect_publisher

@original_publisher = 'SQL1',

@publisher_db = 'DB1',

@redirected_publisher = 'SQLLI';



USE distribution;

GO

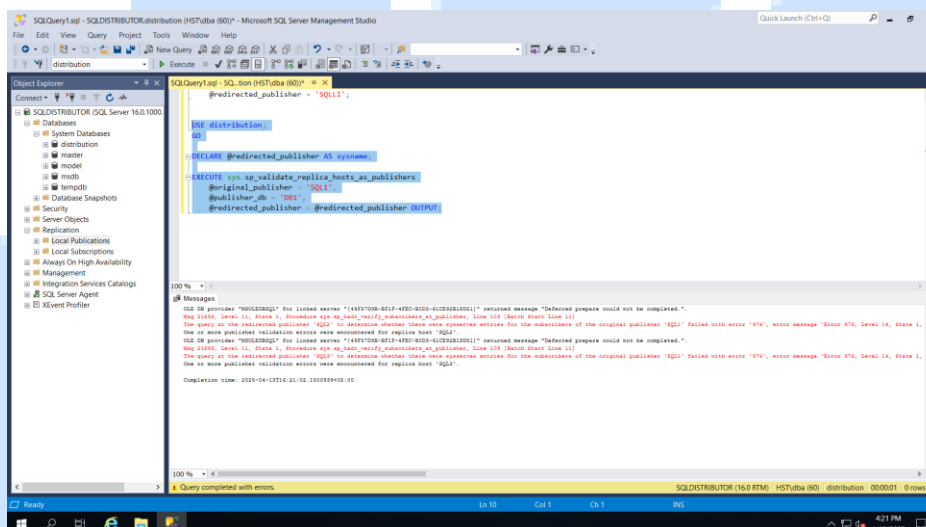
DECLARE @redirected_publisher AS sysname;

EXECUTE sys.sp_validate_replica_hosts_as_publishers

@original_publisher = 'SQL1',

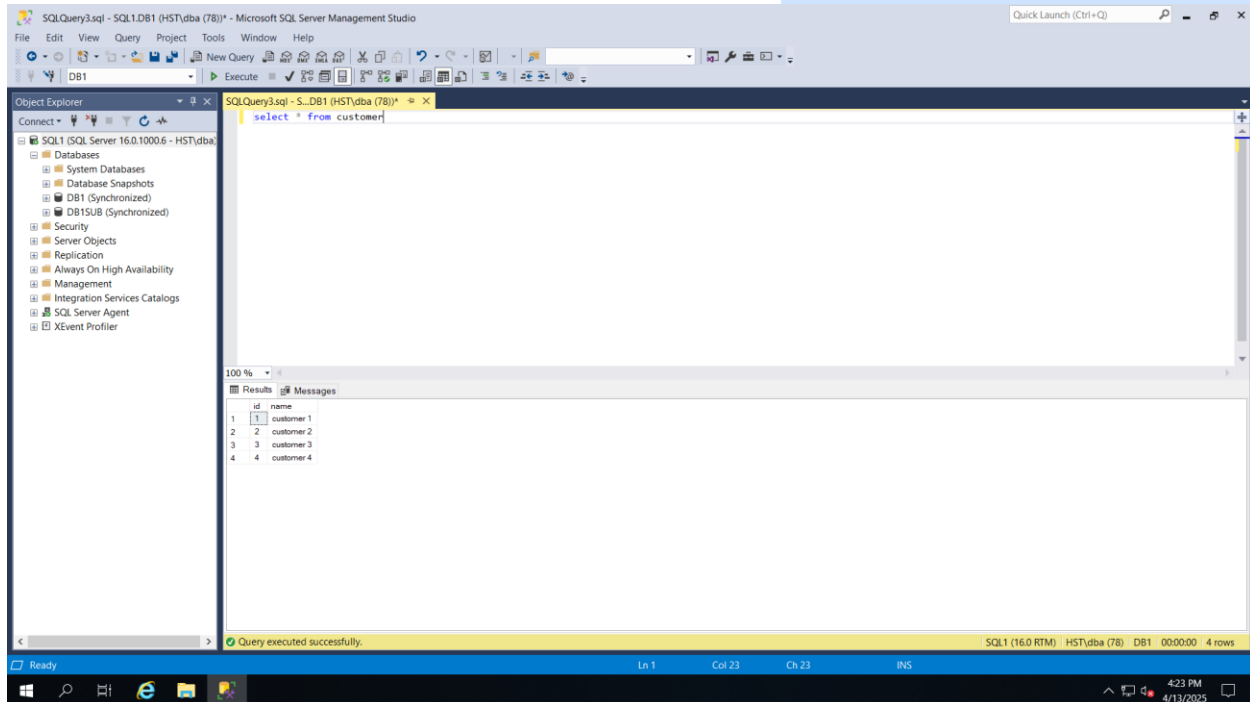
@publisher_db = 'DB1',

@redirected_publisher = @redirected_publisher OUTPUT;



Monitor Replication

**** Verify Table on Both Databases**



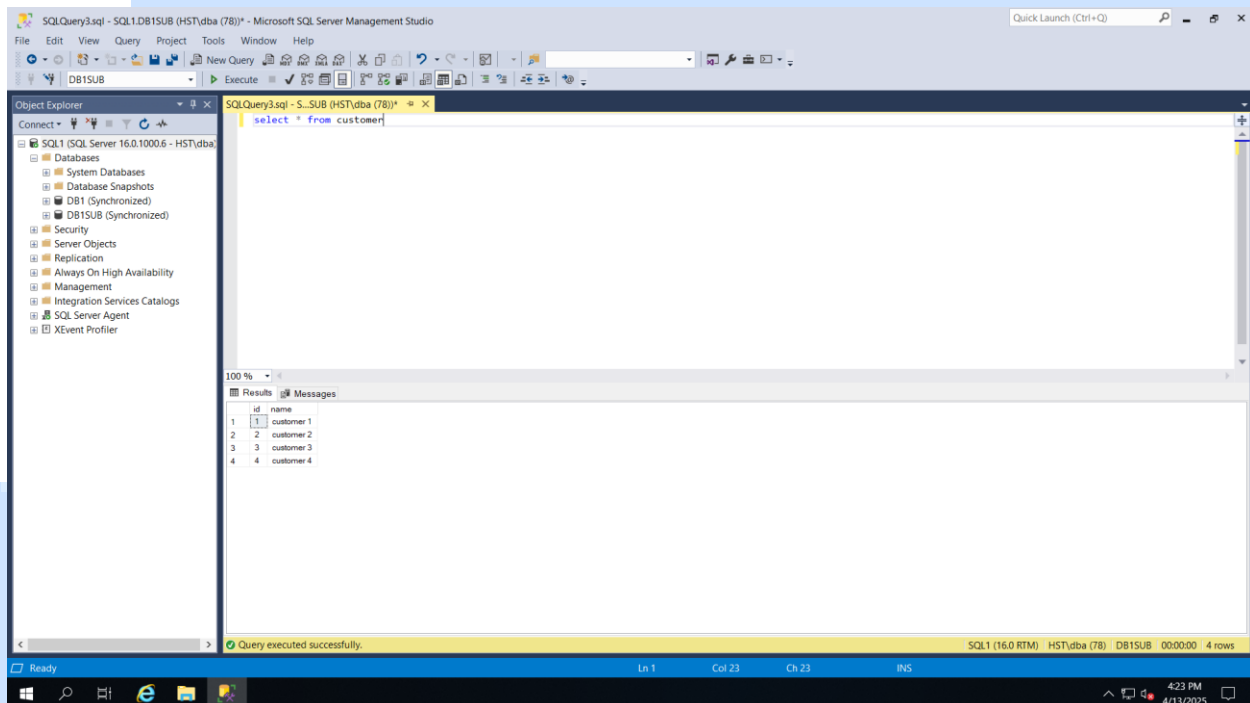
The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'Databases' folder expanded, showing 'DB1 (Synchronized)' and 'DB1SUB (Synchronized)'. The right pane shows a query window with the following SQL code:

```
select * from customer
```

The 'Results' pane displays the following data:

id	name
1	customer 1
2	customer 2
3	customer 3
4	customer 4

The status bar at the bottom indicates 'Query executed successfully.' and 'SQL1 (16.0 RTM) HST\tdba (78) DB1 00:00:00 4 rows'.



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'Databases' folder expanded, showing 'DB1 (Synchronized)' and 'DB1SUB (Synchronized)'. The right pane shows a query window with the following SQL code:

```
select * from customer
```

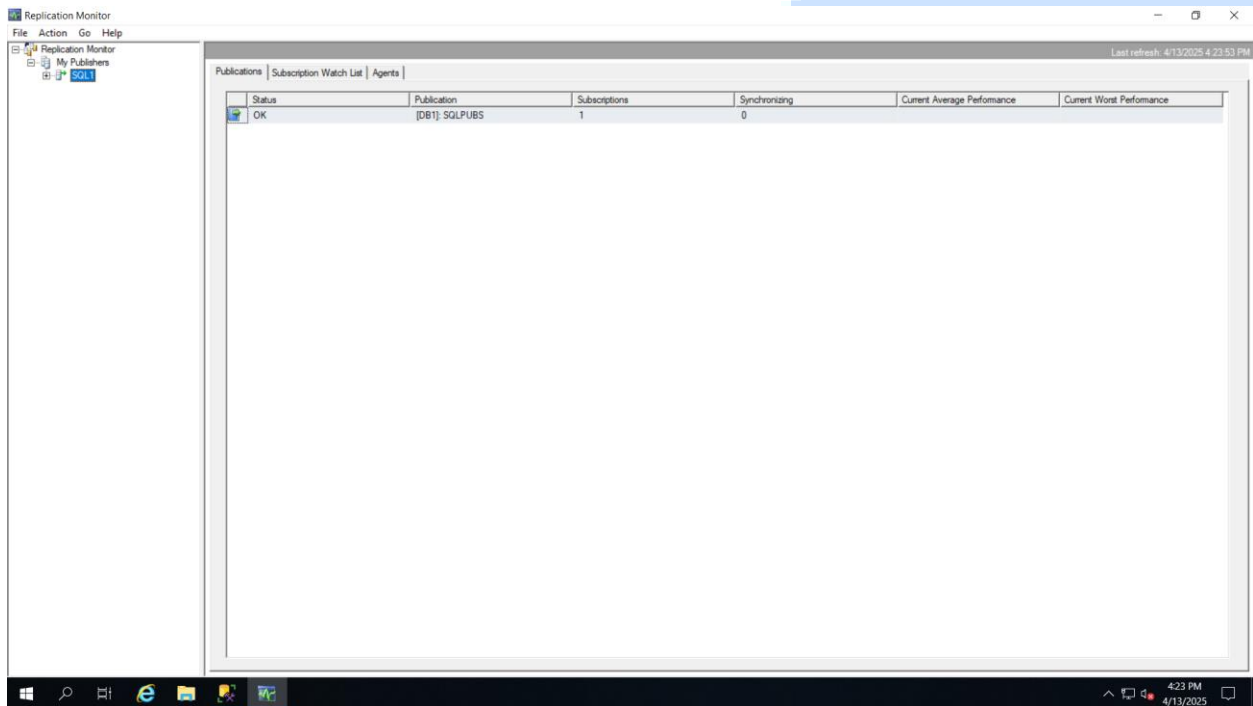
The 'Results' pane displays the following data:

id	name
1	customer 1
2	customer 2
3	customer 3
4	customer 4

The status bar at the bottom indicates 'Query executed successfully.' and 'SQL1 (16.0 RTM) HST\tdba (78) DB1SUB 00:00:00 4 rows'.

**** Verify Agents**

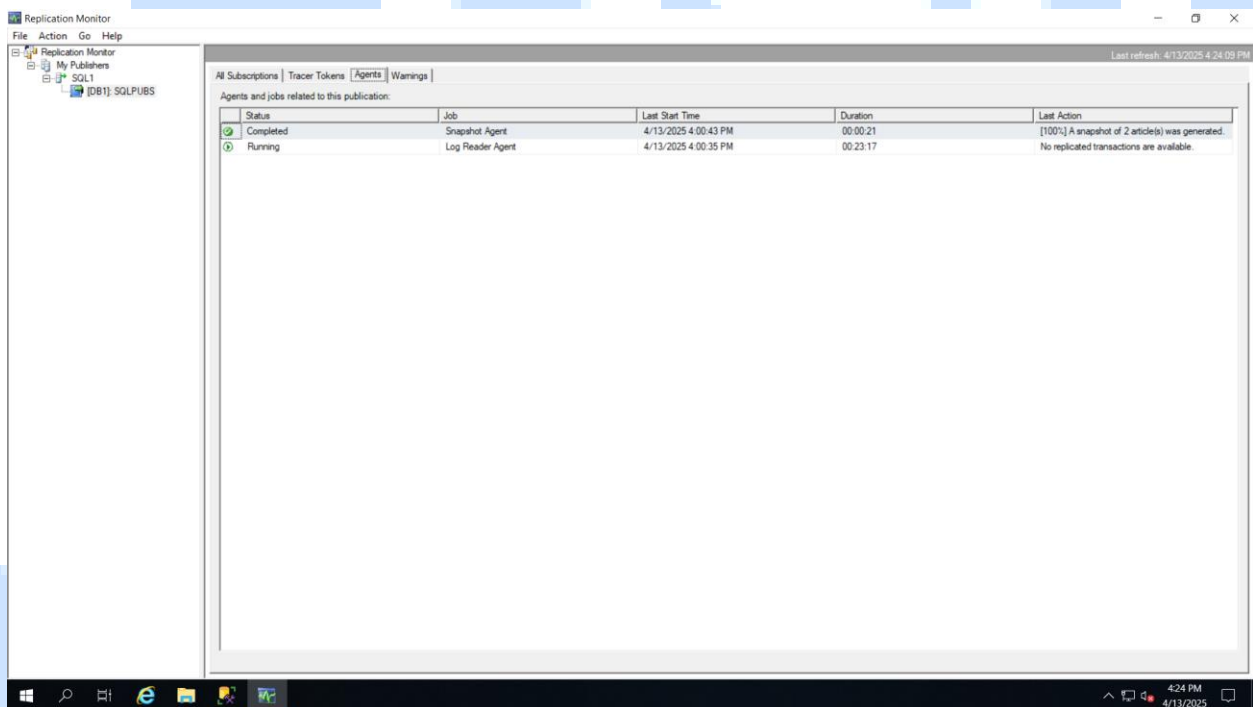
**** Snapshot Agent should complete 100% for Copy all Tables first time to new Database**



Replication Monitor - Publications | Subscription Watch List | Agents |

Status	Publication	Subscriptions	Synchronizing	Current Average Performance	Current Worst Performance
OK	[DB1]: SQLPUBS	1	0		

Last refresh: 4/13/2025 4:23:53 PM



Replication Monitor - All Subscriptions | Tracer Tokens | Agents | Warnings |

Agents and jobs related to this publication:

Status	Job	Last Start Time	Duration	Last Action
Completed	Snapshot Agent	4/13/2025 4:00:43 PM	00:00:21	[100%] A snapshot of 2 article(s) was generated.
Running	Log Reader Agent	4/13/2025 4:00:35 PM	00:23:17	No replicated transactions are available.

Last refresh: 4/13/2025 4:24:09 PM

Logs for Replication

Replication Error

```
SELECT TOP (1000) [id]
,[time]
,[error_type_id]
,[source_type_id]
,[source_name]
,[error_code]
,[error_text]
,[xact_seqno]
,[command_id]
,[session_id]
FROM [distribution].[dbo].[MSrepl_errors] order by time desc;
```

Monitor Replication

```
SELECT publisher,
publisher_db,
publication_id,
CASE publication_type
    WHEN 0 then '0 - Transactional publication'
    WHEN 1 then '1 - Snapshot publication'
    WHEN 2 then '2 - Merge publication'
END AS publication_type_desc,
publication,
CASE agent_type
    WHEN 1 then '1 - Snapshot Agent'
    WHEN 2 then '2 - Log Reader Agent'
    WHEN 3 then '3 - Distribution Agent'
    WHEN 4 then '4 - Merge Agent'
    WHEN 9 then '9 - Queue Reader Agent'
END AS agent_type,
agent_name,
CASE status
    WHEN 1 THEN '1 - Started'
    WHEN 2 THEN '2 - Succeeded'
    WHEN 3 THEN '3 - In progress'
    WHEN 4 THEN '4 - Idle'
    WHEN 5 THEN '5 - Retrying'
    WHEN 6 THEN '6 - Failed'
END AS agent_status,
RIGHT('0' + CAST(cur_latency / 3600 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST((cur_latency / 60) % 60 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST(cur_latency % 60 AS VARCHAR),2) AS cur_latency,
RIGHT('0' + CAST(worst_latency / 3600 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST((worst_latency / 60) % 60 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST(worst_latency % 60 AS VARCHAR),2) AS max_latency,
RIGHT('0' + CAST(best_latency / 3600 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST((best_latency / 60) % 60 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST(best_latency % 60 AS VARCHAR),2) AS min_latency,
RIGHT('0' + CAST(avg_latency / 3600 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST((avg_latency / 60) % 60 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST(avg_latency % 60 AS VARCHAR),2) AS avg_latency,
last_distsync AS last_time_dist_agent_run,
isagentrunningnow AS is_agent_running_now,
agentstoptime AS agent_stop_time,
```

```

CASE warning
    WHEN 1 THEN 'Expiration'
    WHEN 2 THEN 'Latency'
    WHEN 4 THEN 'Merge expiration '
    WHEN 16 THEN 'Merge slow run duration '
    WHEN 32 THEN 'Merge fast run speed '
    WHEN 64 THEN 'Merge slow run speed'
END AS warning,
CASE retention_period_unit
    WHEN 1 THEN CAST(retention AS VARCHAR)+' Week'
    WHEN 2 THEN CAST(retention AS VARCHAR)+' Month'
    WHEN 3 THEN CAST(retention AS VARCHAR)+' Year'
END AS pub_retention_period,
distdb AS distribution_db
FROM distribution.dbo.MSreplication_monitordata
WHERE publisher_db = 'db1'
AND publication IN ('ALL','your_publication_name')
ORDER BY publisher,
agent_type,
publication;

Alerts for Replication

SELECT A.alert_id,
A.error_id,
A.time AS alert_time,
E.time AS error_time,
A.publisher,
A.publisher_db,
CASE A.publication_type
    WHEN 0 THEN 'Snapshot'
    WHEN 1 THEN 'Transactional'
    WHEN 2 THEN 'Merge'
END AS publication_type_desc,
A.publication as publication_name,
A.subscriber,
A.subscriber_db,
A.article,
A.source_object,
A.destination_object,
E.error_text,
A.alert_error_text,
A.agent_id,
CASE A.agent_type
    WHEN 1 THEN 'Snapshot Agent'
    WHEN 2 THEN 'Log Reader Agent'
    WHEN 3 THEN 'Distribution Agent'
    WHEN 4 THEN 'Merge Agent'
    ELSE 'Unknown'
END AS agent_type_desc,
COALESCE(S.name,L.name,D.name,M.name) AS agent_name,
E.session_id AS agent_session_id,
CASE status
    WHEN 0 THEN 'Unserviced'
    WHEN 1 THEN 'serviced'
END AS status_desc
FROM msdb.dbo.sysreplicationalerts AS A
LEFT JOIN distribution.dbo.MSrepl_errors AS E ON A.error_id = E.id

```

```

LEFT JOIN distribution.dbo.MSsnapshot_agents AS S ON S.id = A.agent_id
LEFT JOIN distribution.dbo.MSlogreader_agents AS L ON L.id = A.agent_id
LEFT JOIN distribution.dbo.MSdistribution_agents AS D ON D.id = A.agent_id
LEFT JOIN distribution.dbo.MSmerge_agents AS M ON M.id = A.agent_id
WHERE A.time >= GETUTCDATE()-1
AND A.publisher_db = 'db1'
AND A.publication = 'sqlpubs'
AND A.subscriber_db = 'db1sub'
ORDER BY A.alert_id DESC;

```

Any Pending Commands for Replication

```

SELECT
p.name as publisher_server,
da.publisher_db,
s.name as subscriber_server,
da.subscriber_db,
da.publication as publication,
a.destination_object as table_name,
ds.DelivCmdsInDistDB as delivered_commands_in_distribution_db,
ds.UndelivCmdsInDistDB as undelivered_commands_in_distribution_db
FROM distribution.dbo.MSdistribution_status ds
INNER JOIN distribution.dbo.MSdistribution_agents da ON da.id = ds.agent_id
INNER JOIN distribution.dbo.MSArticles a ON a.publisher_id = da.publisher_id AND a.publisher_db = da.publisher_db AND
a.article_id = ds.article_id
INNER JOIN master.sys.servers s ON s.server_id = da.subscriber_id
INNER JOIN master.sys.servers p ON p.server_id = da.publisher_id
ORDER BY undelivered_commands_in_distribution_db DESC,
table_name ASC

```

Agents History

```

SELECT H.time AS message_log_time,
H.agent_id,
A.name AS agent_name,
H.runstatus,
CASE runstatus
    WHEN 1 THEN 'Start'
    WHEN 2 THEN 'Succeed'
    WHEN 3 THEN 'In progress'
    WHEN 4 THEN 'Idle'
    WHEN 5 THEN 'Retry'
    WHEN 6 THEN 'Fail'
END AS runstatus_desc,
H.start_time AS job_exec_start_time,
H.duration AS session_duration_in_sec,
H.comments AS message_text,
H.current_delivery_rate AS current_delivery_rate_per_sec,
H.current_delivery_latency/1000 AS current_delivery_latency_in_sec,
H.delivery_rate AS delivery_rate_per_sec,
H.delivery_latency/1000 AS delivery_latency_in_sec,
H.delivered_transactions AS transactions_delivered_in_session,
H.delivered_commands AS commands_delivered_in_session,
H.average_commands AS average_commands_in_session,
H.total_delivered_commands, --total commands delivered since subscription was created
A.publisher_db,
A.publication,
A.subscriber_db,

```

```
H.error_id,  
E.time AS error_time,  
E.error_code,  
E.error_text  
FROM distribution.dbo.MSdistribution_history AS H  
INNER JOIN distribution.dbo.MSdistribution_agents AS A ON H.agent_id = A.id  
LEFT JOIN distribution.dbo.MSrepl_errors AS E ON E.id = H.error_id  
WHERE H.time >= GETUTCDATE()-2  
ORDER BY message_log_time desc;
```

```
SELECT H.time AS message_log_time,  
H.agent_id,  
A.name AS agent_name,  
H.runstatus,  
CASE runstatus  
    WHEN 1 THEN 'Start'  
    WHEN 2 THEN 'Succeed'  
    WHEN 3 THEN 'In progress'  
    WHEN 4 THEN 'Idle'  
    WHEN 5 THEN 'Retry'  
    WHEN 6 THEN 'Fail'  
END AS runstatus_desc,  
H.start_time AS job_exec_start_time,  
H.duration AS session_duration_in_sec,  
H.comments AS message_text,  
H.delivered_transactions AS transactions_delivered_in_session,  
H.delivered_commands AS delivered_commands_per_sec,  
H.delivery_rate AS delivery_rate_per_sec,  
A.publisher_db,  
A.publication,  
H.error_id,  
E.time AS error_time,  
E.error_code,  
E.error_text  
FROM distribution.dbo.MSlogreader_history AS H  
INNER JOIN distribution.dbo.MSlogreader_agents AS A ON H.agent_id = A.id  
LEFT JOIN distribution.dbo.MSrepl_errors AS E ON E.id = H.error_id  
WHERE H.time >= GETUTCDATE()-7  
ORDER BY message_log_time desc
```