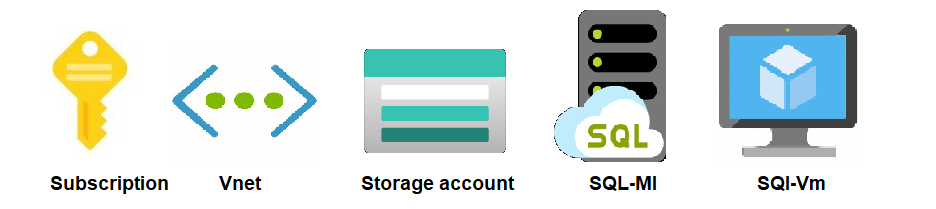
# Database migration from Managed Instance to SQL Server with transactional replication

## **Transactional replication setup**

We will set up a transactional replication for migrating the database from Managed Instance to SQL Server 2017 hosted on Azure VM.

* Managed Instance will be both Publisher and Distributor,
* SQL Server will be Subscriber.

1



# Prerequisites:

# Need Azure Subscription.

# Storage account with one file share in it (Create using azure portal).

# Need a Vent Peering (you need this only if your SQL-MI & SQL-VM are in different network)

# One Azure Managed Instance (Create using azure portal).

# One SQL VM (Create using azure portal).

# 2

# 

## **Requirements**

**VNET PEERING**:

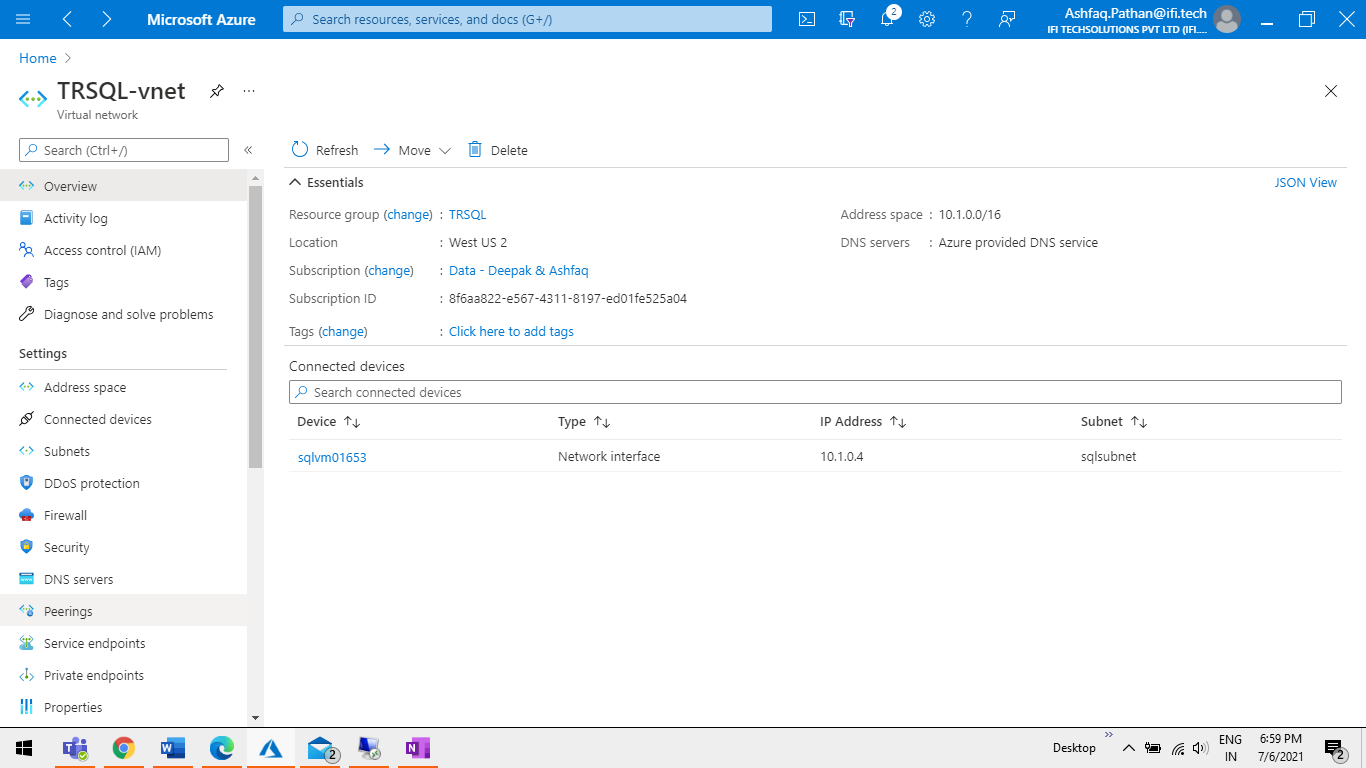
For replication to work, there must be a network connectivity between SQl-MI and SQL Server. ensured by having both servers on the same VNET. If that were not the case then VNET peering setup would be required.

In my case I do not have my SQL-MI & SQL-VM on same network I have done a VNET Peering to get them on same network below are steps with Screenshots.

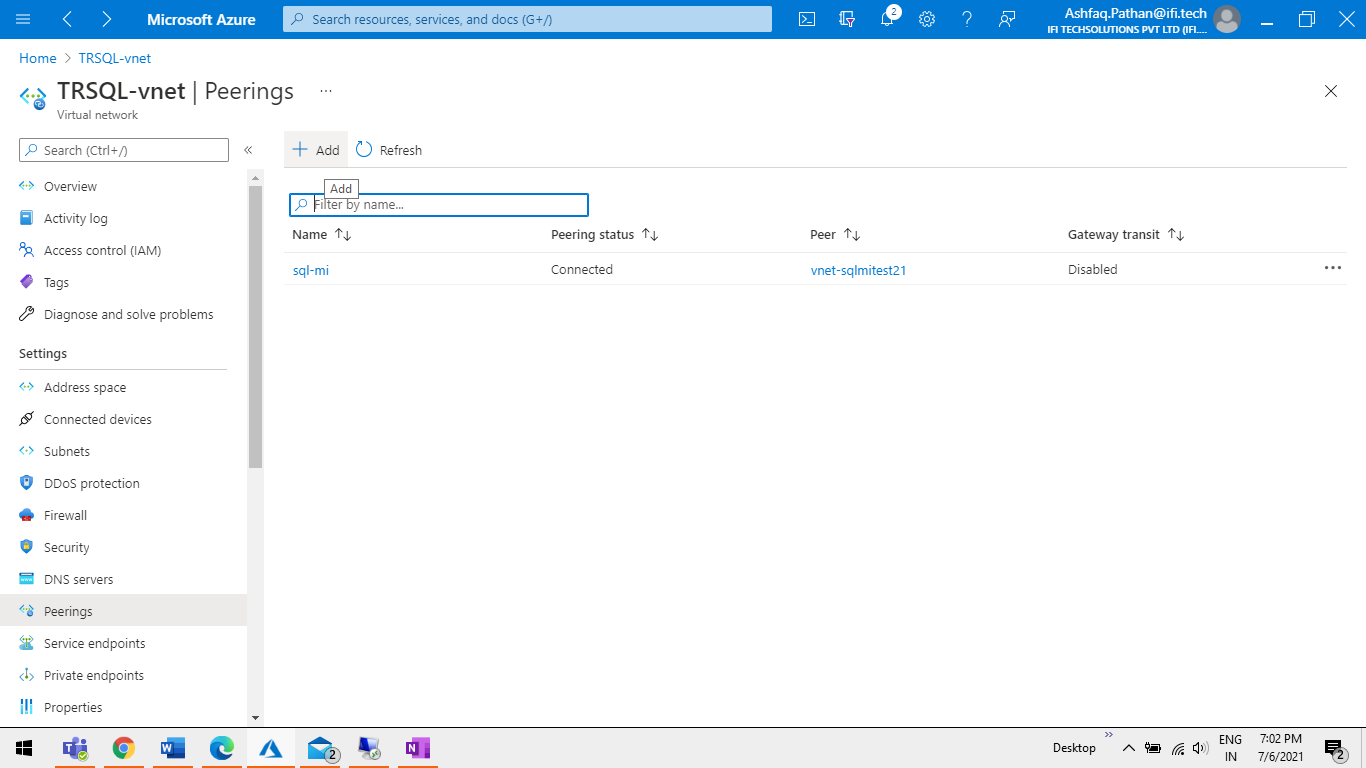
1: VNET Peering. Form sql-vm to sql-mi.

Step1 Open Vnet of Sql-vm you can get this from Resource group where you have your

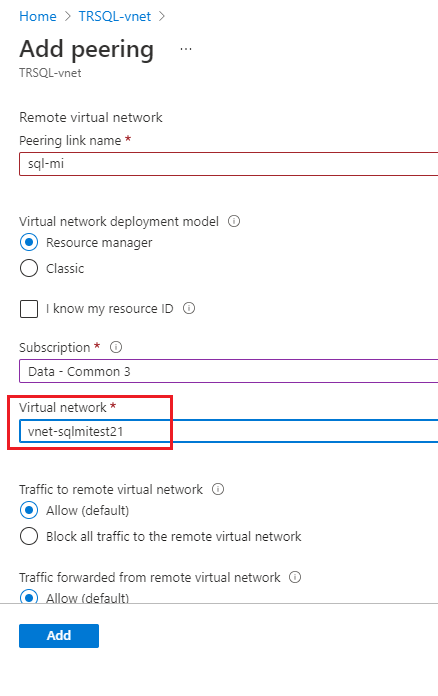
Sql-vm Click on Peering’s option in left bar.



Step 2 Click on add button.



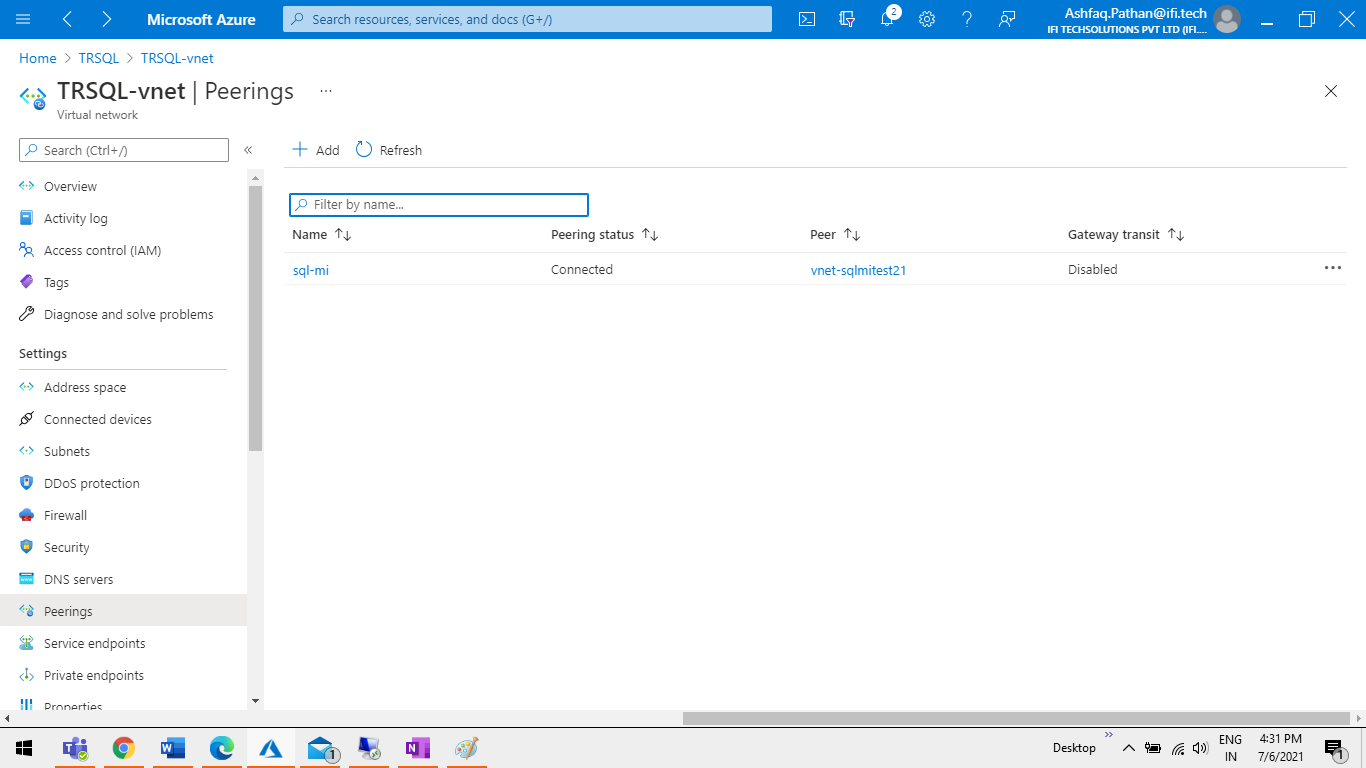
Step 3 Add name and select subscription and SQL-MI VNET from dropdown.



Step 4 Check Peering status

TRSQL-vent = I have sql-vm in this vnet.

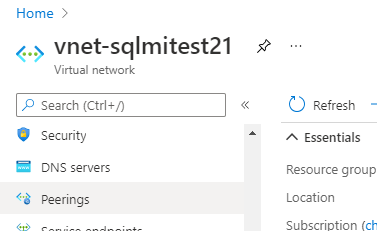
Vnet-sqlmitest21 = I have sql-mi in this vnet.

`

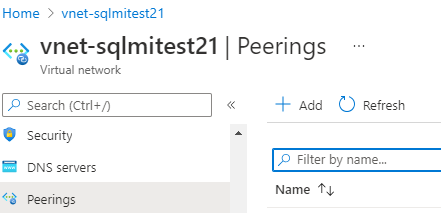
2: VNET Peering .Form sql-mi to sql-vm.

Step1 Open Vnet of Sql-mi you can get this from Resource group where you have your

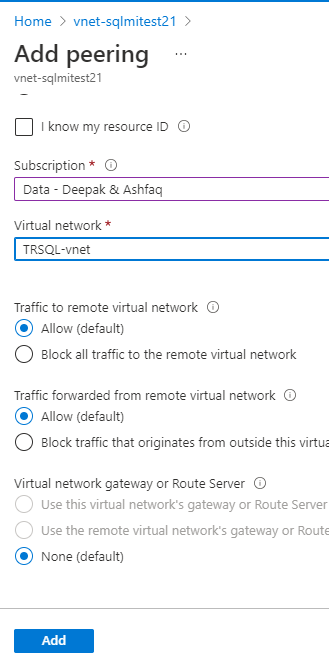
Sql-mi Click on Peering’s option in left bar.



Step 2 Click on add button.



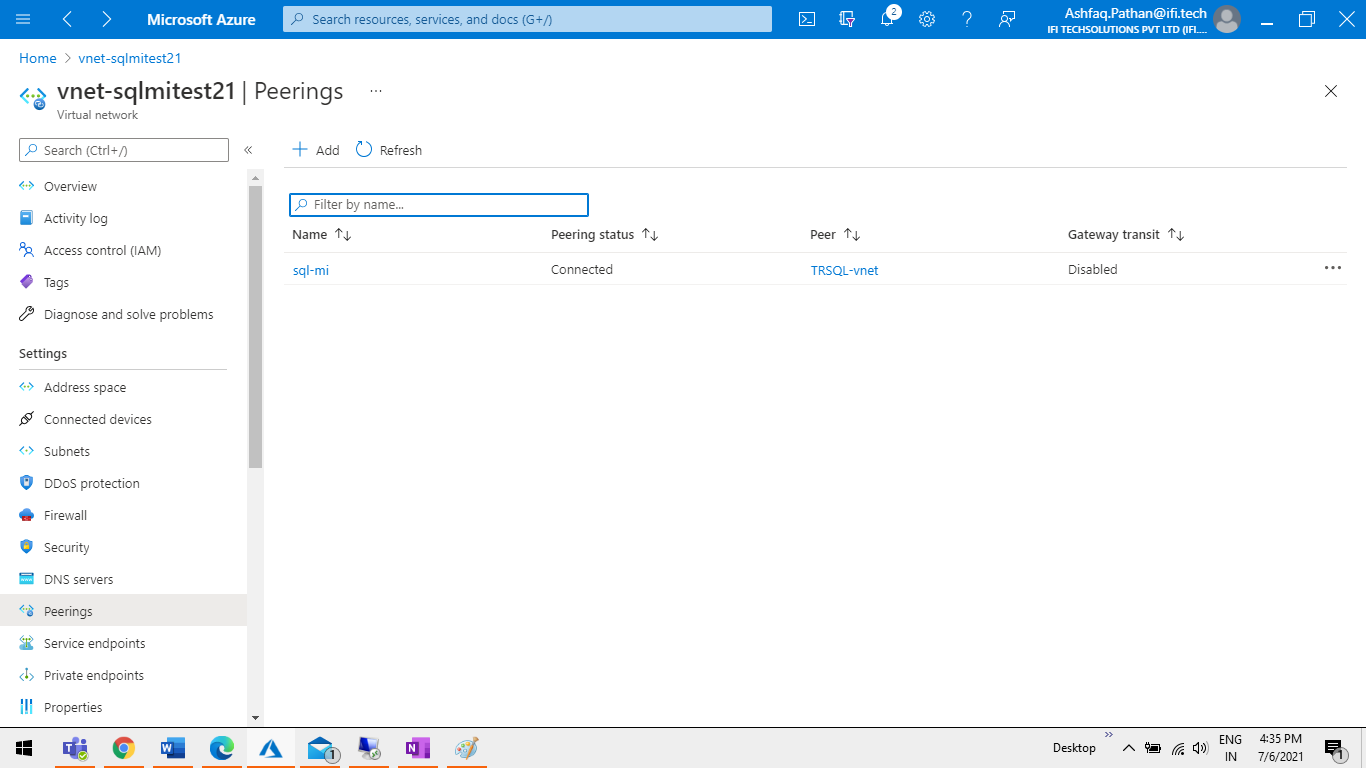
Step 3 Add name and select subscription and SQL-VM VNET from dropdown.



Step 4 Check Peering status

Vnet-sqlmitest21 = I have sql-mi in this vnet.

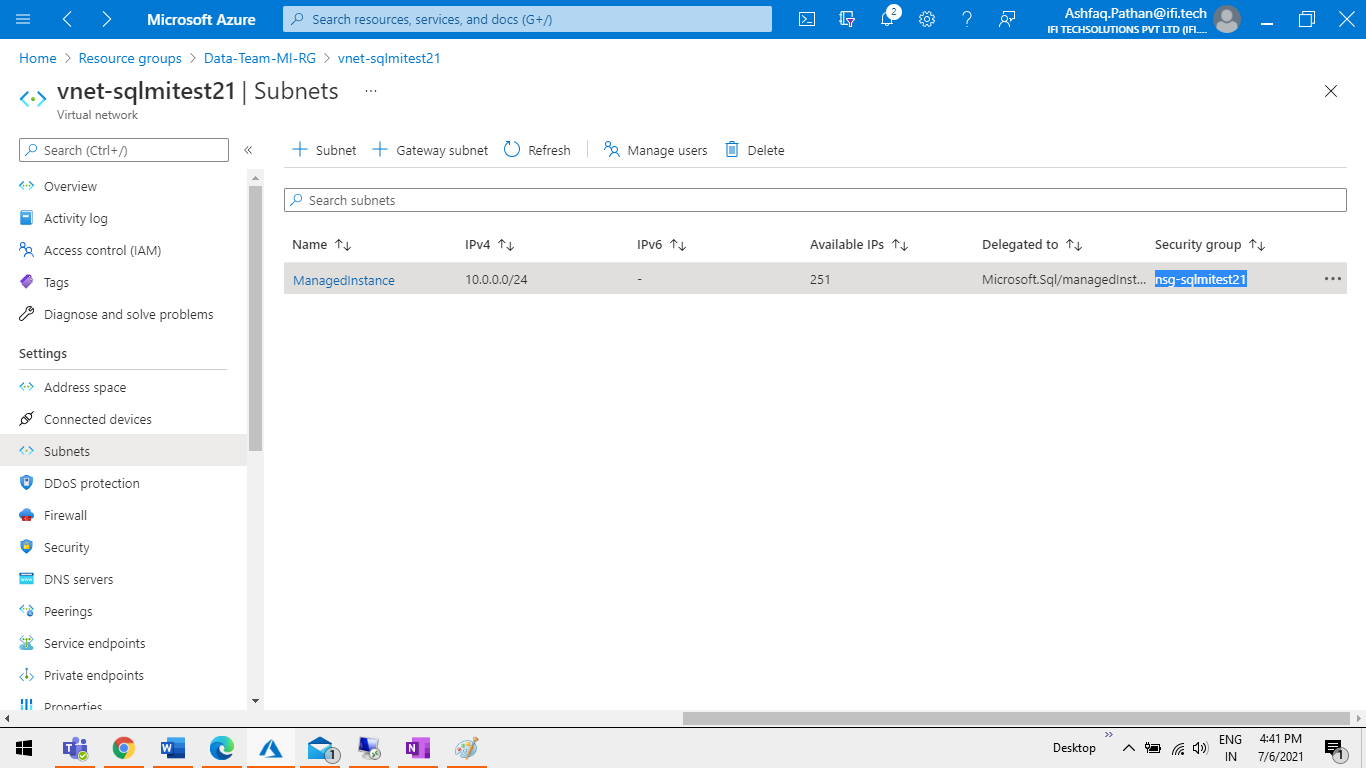
TRSQL-vent = I have sql-vm in this vnet.



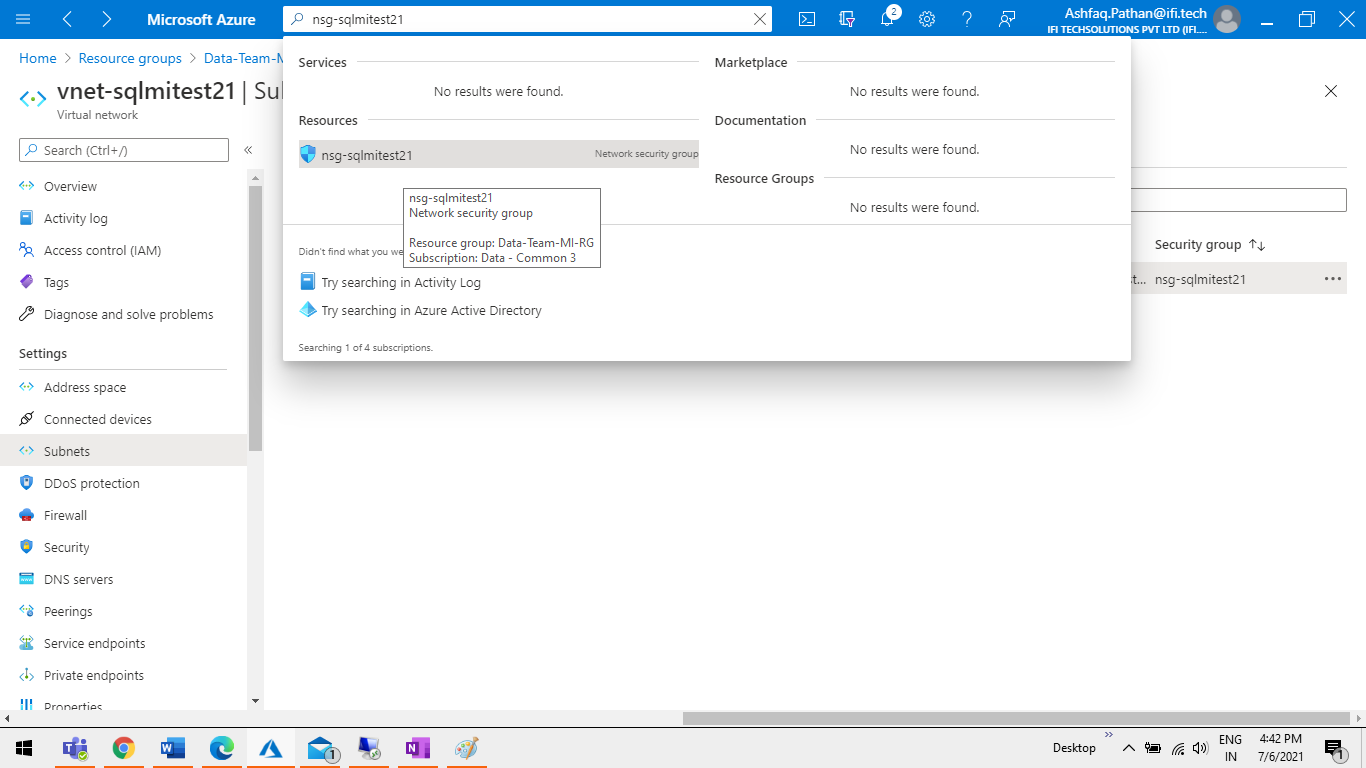
**ADD Outbound Port Rule**:

Outbound TCP port 445 needs to be open for transactional replication, so in case you have NSG make sure this port is allowed. Allowing this port will enable Managed Instance to access Azure Storage account, which will be covered a bit later. To allow the port, go to the virtual network of your Managed Instance and identify the security group name.

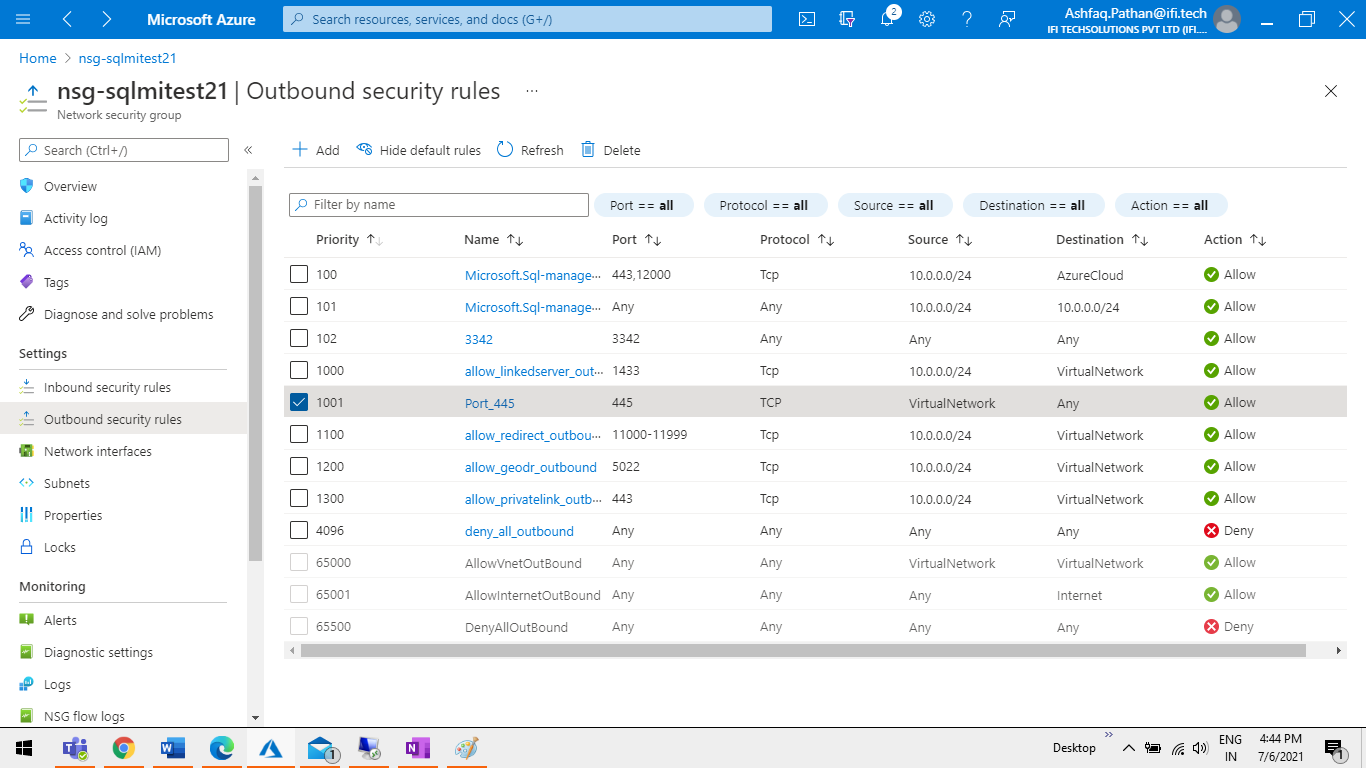
Step1: get the Security group name.



Step2 Open NSG search that in portal and open it.



Step3 Add Port 455 in outbound security riles.

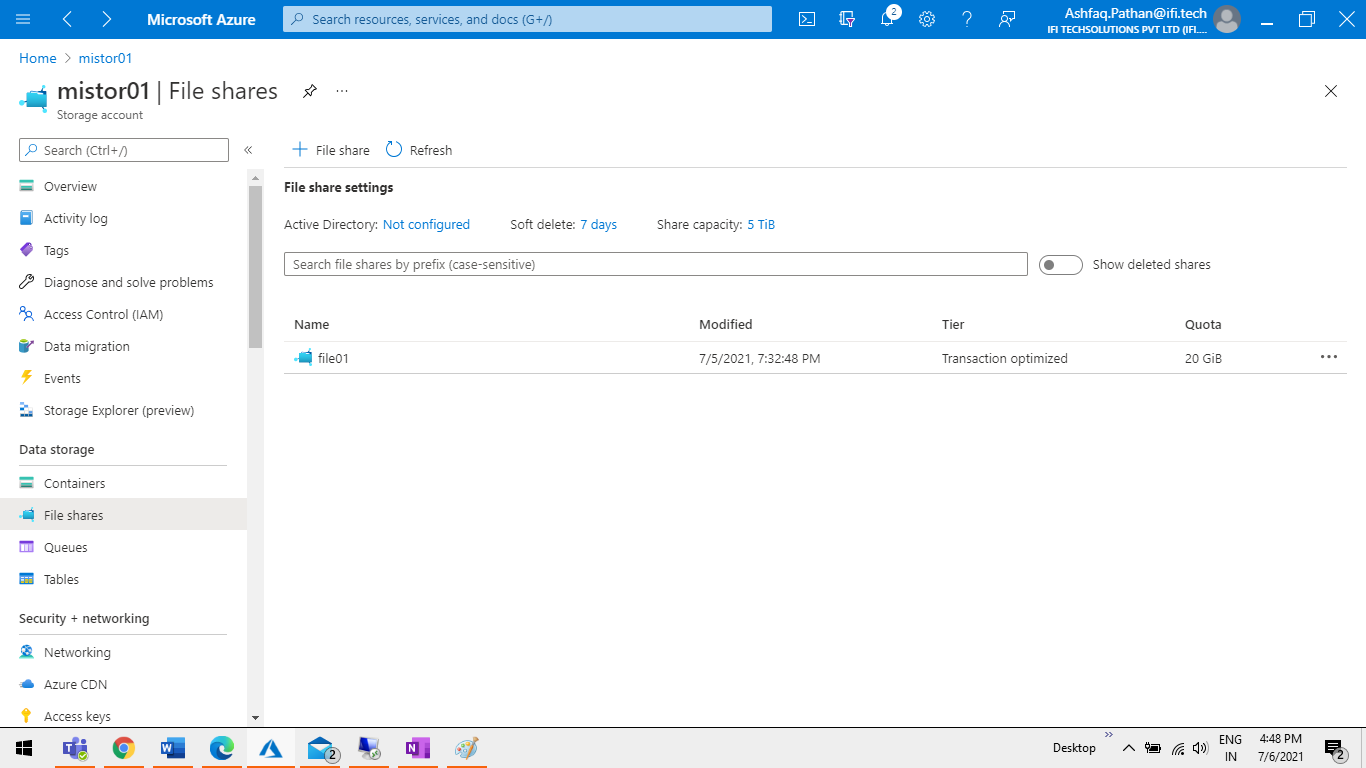


**Storage account:**

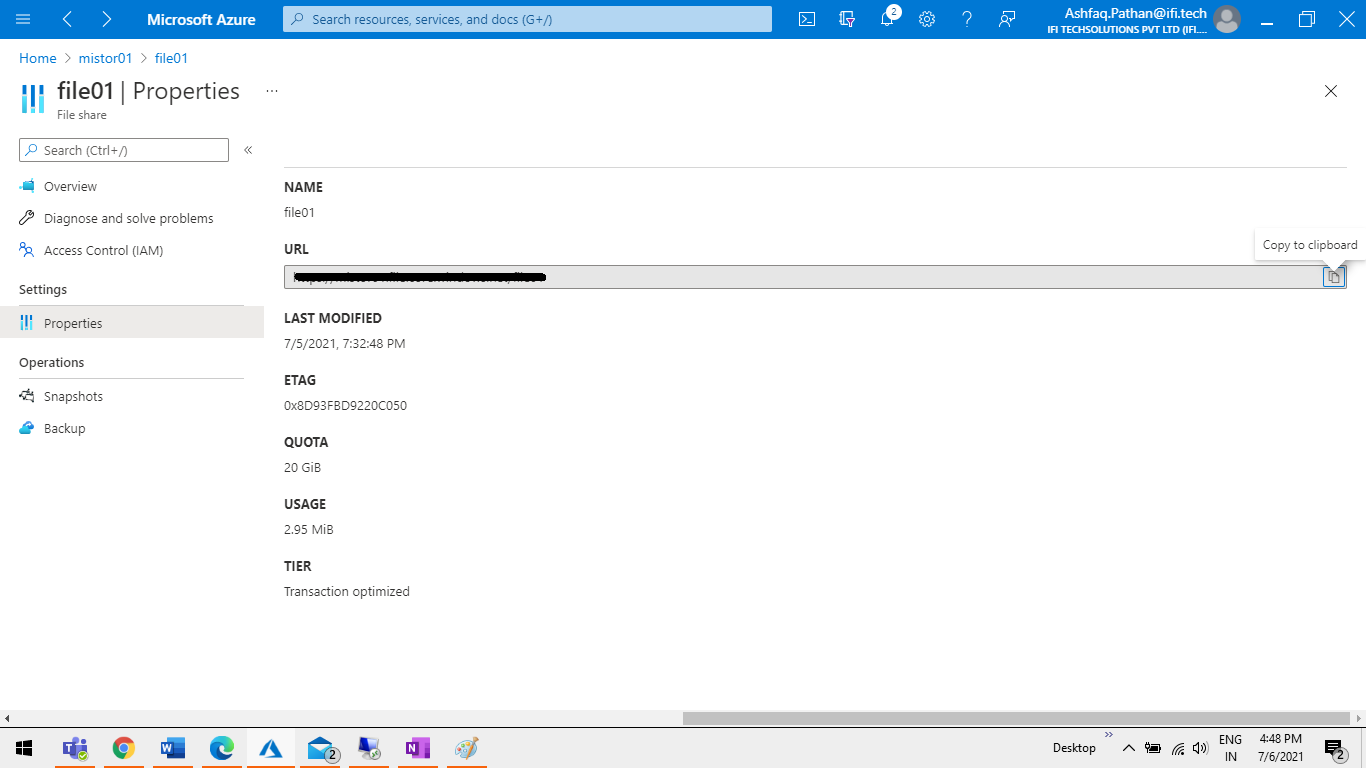
Transactional replication will require a storage account, so if you don’t have one yet, use the Azure Portal to create it. Alternatively, you can use one of your existing storage accounts. Within the Storage account, create a File Share with quota of 20 GiB.

Go to the File share Properties and copy the storage **URL** because this value will be needed in the following T-SQL script.

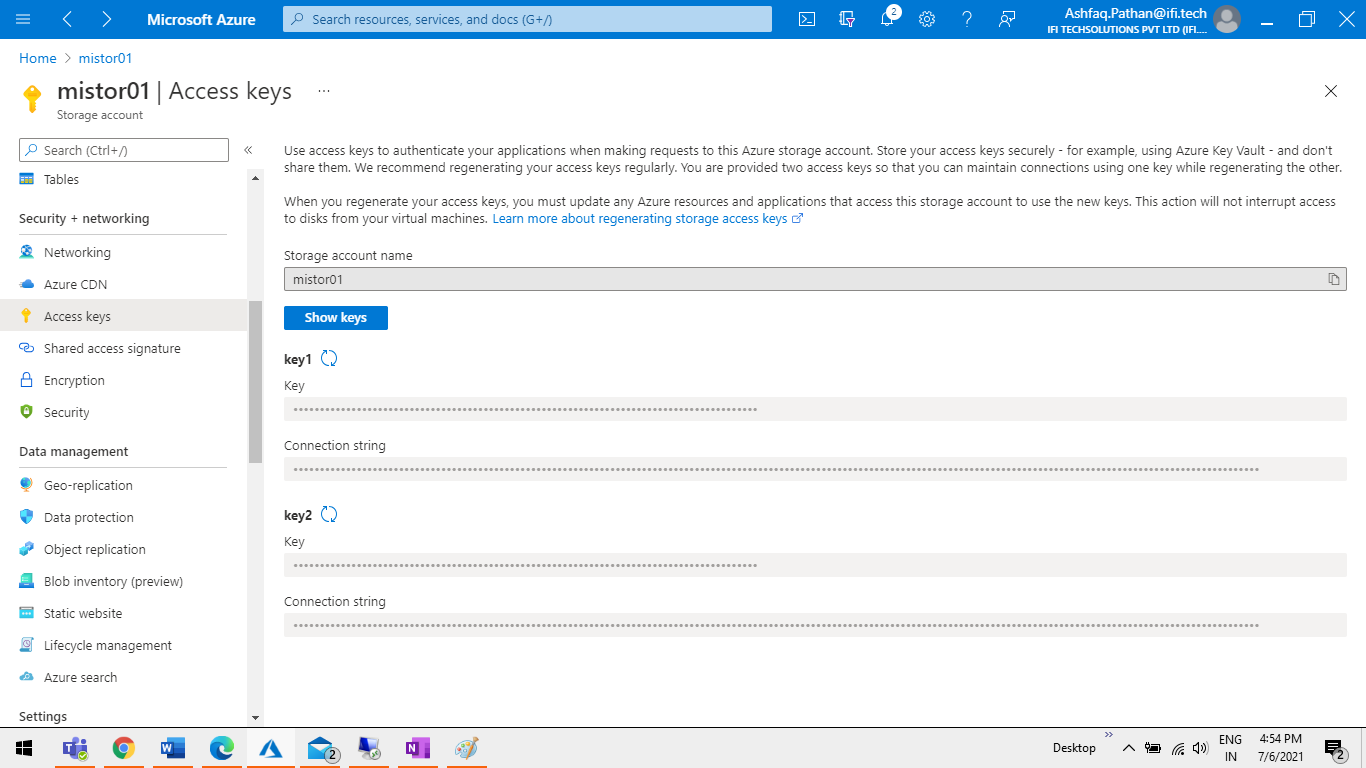
Step1 Open Storage account add new File share and open it.



Step2 Copy URL from File share properties. Ex: https://edfffsdf/asdsda/as

****

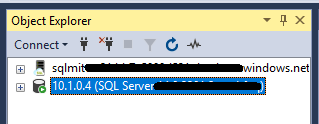
Step 3 Copy key for that go to Storage account > Access key > Connection string.



**Important: From Storage account we need 2 things URL Path & Connection String**

# Configure replication.

Step1: Start VM an open SSMS and connect SQL-MI with Private endpoint & SQL-VM with Private IP.



## Step2 Configure distribution

Connect to your sql-mi managed instance using SQL Server Management Studio and run the following T-SQL code to configure your distribution database.

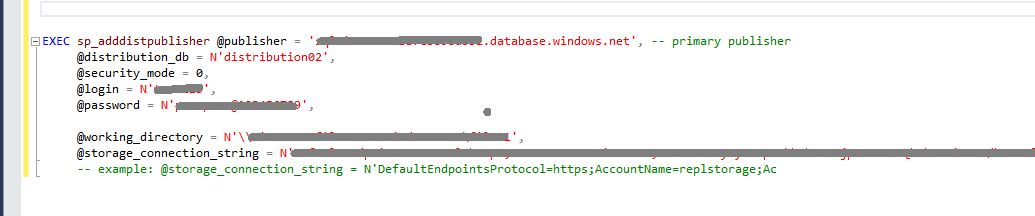
|  |
| --- |
| USE [master]  GO    EXEC sp\_adddistributor @distributor =’Sql-mi.database.windows.net', @password = 'xyz123456789'  EXEC sp\_adddistributiondb @database = N'distribution02'  GO |



## Step3 Configure publisher to use distributor.

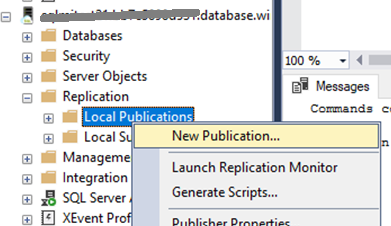
On your publisher SQL Managed Instance sql-mi, run the following code to register the new distributor with your publisher.

|  |
| --- |
| EXEC sp\_adddistpublisher @publisher = 'sql-mi.database.windows.net', -- primary publisher  @distribution\_db = N'distribution02',  @security\_mode = 0,  @login = N'user123',  @password = N'pasword@123456789',    @working\_directory = N'\\storage-account-name.file.core.windows.net\file\_share\_name',  @storage\_connection\_string = N'Connection-string’  -- example: @storage\_connection\_string =N'DefaultEndpointsProtocol=https;AccountName=replstorage;Ac |

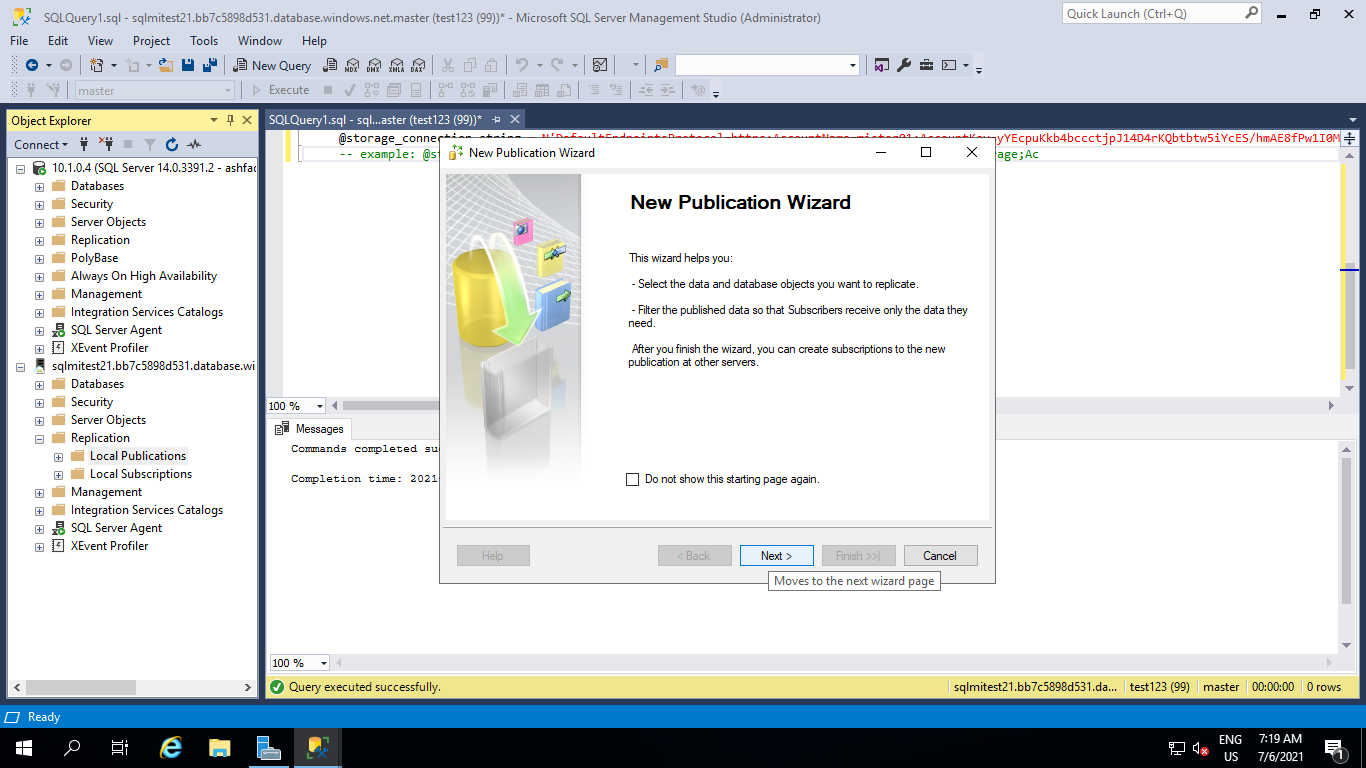


## Step4 Create publication.

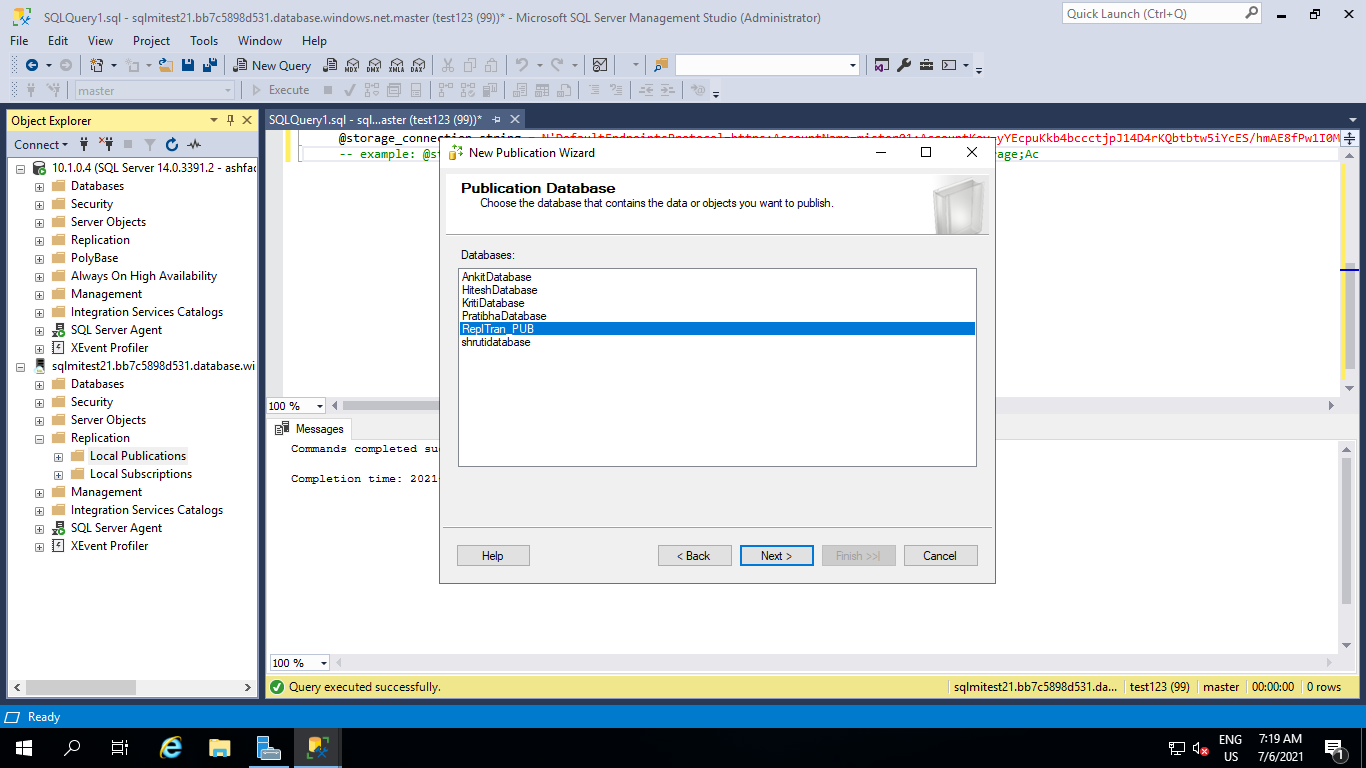
Step 4.1 SQL-MI > Replication > Local Publications > Right Click Create New Publication.



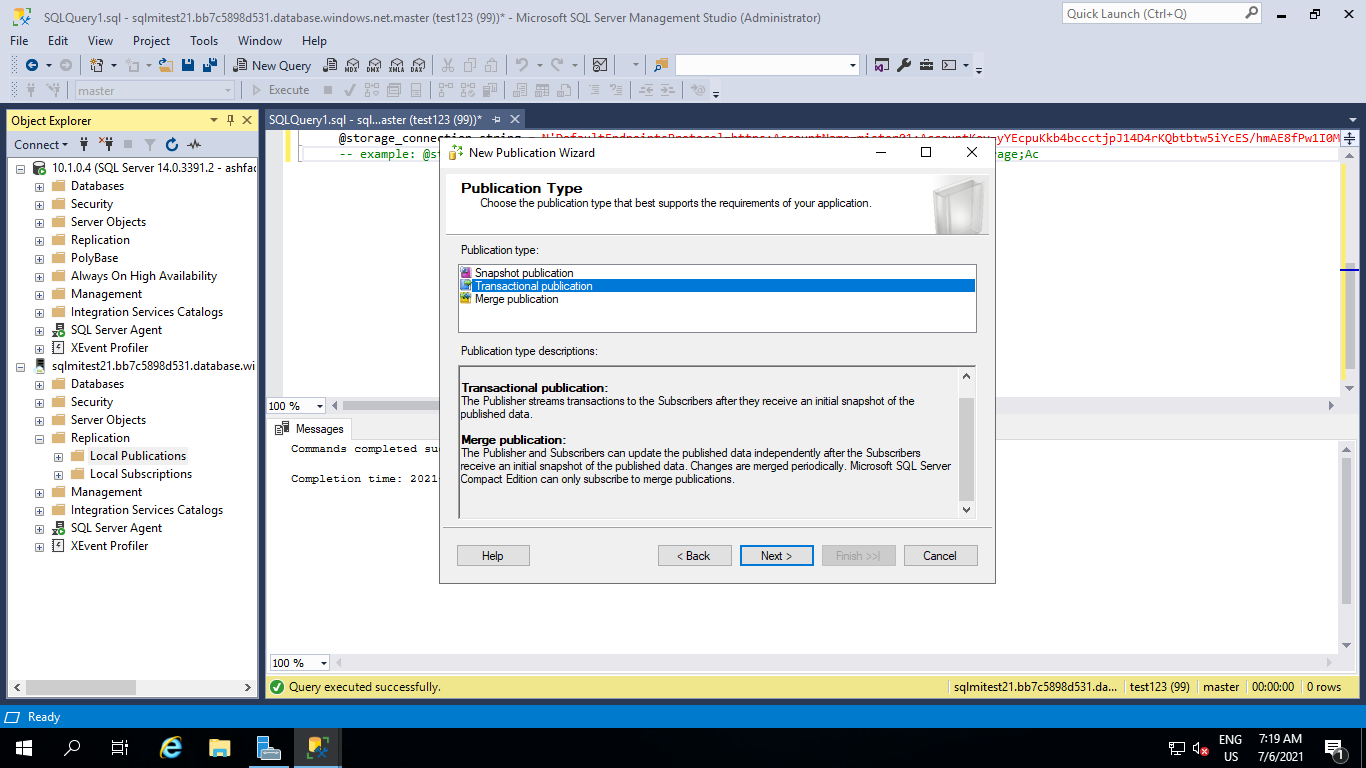
Step 4.2 Click on Next Button.



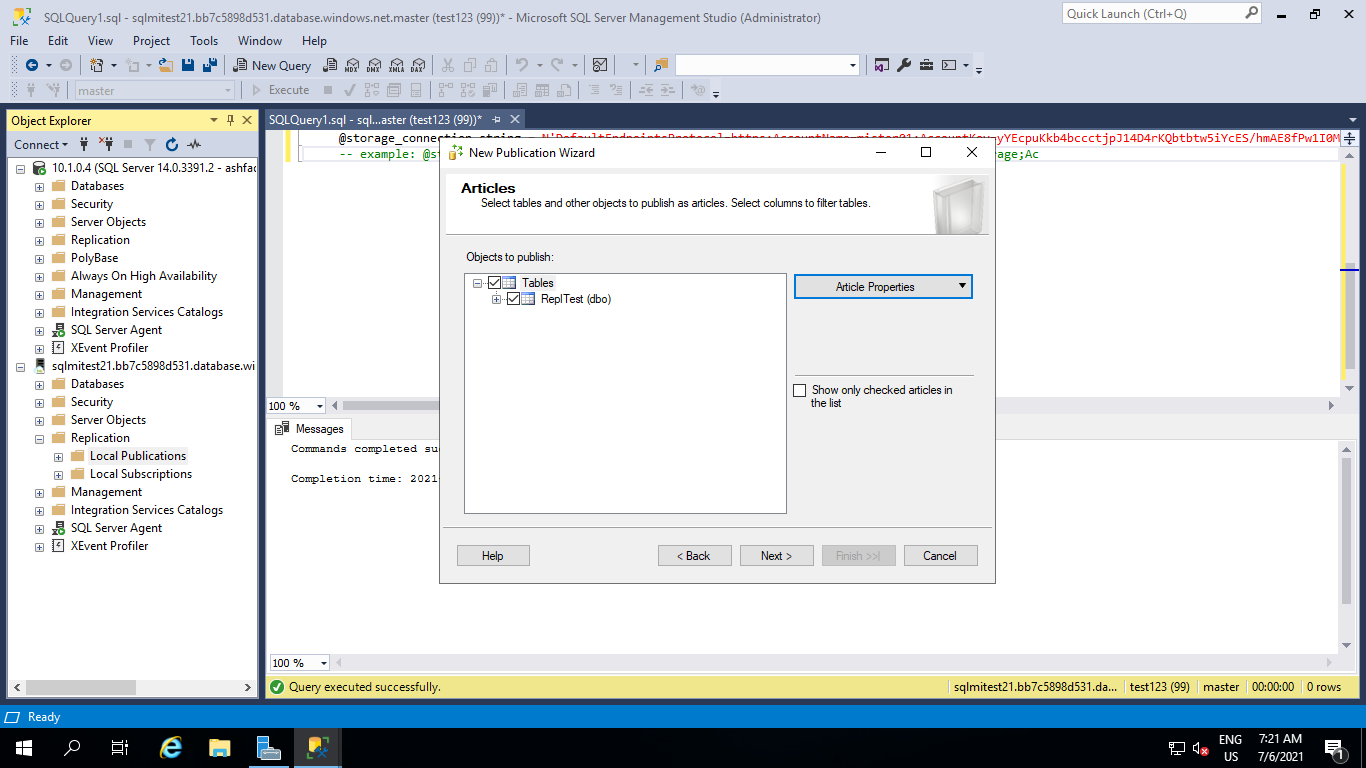
Step 4.3 Select the database you want to publish and again click Next.



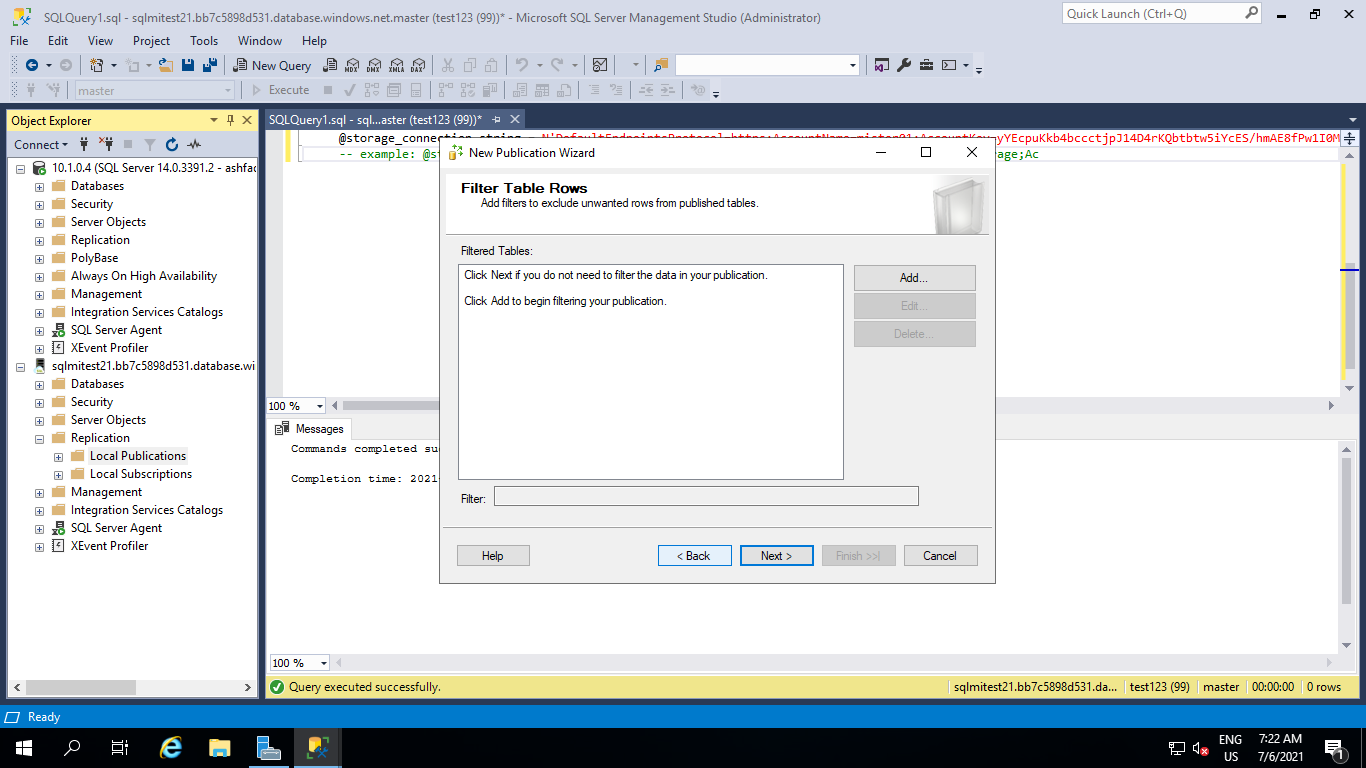
Step 4.4 As on the screenshot below, for Publication Type chose Transactional publication and click Next.



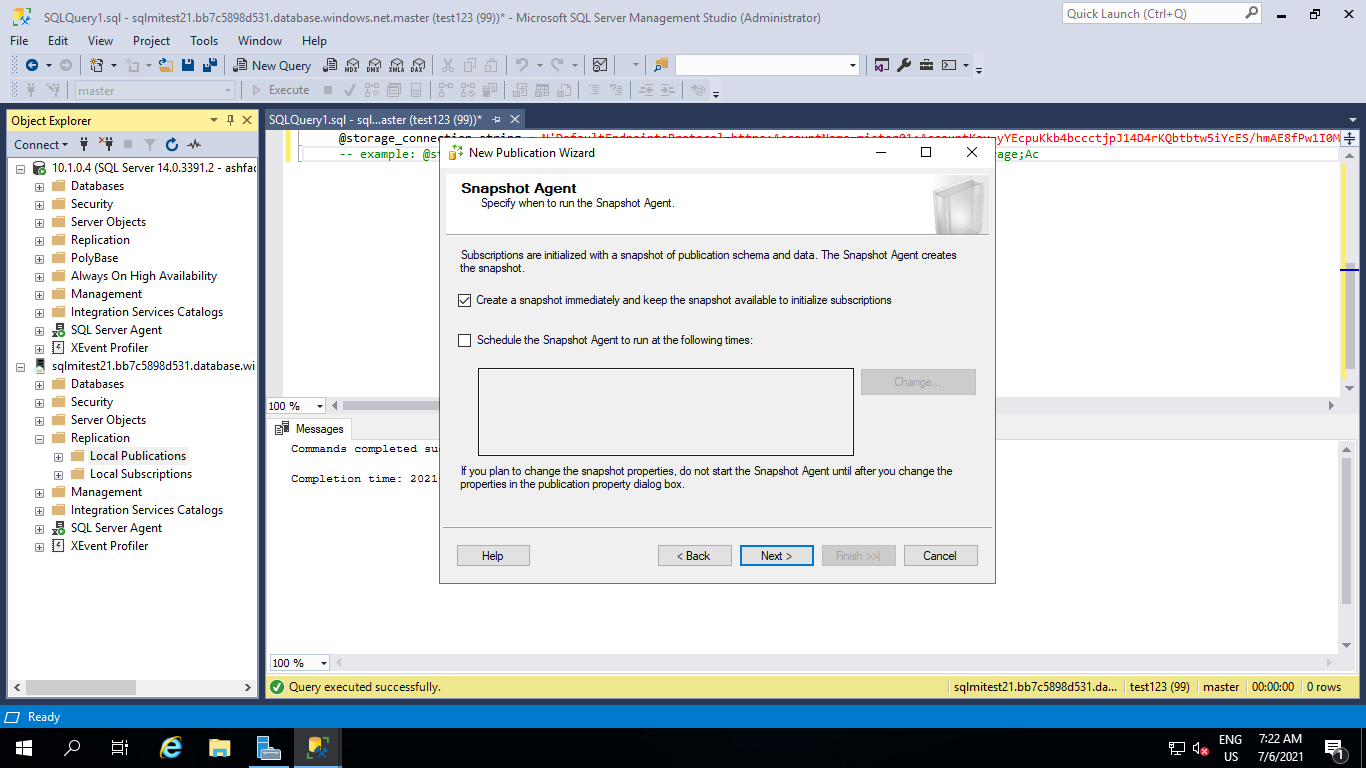
Step 4.5 Then on the Articles window, choose objects you want to replicate. In this example we will chose all Tables and click Next button.



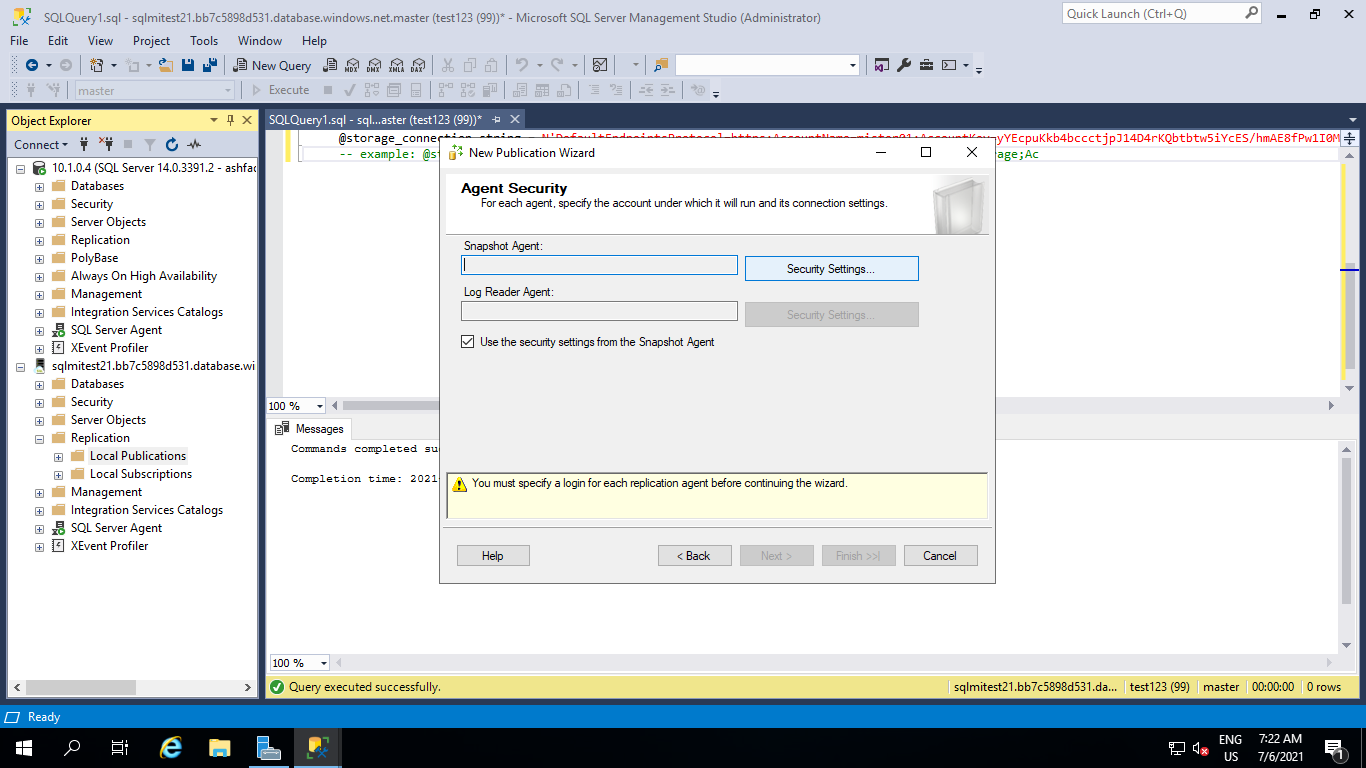
Step 4.6 On the **Filter Table Rows** window we don’t add any filters, just click Next.



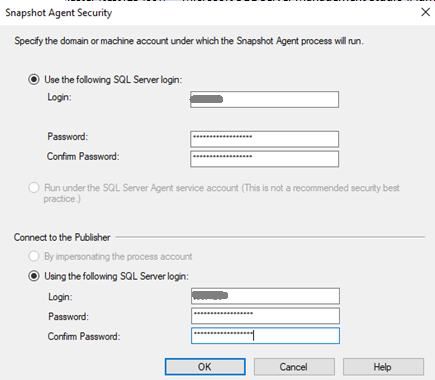
Step 4.7 On **Snapshot Agent** windows check “Create a snapshot immediately…” and click Next.



Step 4.8 On **Agent Security** windows click on Security Settings to provide credentials for both Snapshot Agent and connection to the Publisher.

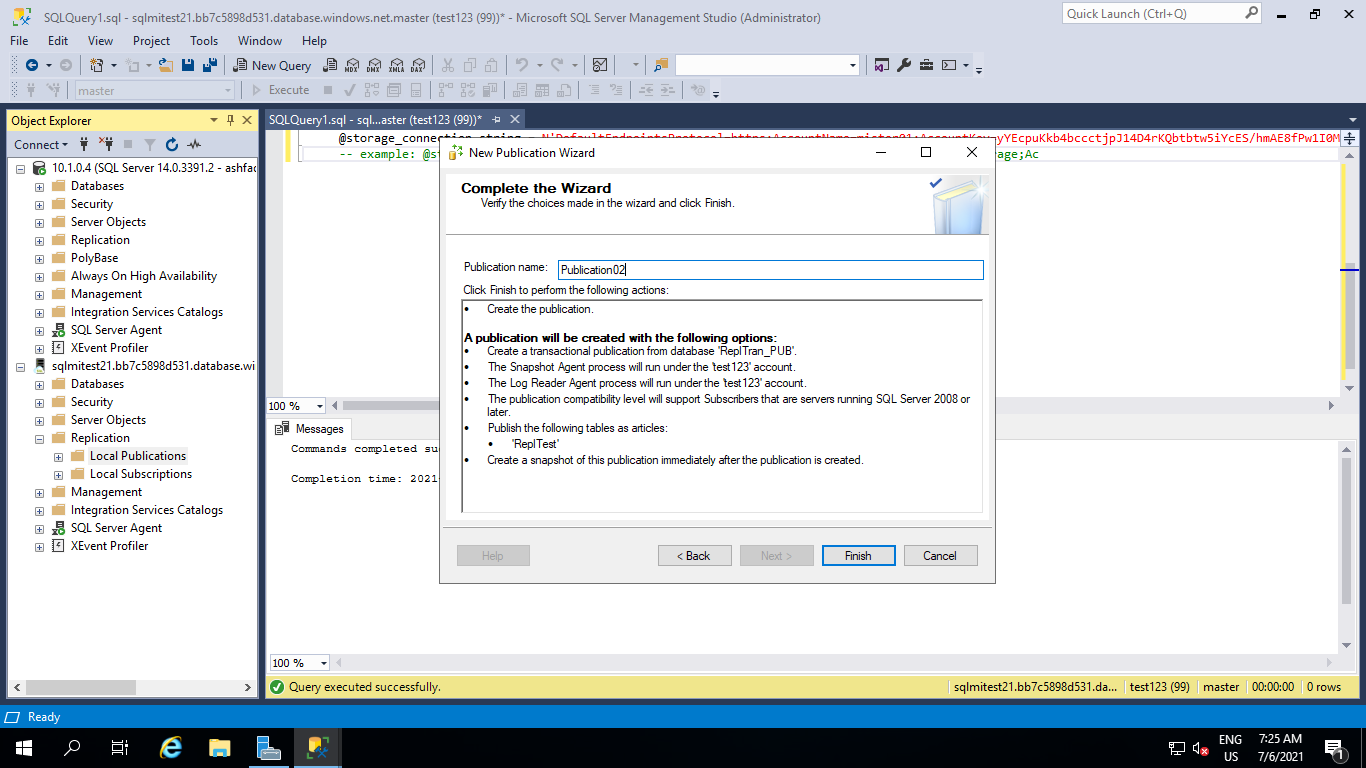


Step 4.9 add user & password of SQL-MI and click on ok.

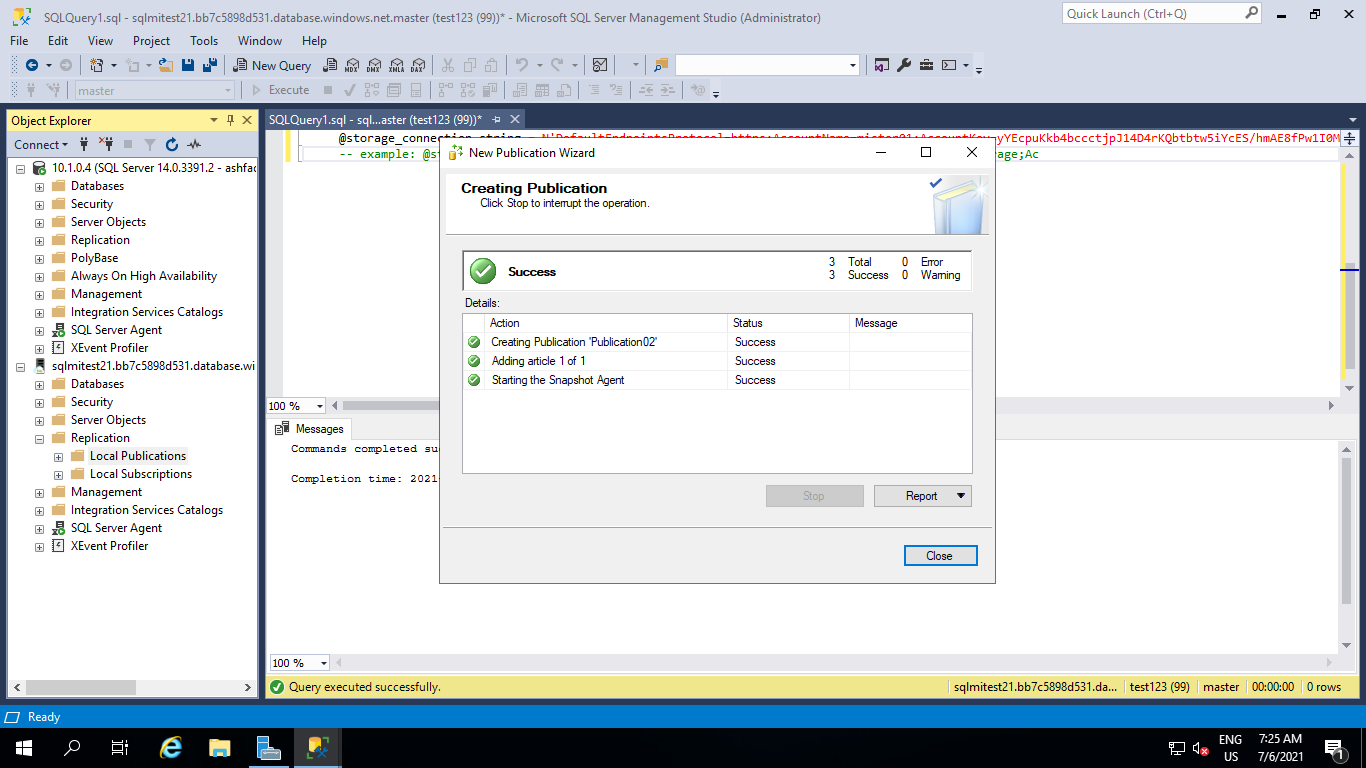


Step 4.10 On the **Wizard Actions** window check **Create the Publication**and click Next.

Finally in the **Complete the Wizard**, provide the name for the publication and click Finish.



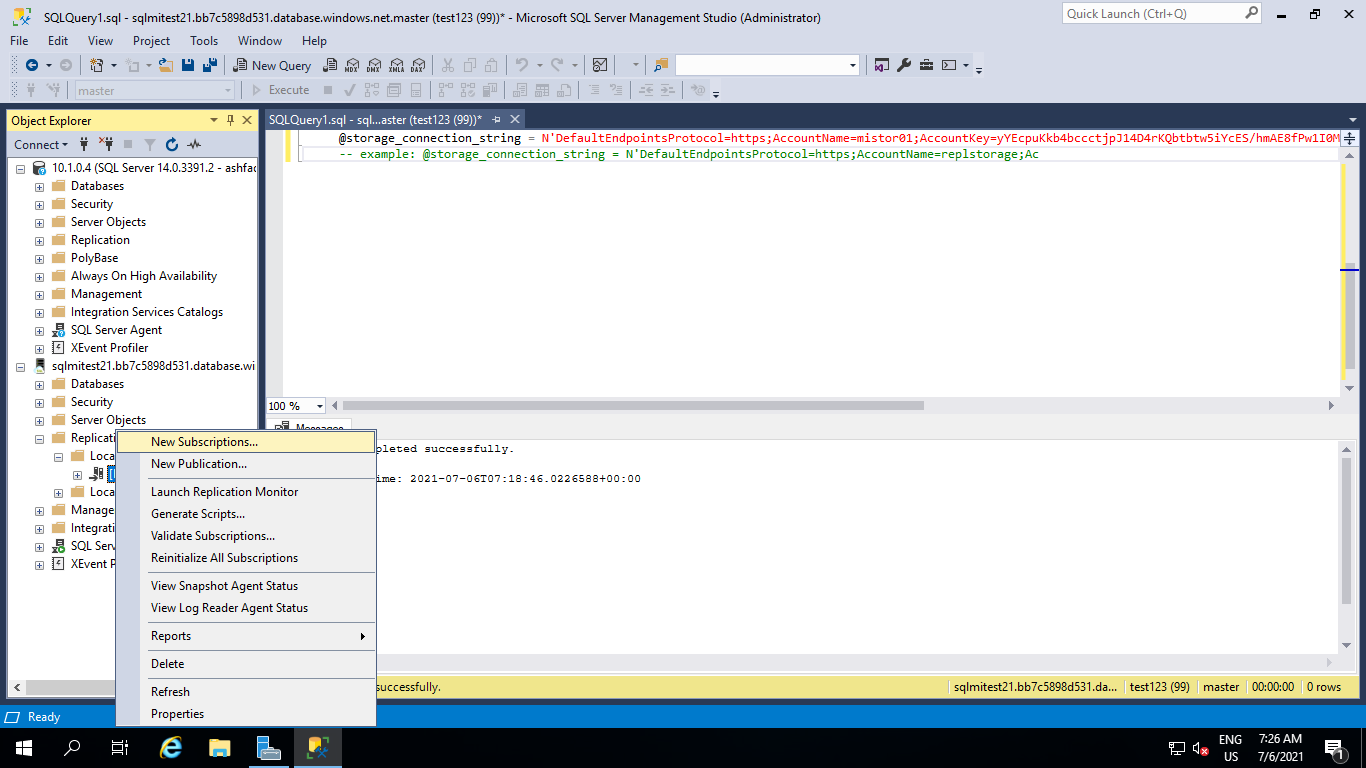
Step 4.11 This Wizard will show progress and once the process is done you should see a Success sign with 0 Errors and 0 Warnings. Click the Close button. With this the publication is successfully created.



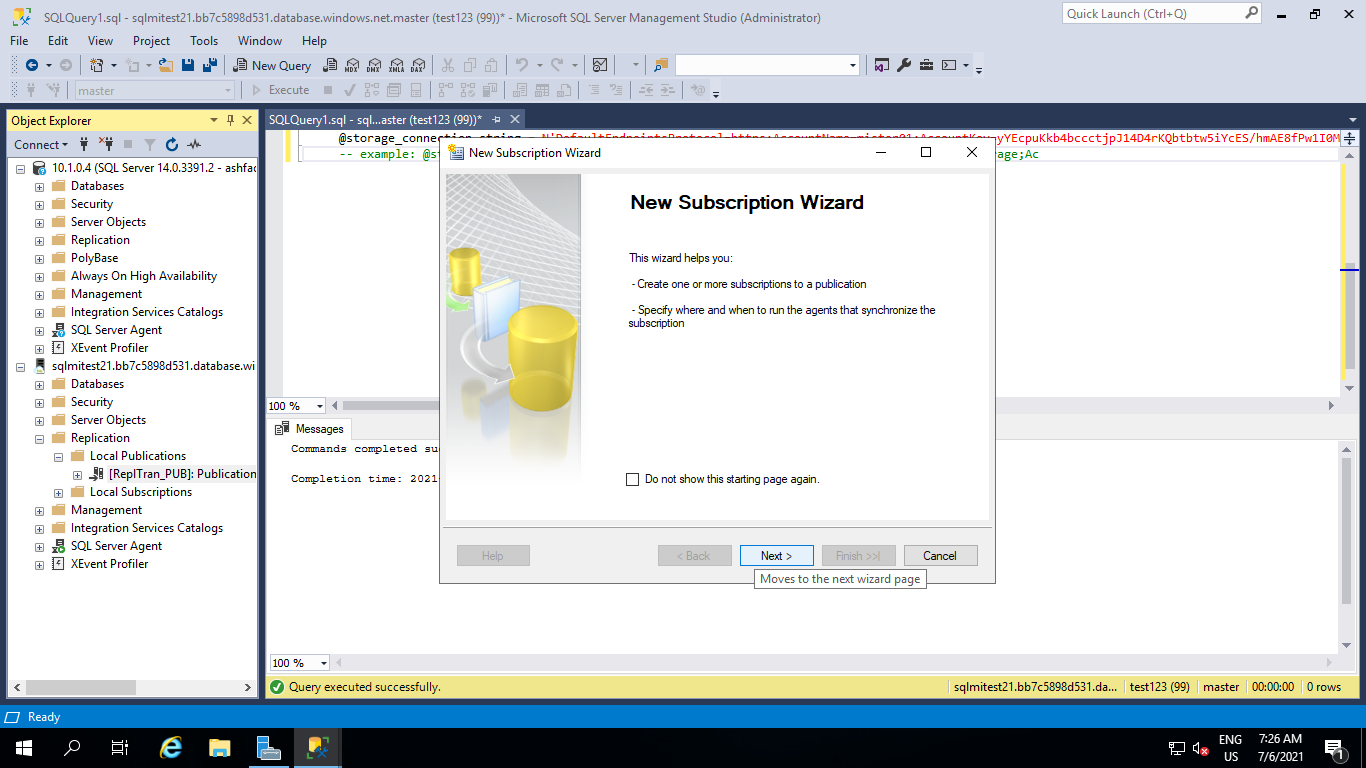
To check the status of the publication, refresh the Local Publications in SSMS and you will find newly created publication. From the context menu of the newly created Publication, click Launch Replication Monitor and find the publisher that has been just created. On the Agents tab you can monitor the progress of the Snapshot Agent and Log Reader Agent. Time needed for a snapshot to be created varies and depends on the size of data that’s participating in the replication. Once Snapshot Agent’s Last Action is “[100%] A snapshot of x article(s) was generated.” the publication is ready.

### Step5 Configure replication subscription.

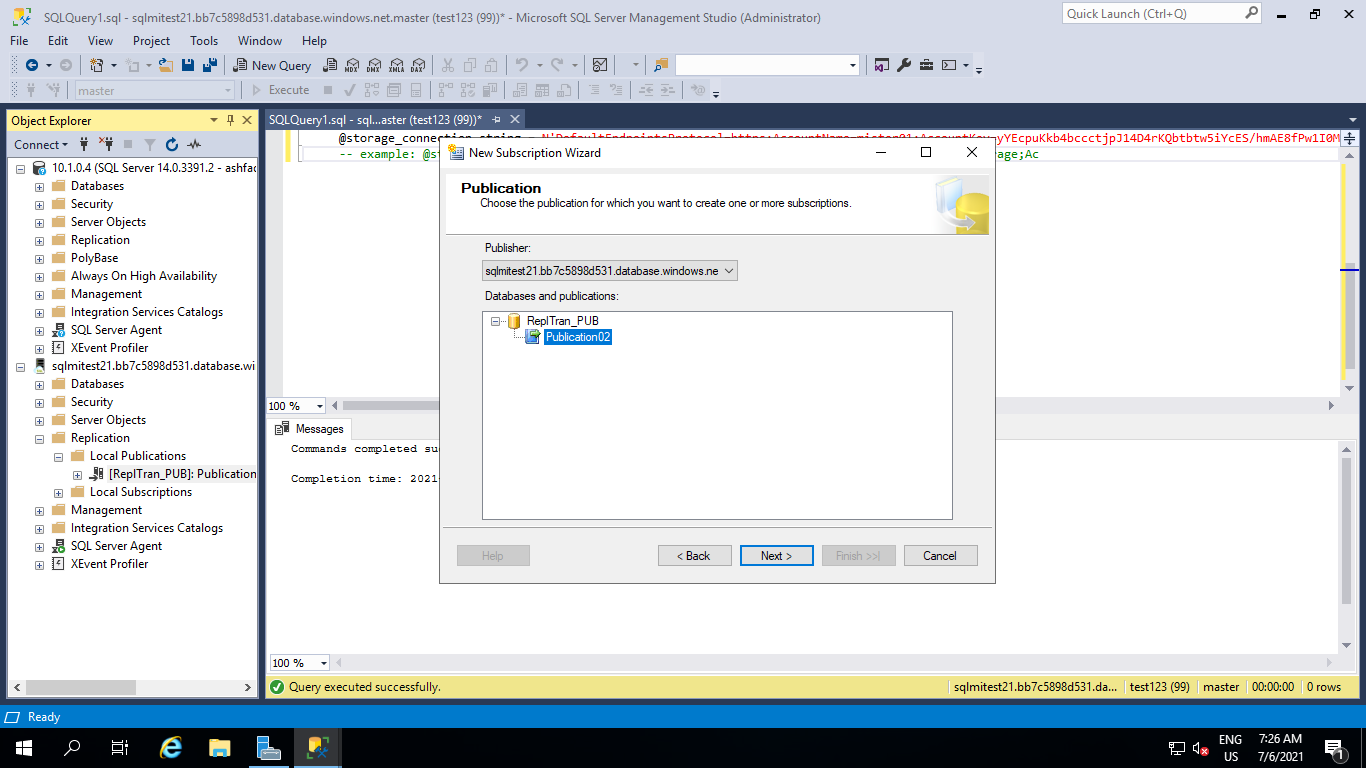
Now when distribution and publication are configured, we need to create a subscription as the last piece of transactional replication setup. In the context menu of the Publication we’ve created, we are going to click on New Subscriptions. New Subscription Wizard will open. Let’s follow its steps.



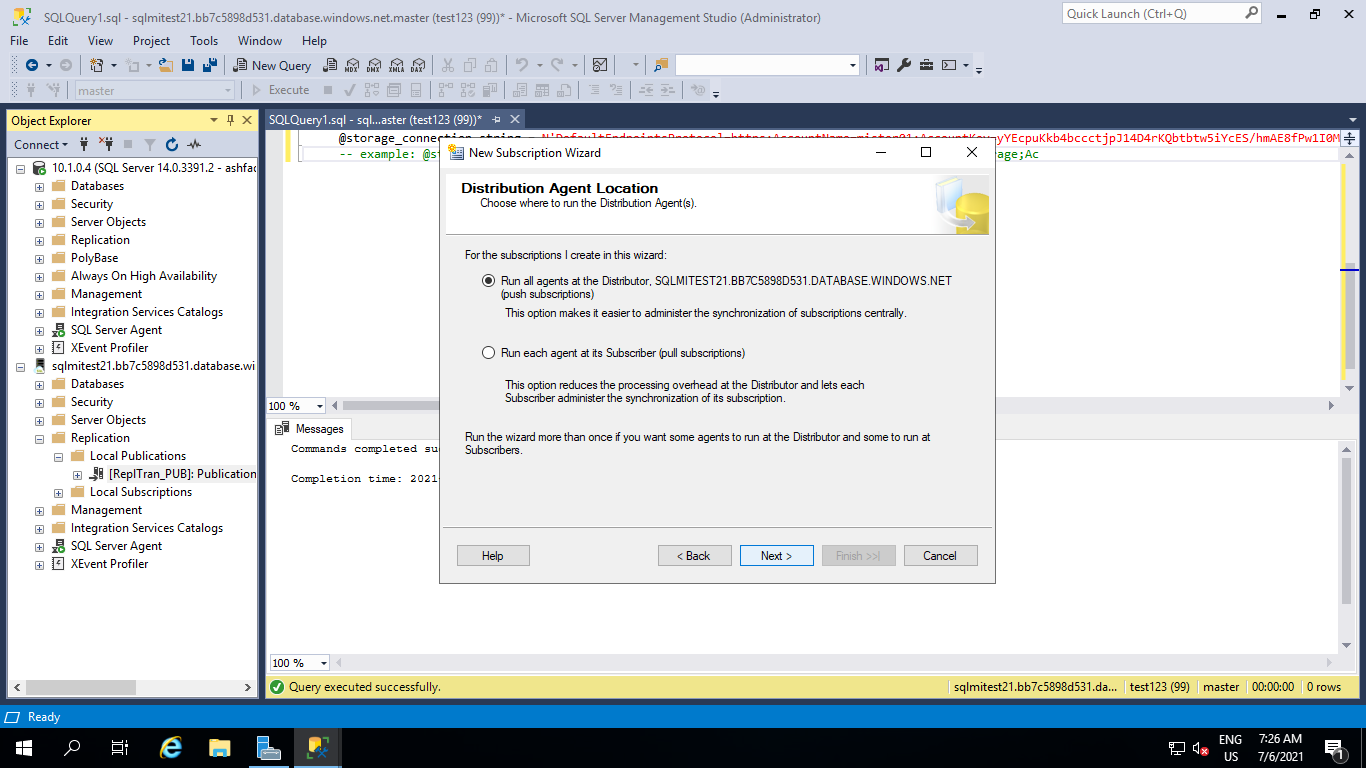
Step 5.2 New Subscription Wizard will open click on Next button.



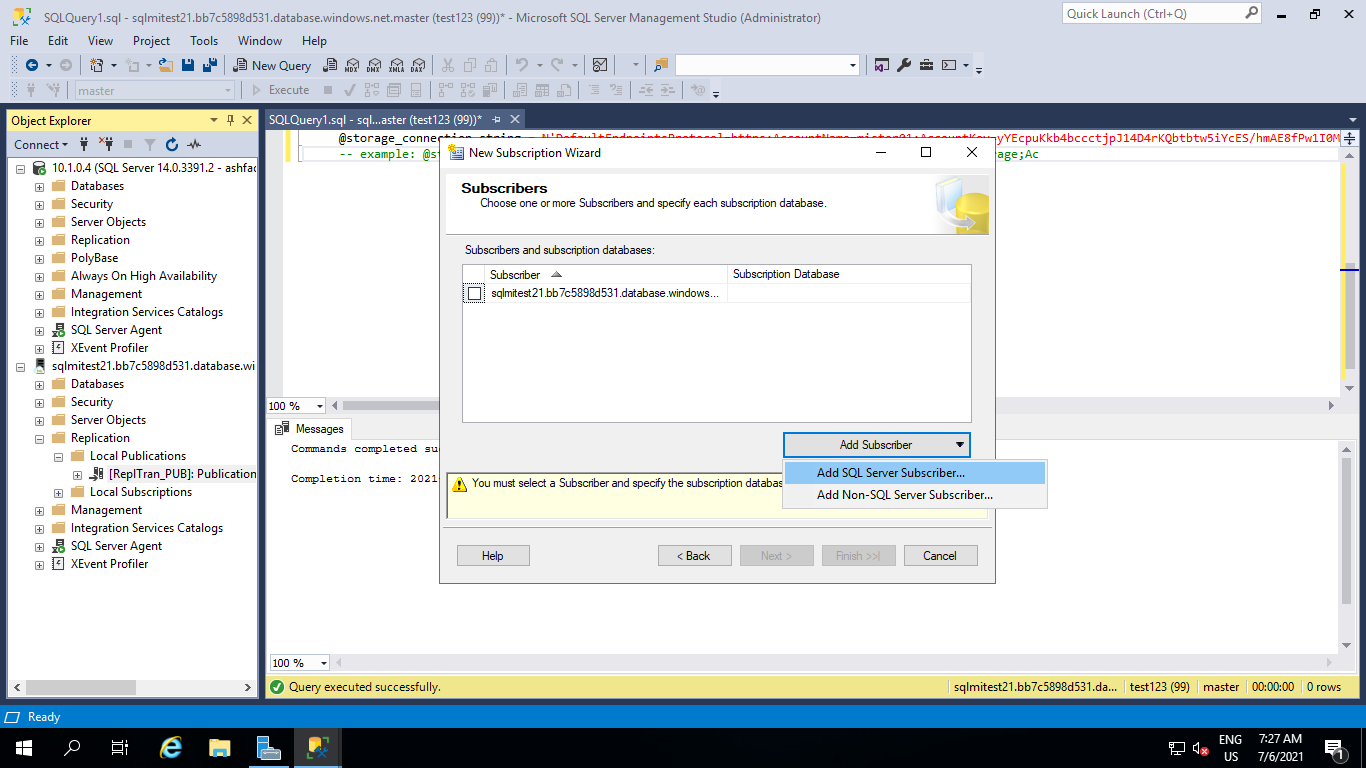
Step 5.3 On the **Publication** window, select the **Publisher**, database and the publication and click Next.



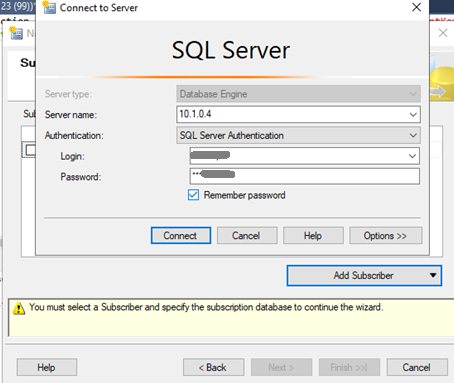
Step 5.4 **Distribution Agent Location** select option to **Run all agents at the Distributor** and click Next.



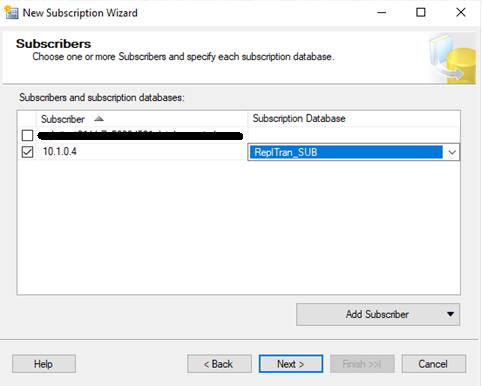
Step 5.5 On the Subscriber window, click on **Add Subscriber** and then **Add SQL Server Subscriber**.



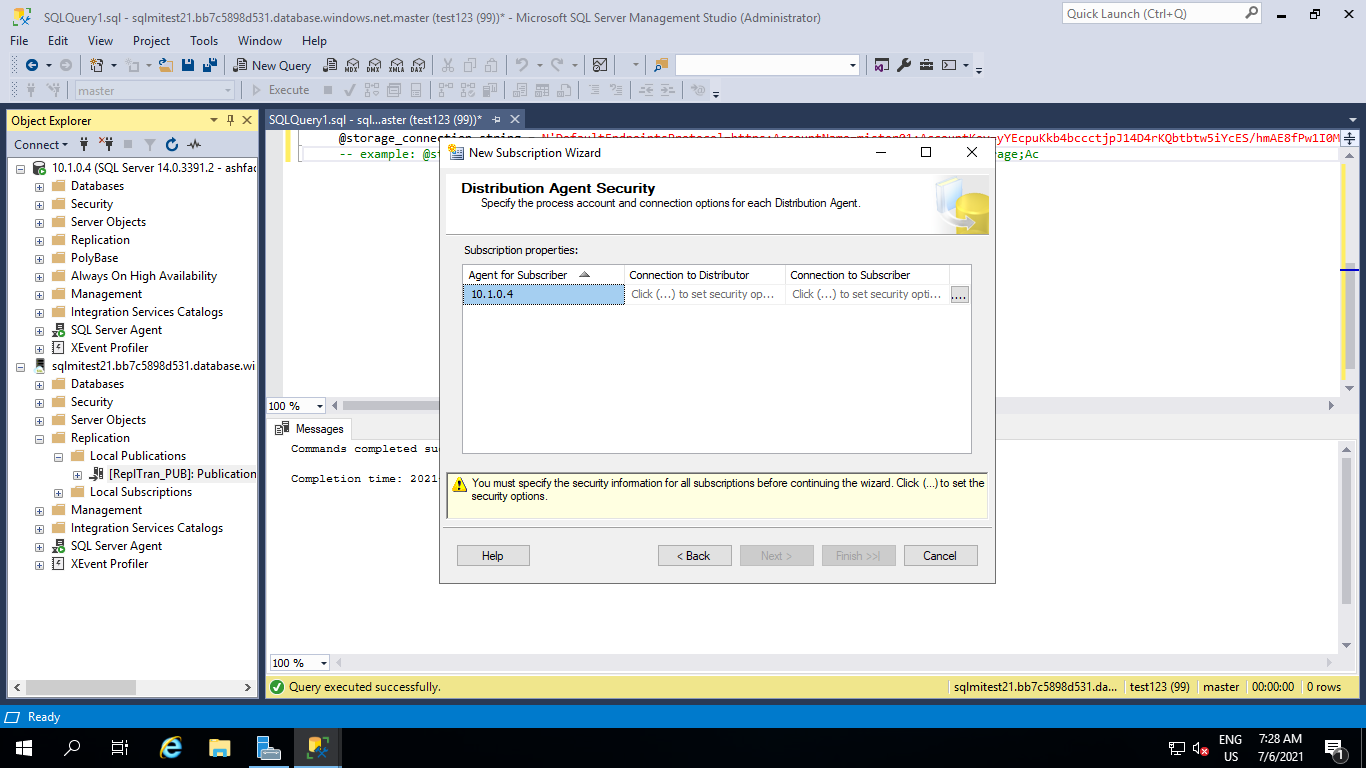
Step 5.6 hen connect to the SQL Server that will be the Subscriber. This SQL Server will be the target for the database migration. Use the server’s local IP address for this connection.



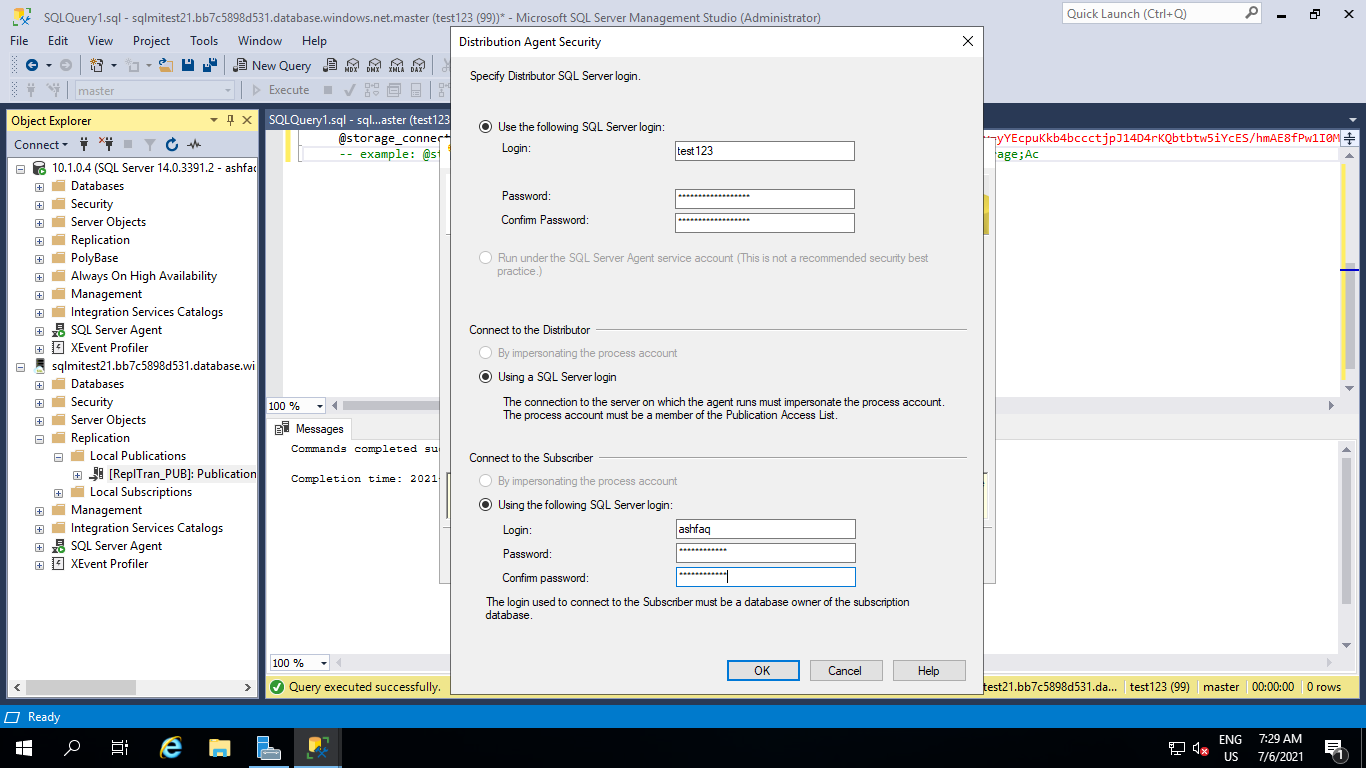
Step 5.7 For the newly added **Subscriber**, for the **Subscription Database** chose **New database** option from the drop down menu, and in the **New Database** window, provide the Database name. This will be the name of the target database for the migration. Then click OK and then Next.



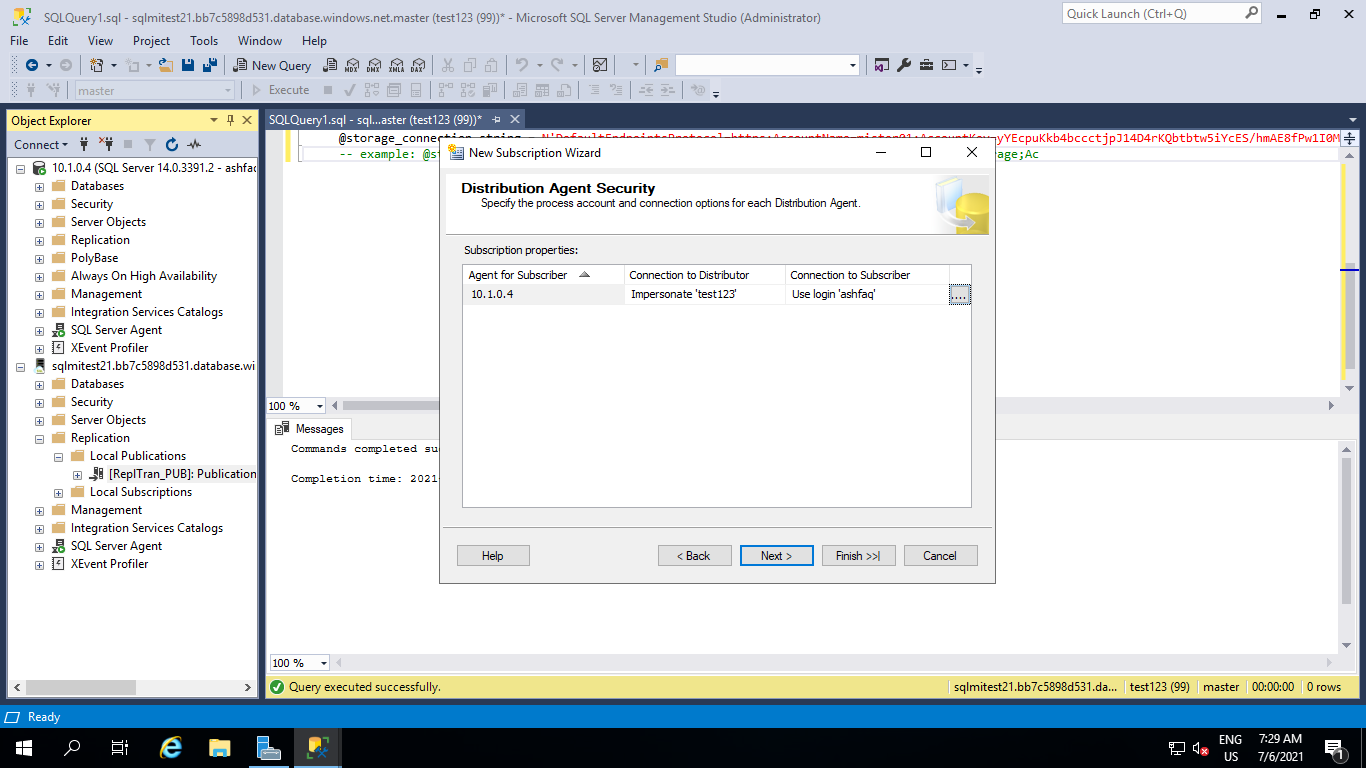
Step 5.8 On the **Distribution Agent Security** window, click on the button with three dots,



Step 5.9 **Distribution Agent Security** window provide login and password for Distributor and Subscriber.



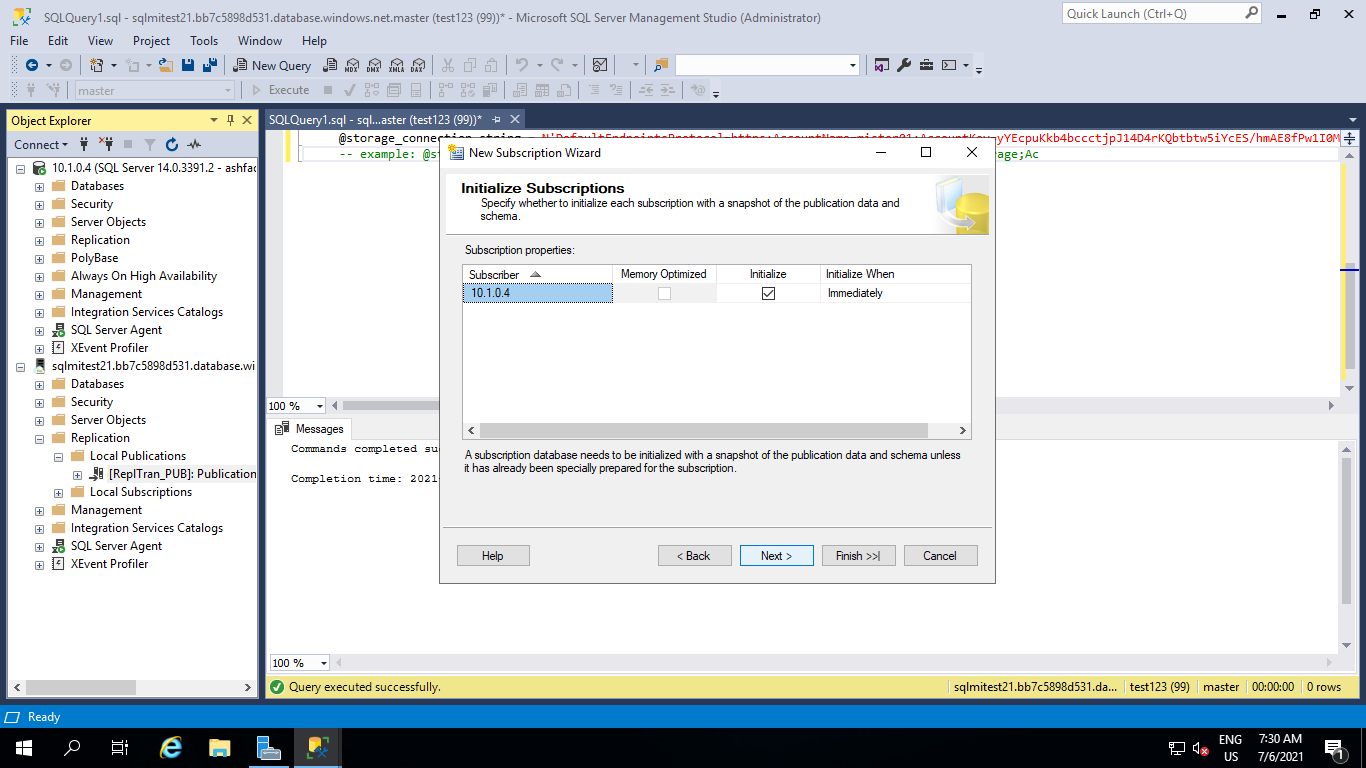
Step 5.10 Click on Next button.



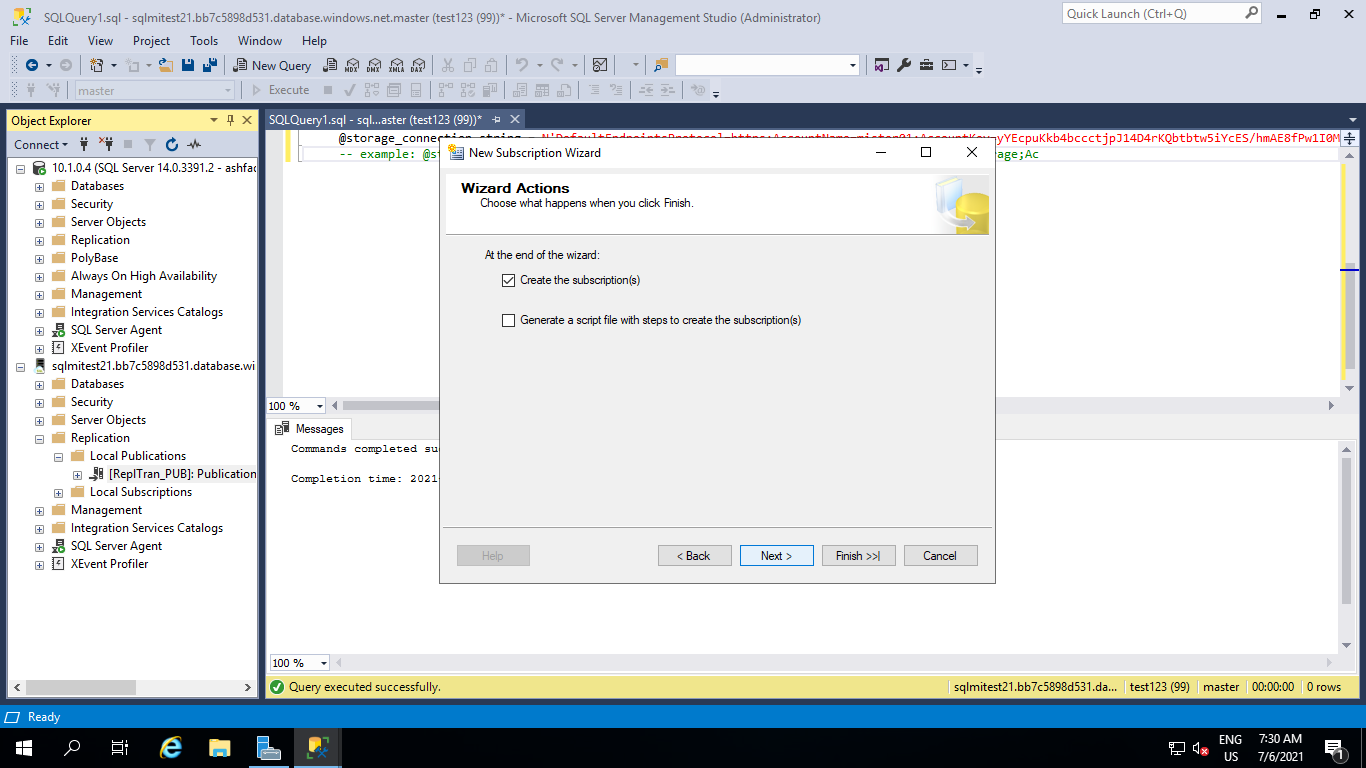
Step 5.11 **Synchronization Schedule** leave the **Agent Schedule** set to **Run continuously**. This will enable continuous data flow from the source database to the destination and help with the online migration.



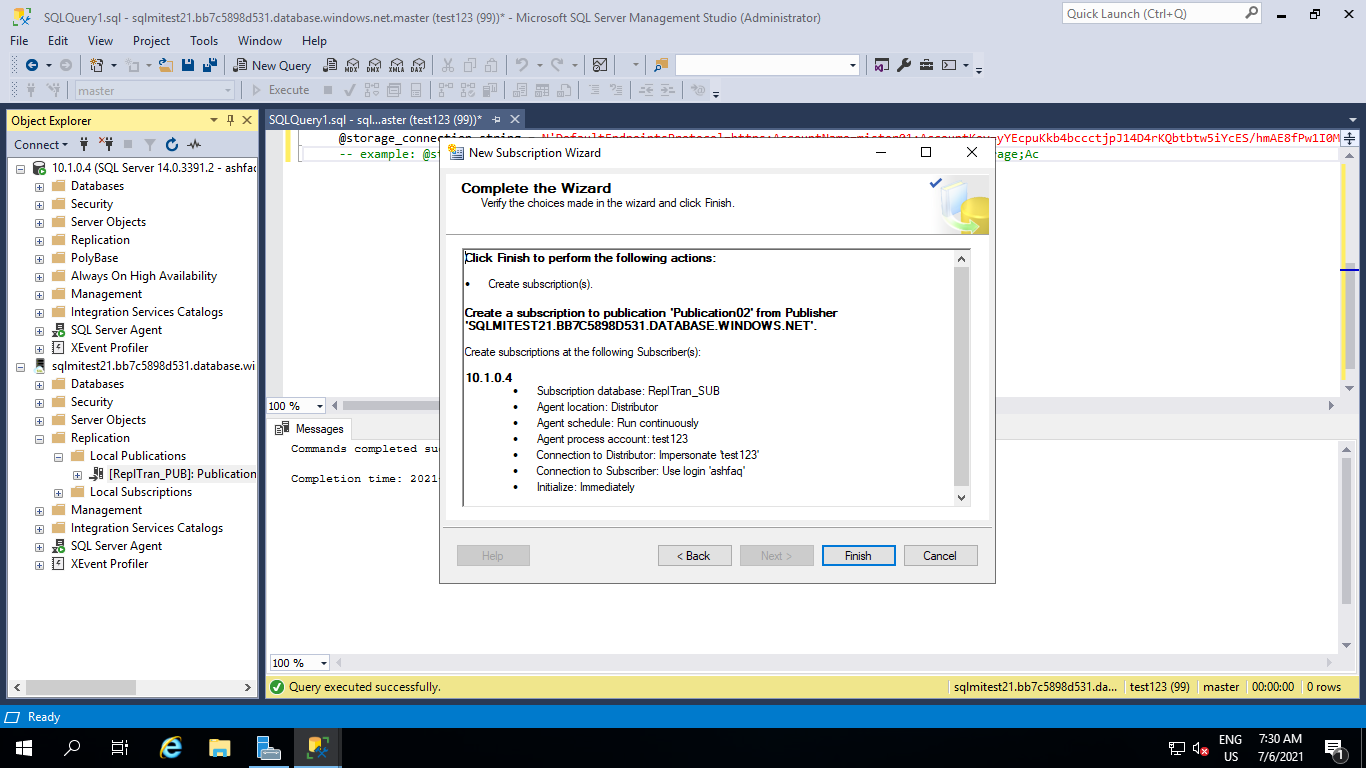
Step 5.12 On the **Initialize Subscription** window leave **Initialize When** set to **Immediately Click on Next**.



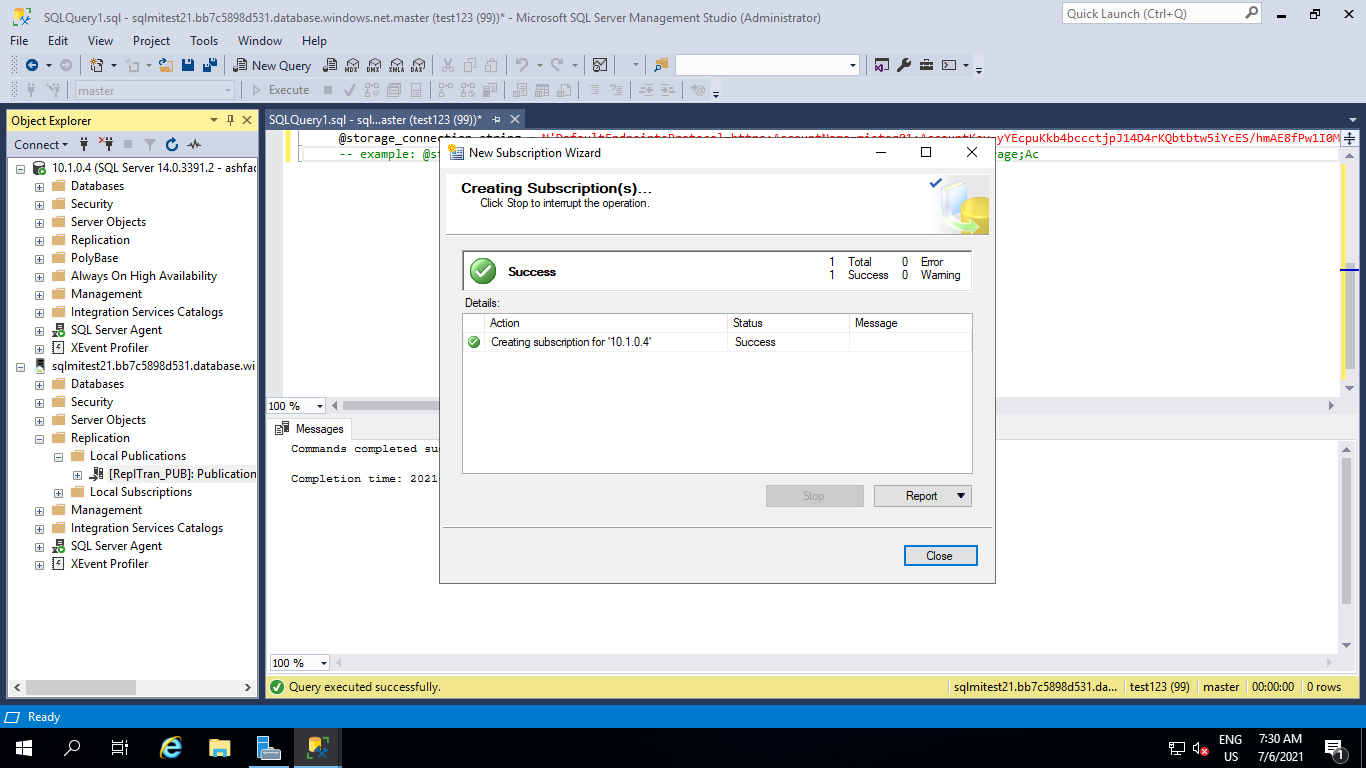
Step 5.13  **Wizard Actions**, leave Create the subscription(s) checked and click Next.



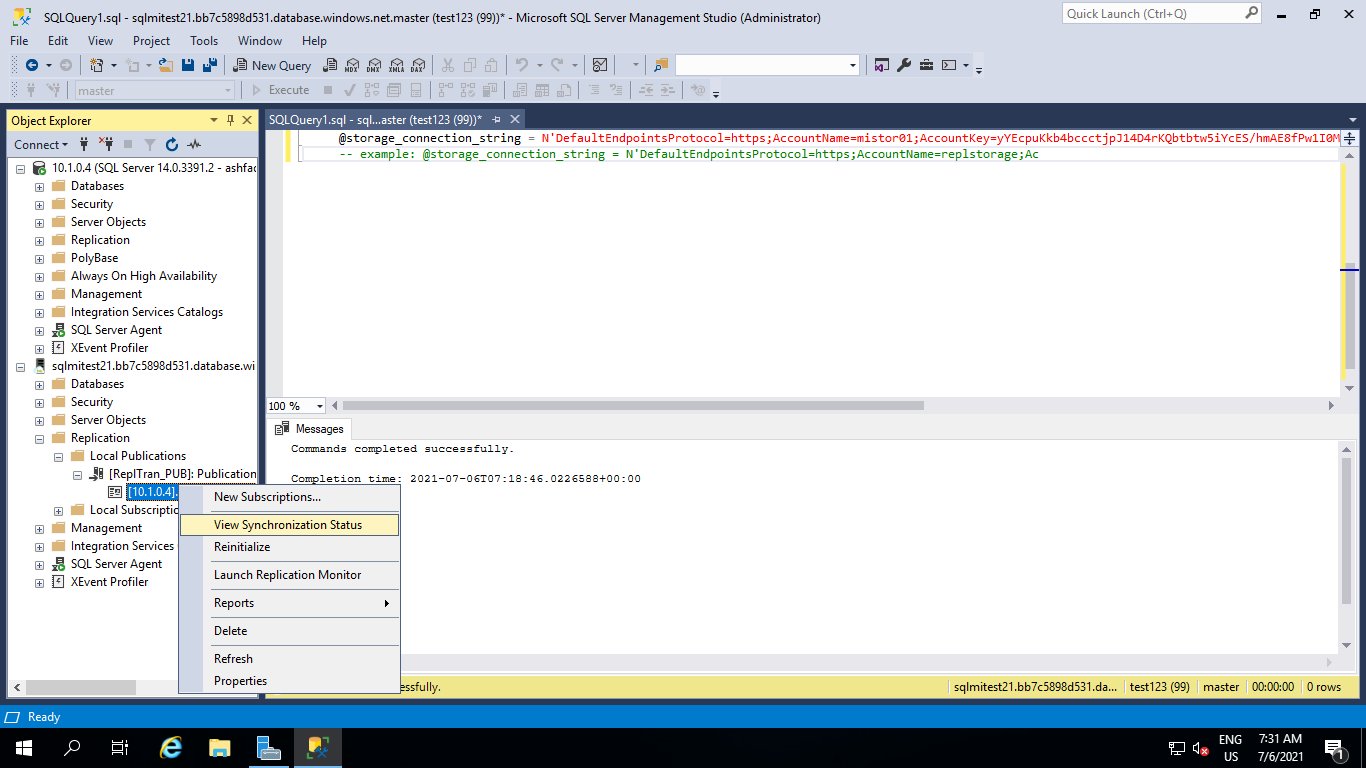
Step 5.14 Complete the Wizard click Next .

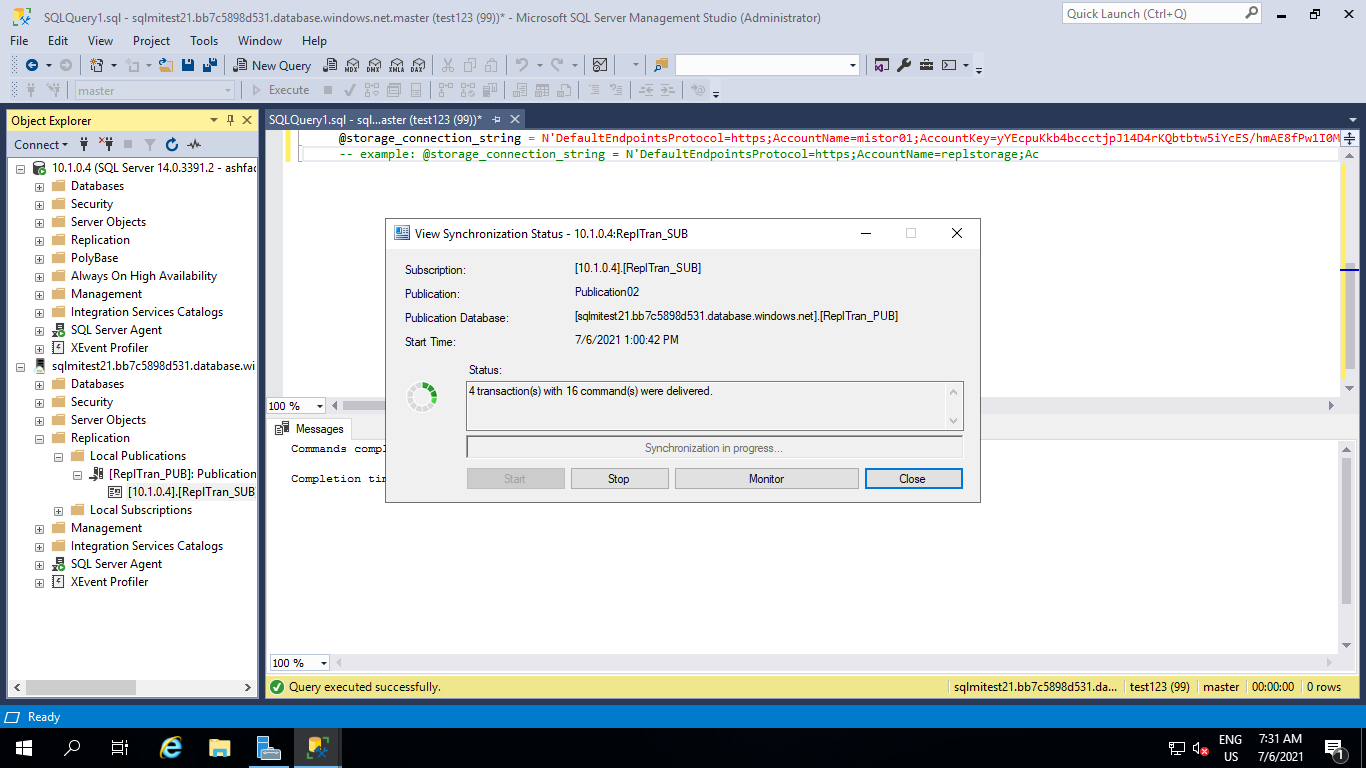


Step 5.15 Success status on the final **Creating Subscription(s)** step Click on close



Step 6 Once this is done, from the context menu of the newly created subscription, open the **View Synchronization Status**. You will see its progress and eventually all transactions will be replicated.

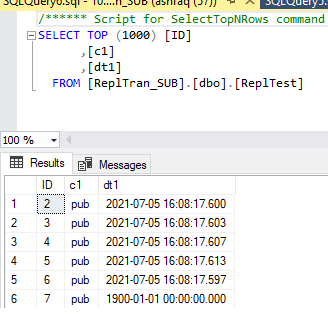




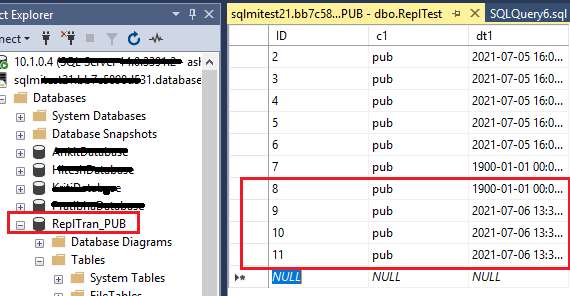
**Step 7** Test replication.

Once replication has been configured, you can test it by inserting new items on the publisher and watching the changes propagate to the subscriber.

**Step 7.1**  **ReplTran\_sub** is my Subscriber database and **ReplTest** is my table and some data.



**Step 7.2** Add some more new rows to publisher table **ReplTran\_PUB**.



**Step 7.3** Now Check the Subscriber Table.



### E . Known errors:

**1** Could not connect to Subscriber**:** Verify your VNet peering is configured correctly. or you may useing public IP.

**2** Failed to connect to Azure Storage: Using a forward slash instead of a backslash in the file path for the file share can cause this error. Ex: **\\replstorage.file.core.windows.net\replshare**

**3** Windows logins are not supported: use SQL Authentication.