## **Data Ingestion into Azure SQL Datawarehouse using SSIS**

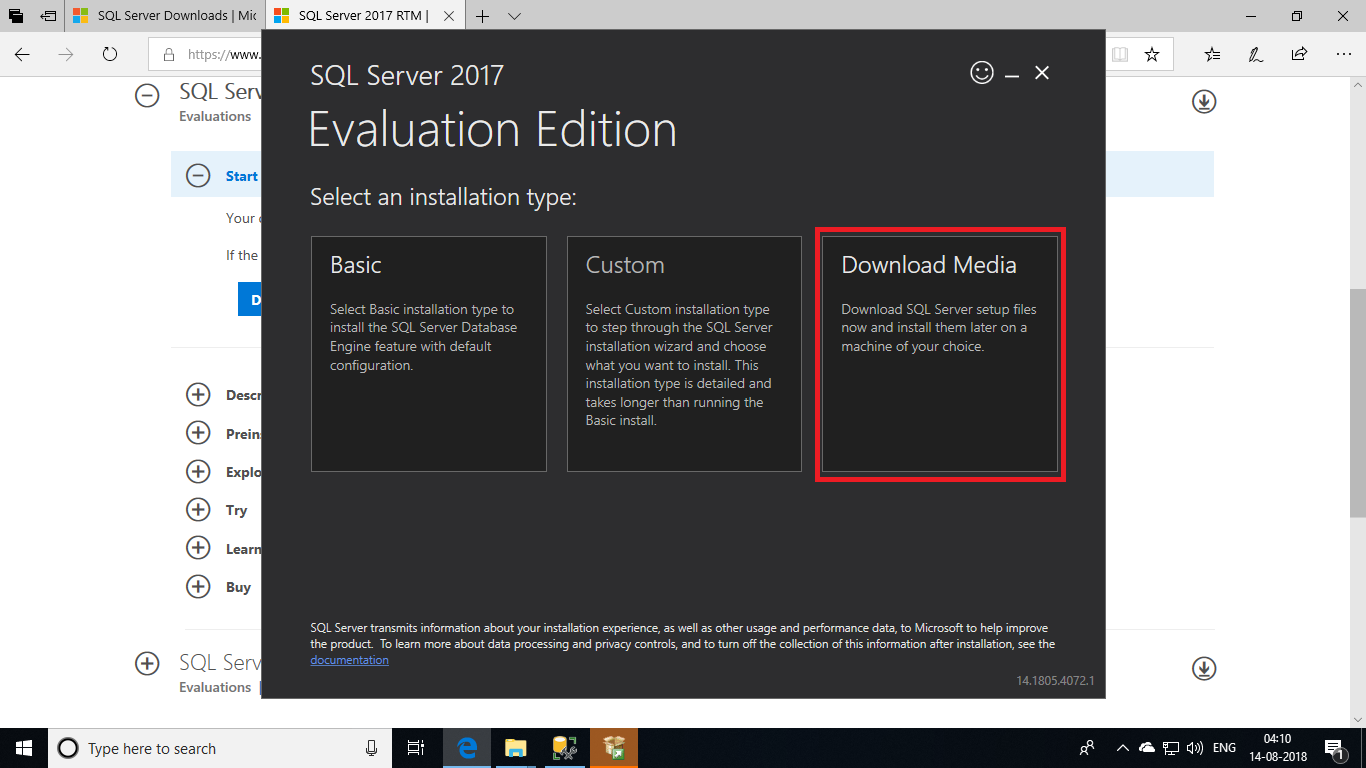
This demo is to load data that is available in an on-premises SQL database. For implementing this, we need a few prerequisites as shown below.

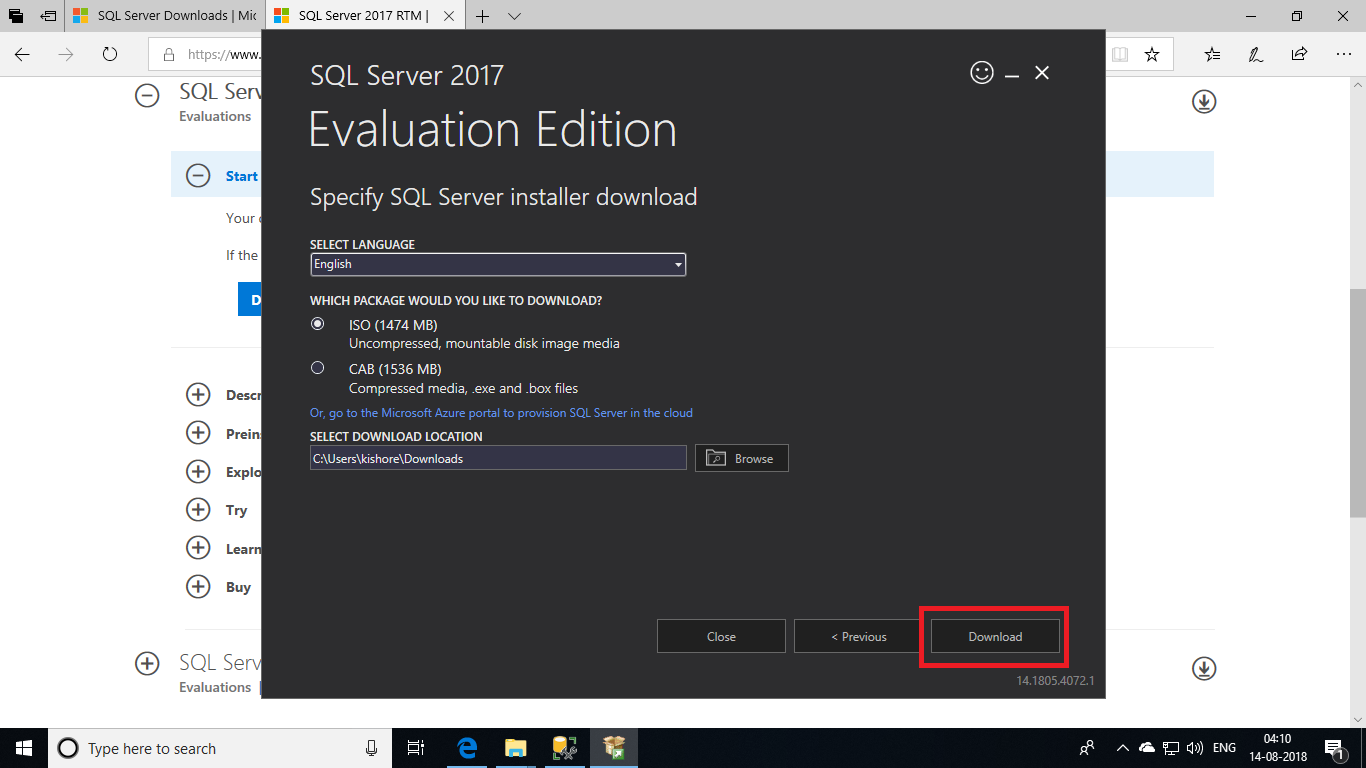
## **Prerequisites:**

1. **SQL Server Integration Services (SSIS)** – You can create a windows Azure virtual machine with SQL server in it.
2. **Visual Studio** – Azure virtual machine can be preferred for the same.
3. **SQL Server Data Tools for Visual Studio (SSDT)**. To get SQL Server Data Tools for Visual Studio, see <https://go.microsoft.com/fwlink/?linkid=875613> . Download the file and install in the VM where you have your SQL Server and Visual Studio.
4. **An Azure SQL Data Warehouse database -** This tutorial connects to an SQL Data Warehouse instance and loads data into it. Hence you need an Azure SQL Data Warehouse created in prior. Make sure that you create a bank SQL Data Warehouse and add VM’s IP address in the firewall access.

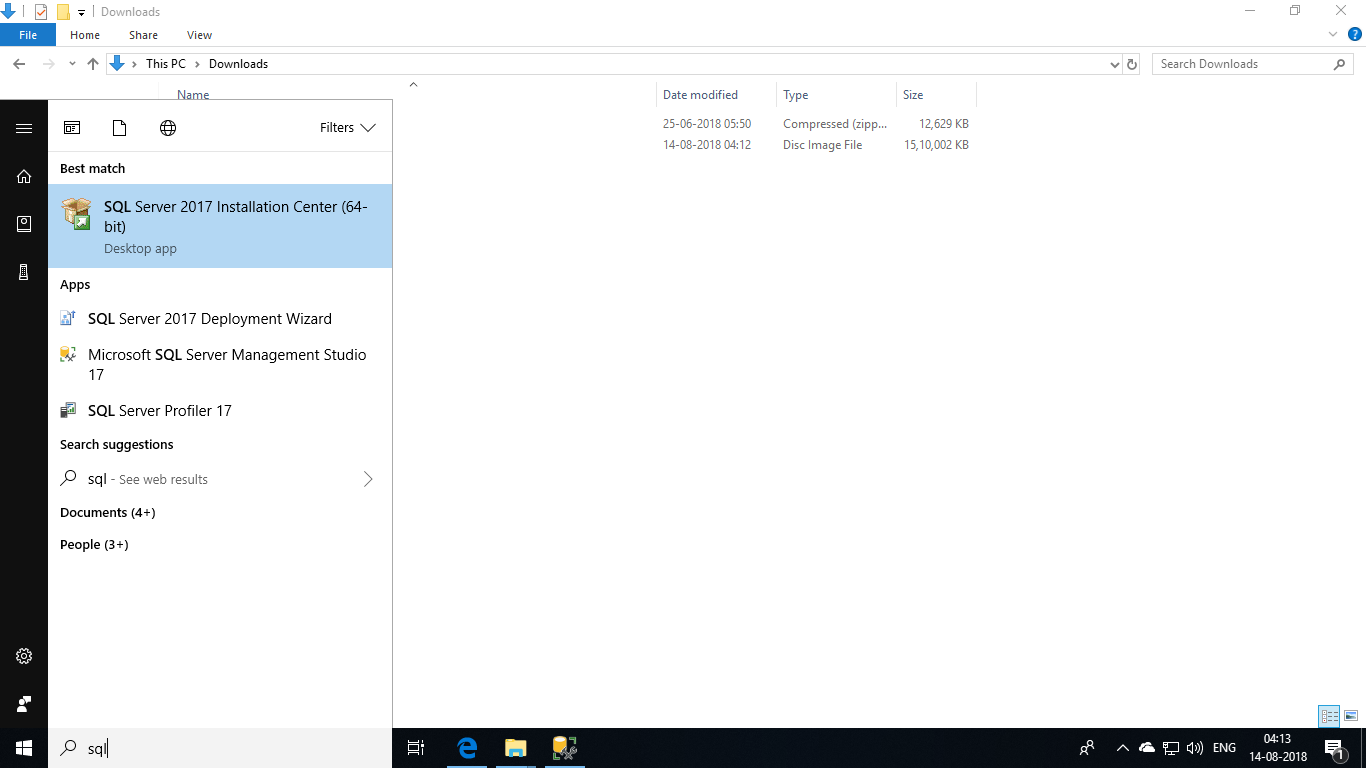
**Configuring SQL Server:**

Download SQL Server Setup and install it if in case you don’t have it installed. We need an additional service name **SQL Server Full Text Search** enabled in the SQL Server for implementing some sample workloads into them. Azure virtual machines do not come with this service enabled in them by default. Hence download the setup of SQL Server from here - <https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2017-rtm> to modify the SQL Server configuration in VM. After downloading it, open the ISO image.

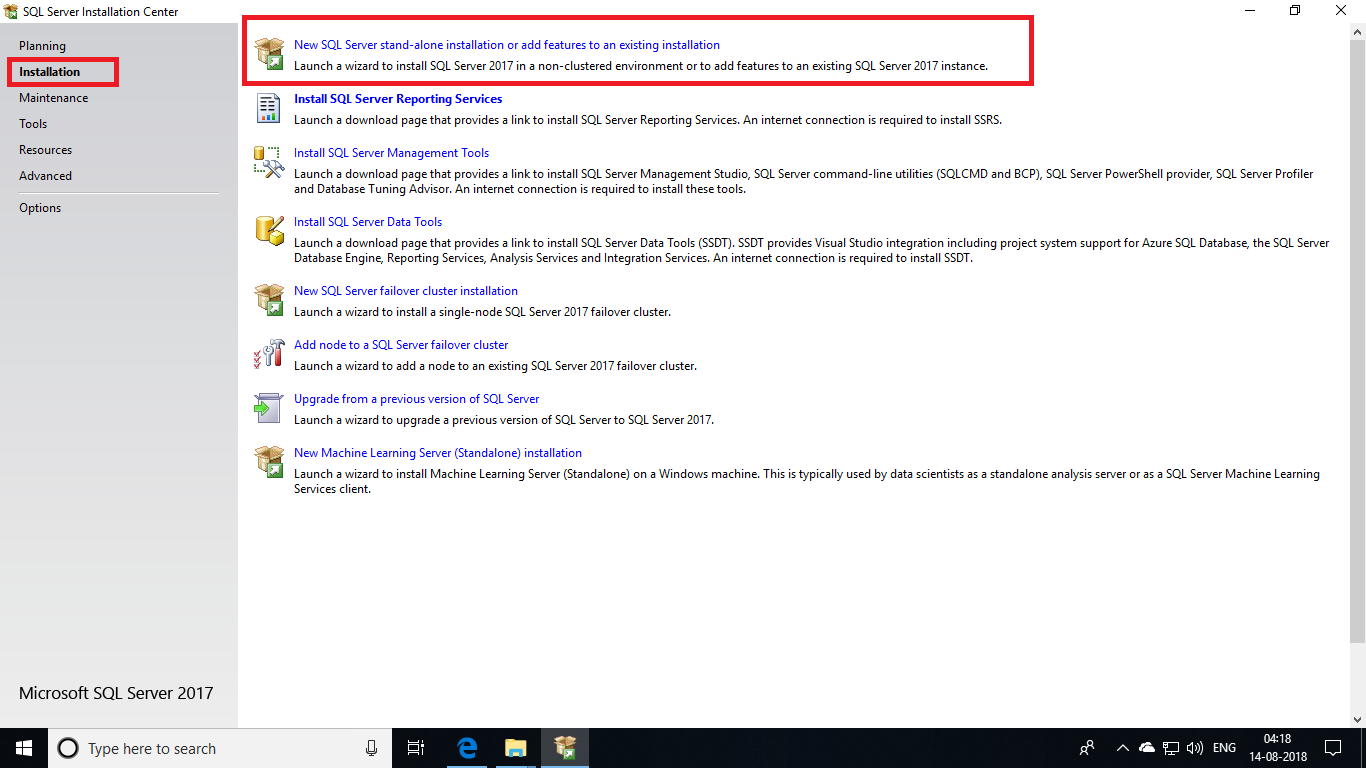




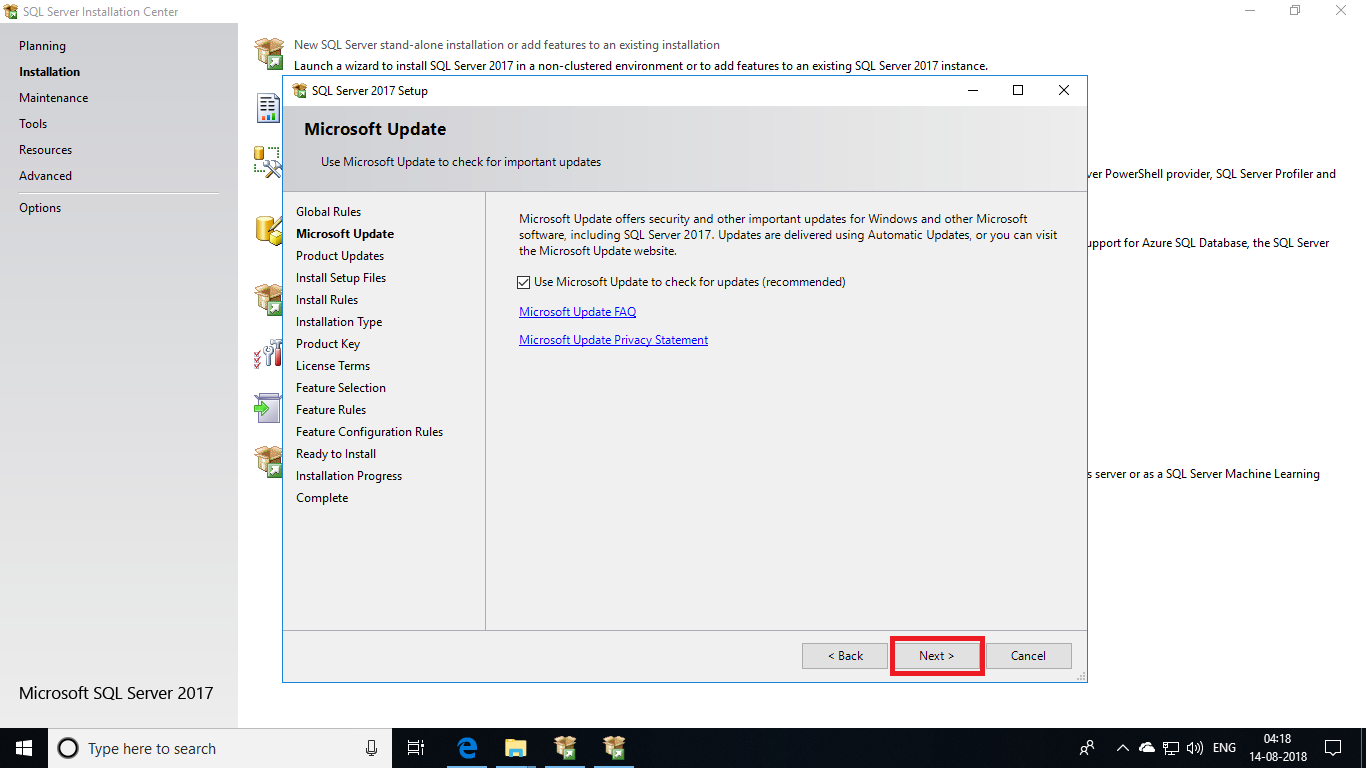
Now you can open the setup or find the SQL Server Installer in the start menu and open it.

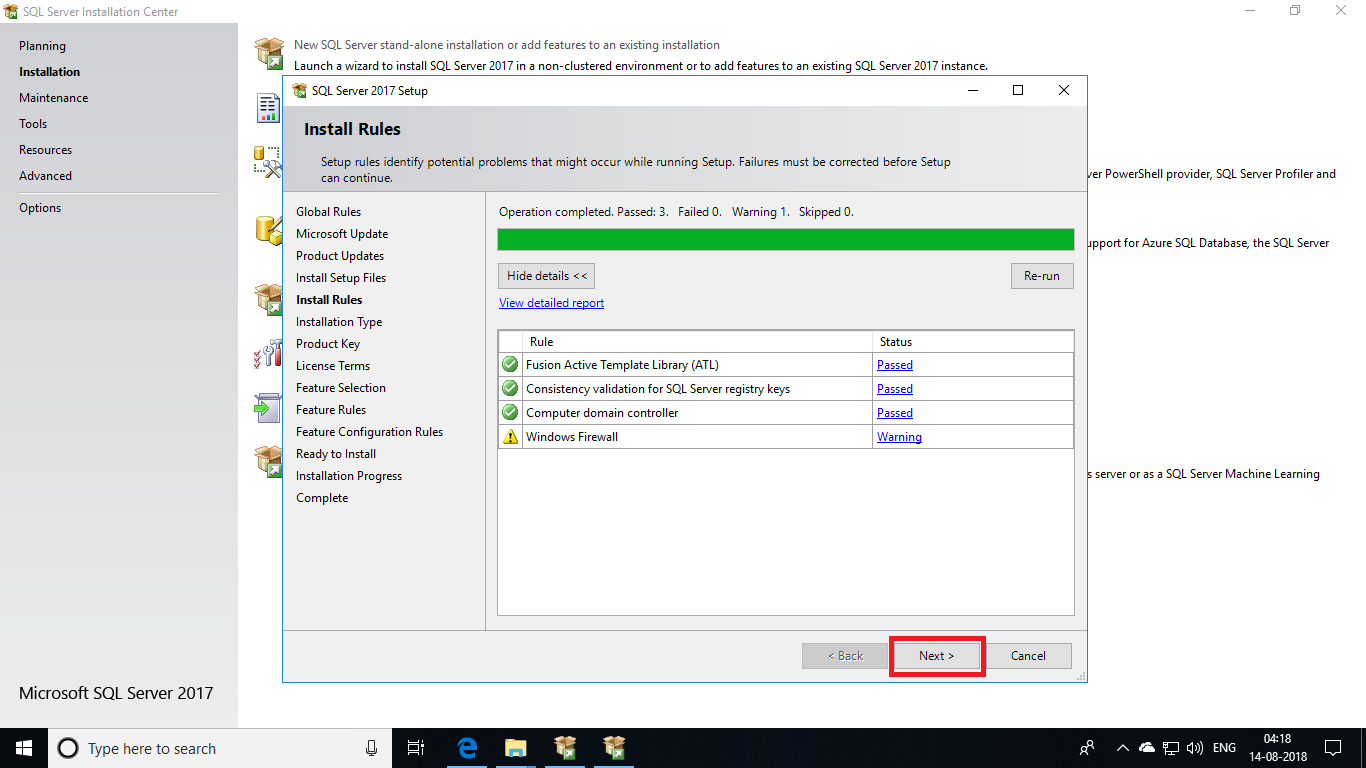


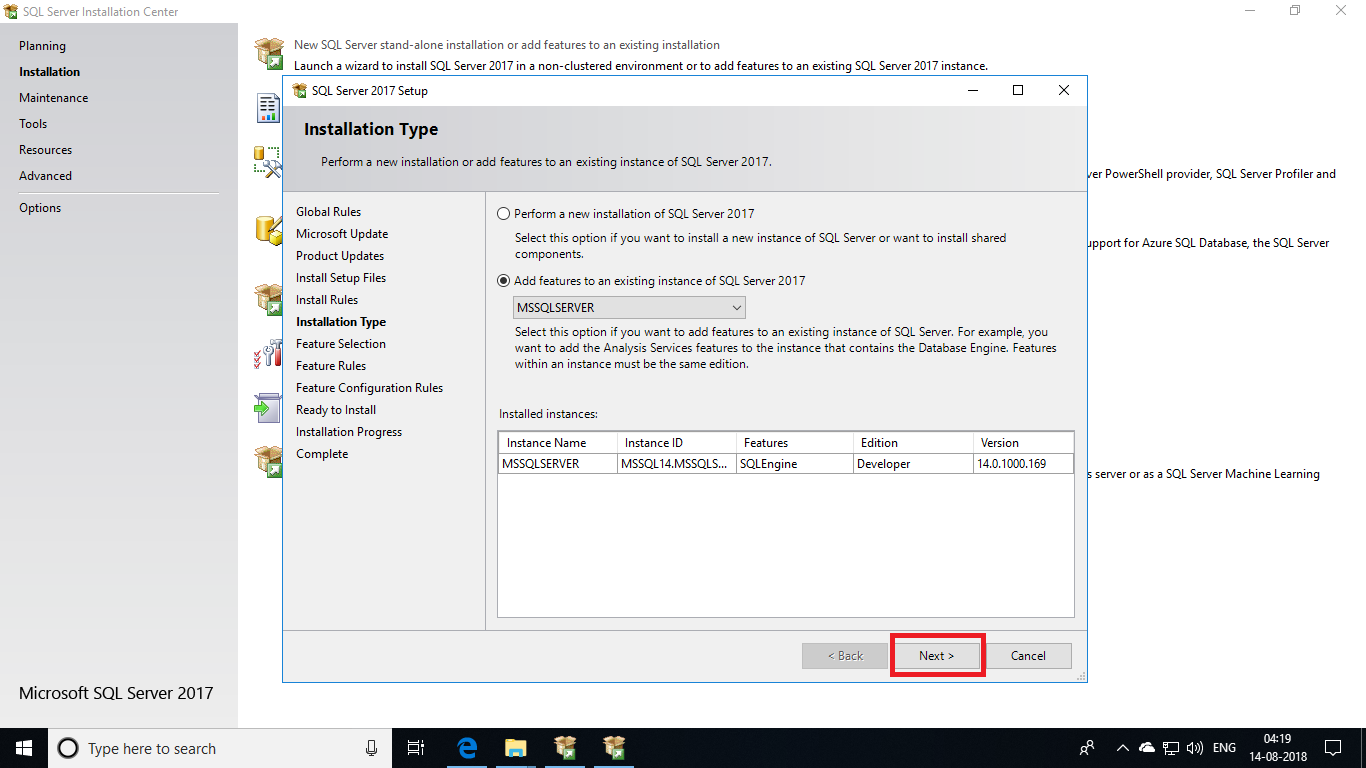
Once you open the setup menu, click on the **Installation** option and choose the menu as denoted below.



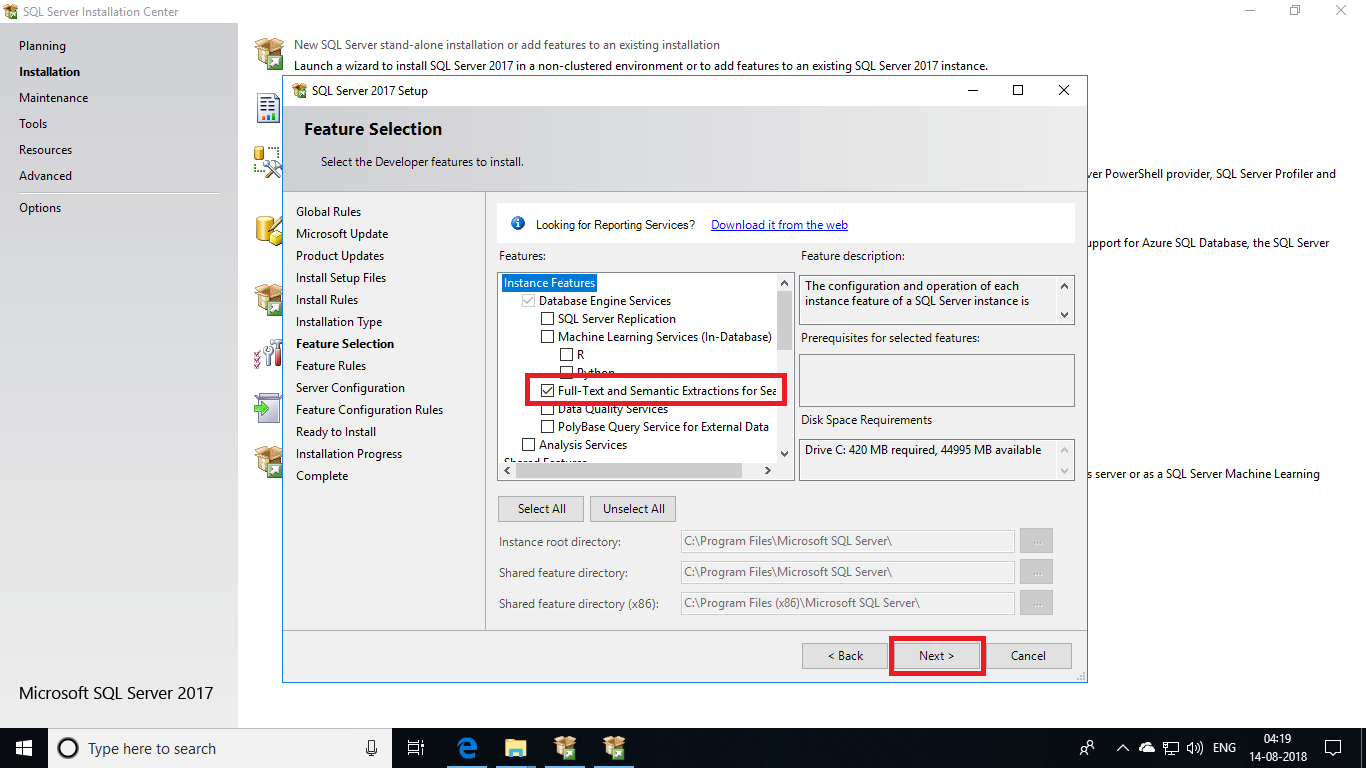
Follow the instructions in the below given screenshots.

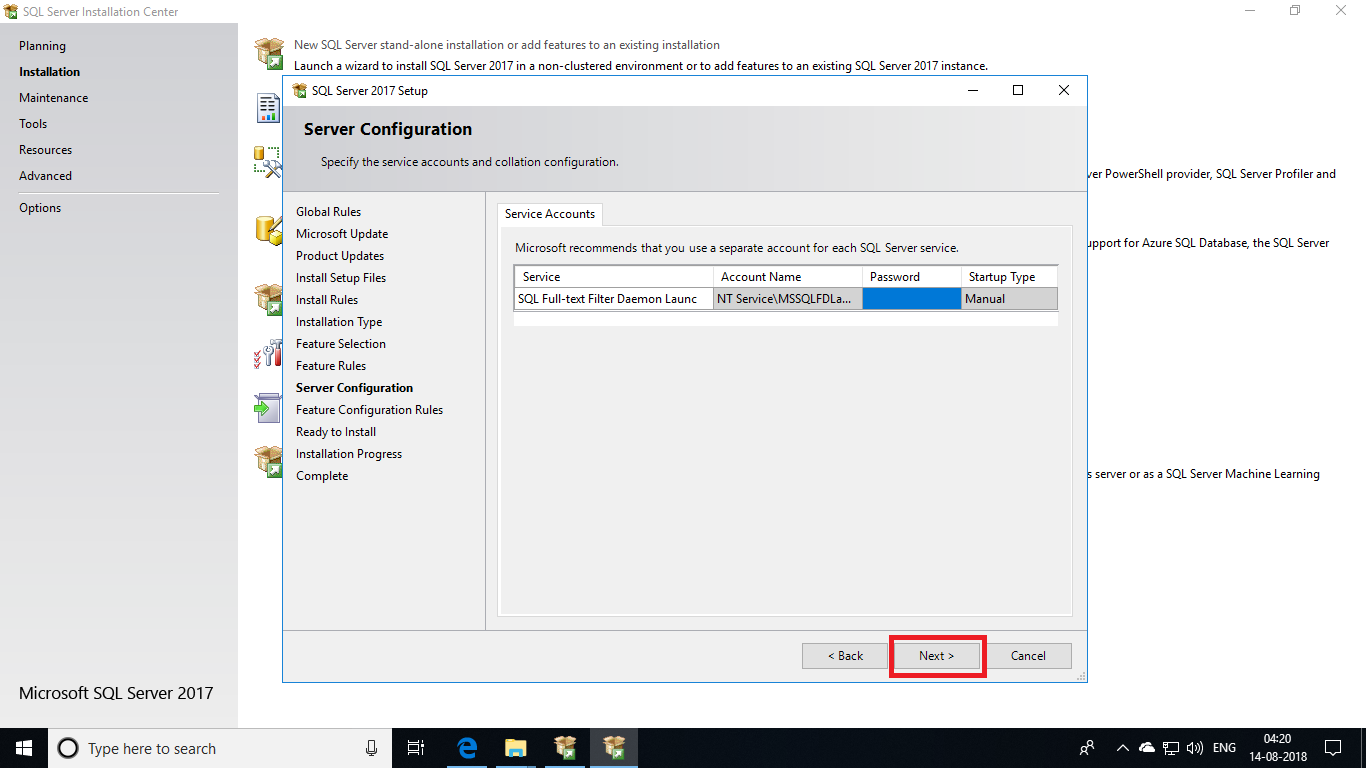


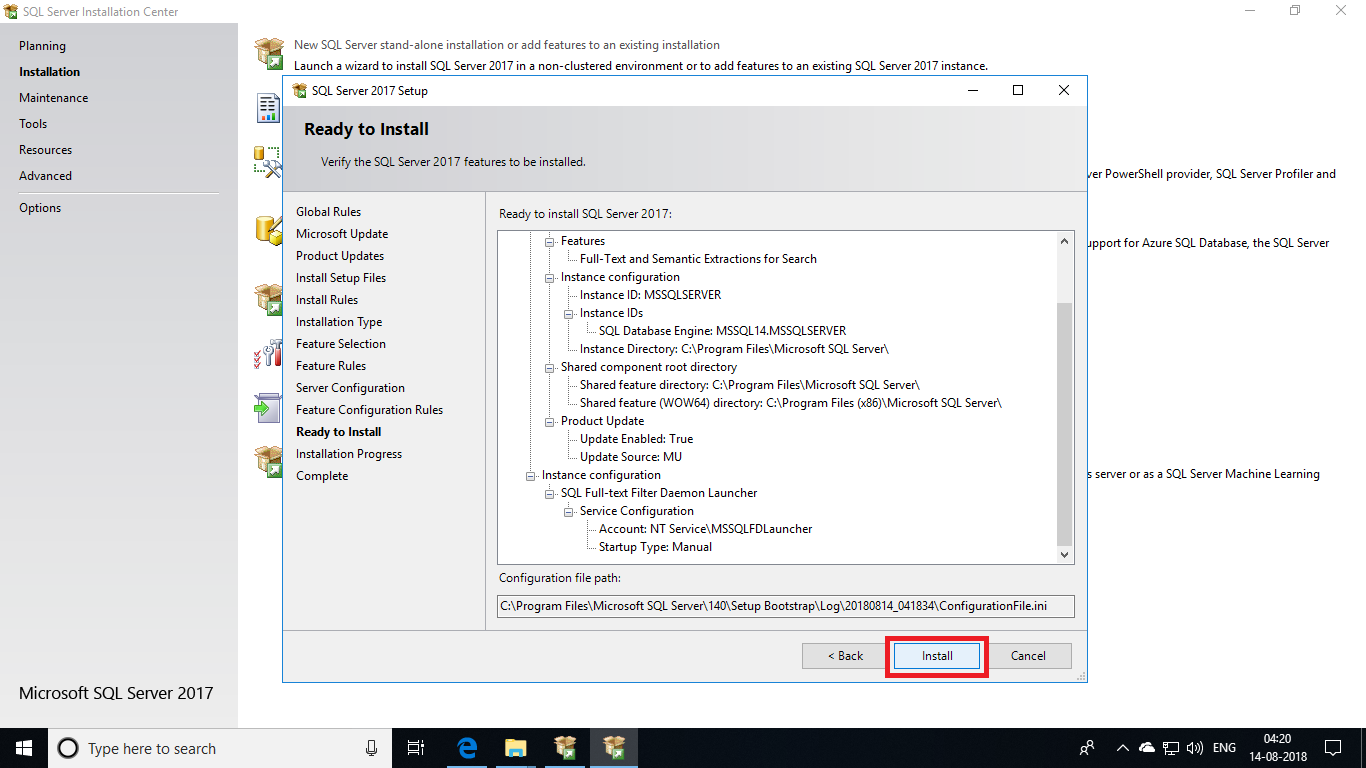


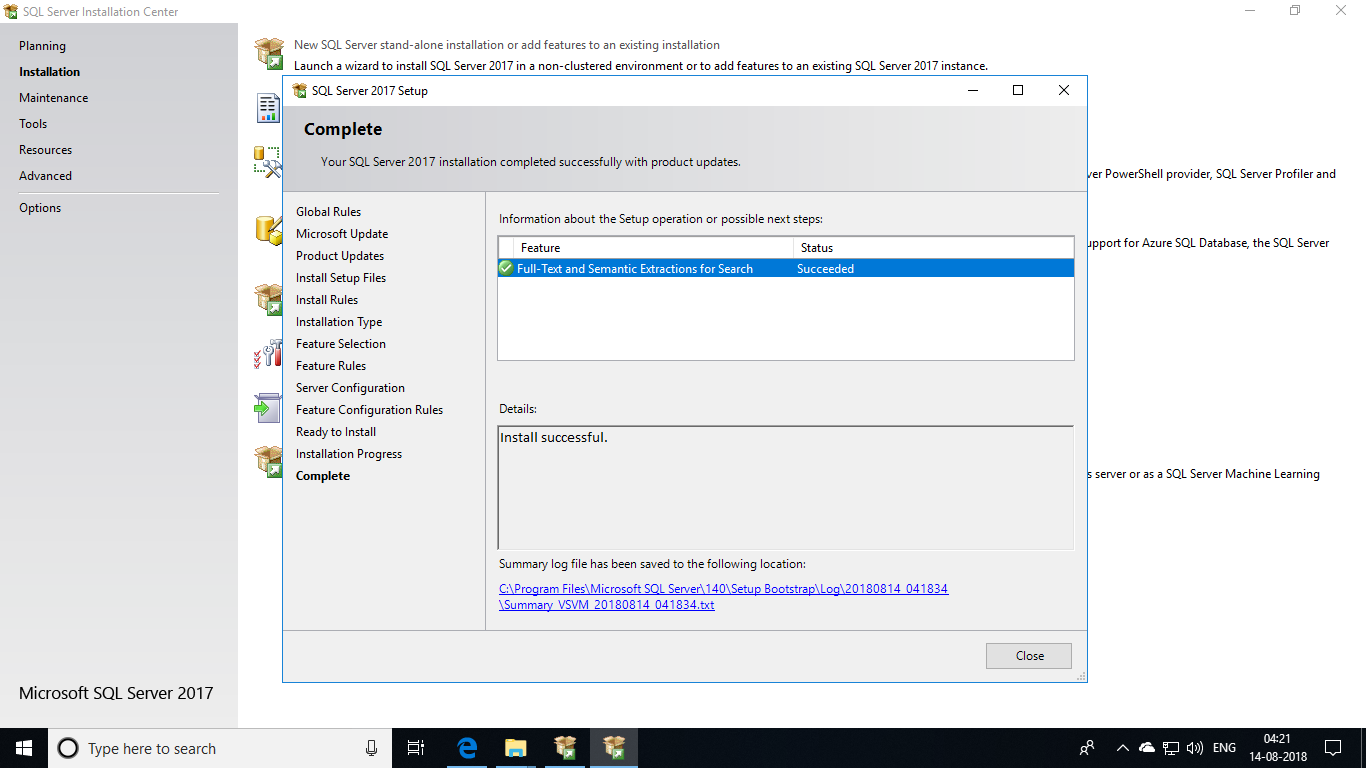


In this page, select the **Full Text and Semantic Extensions** and click **Next**.





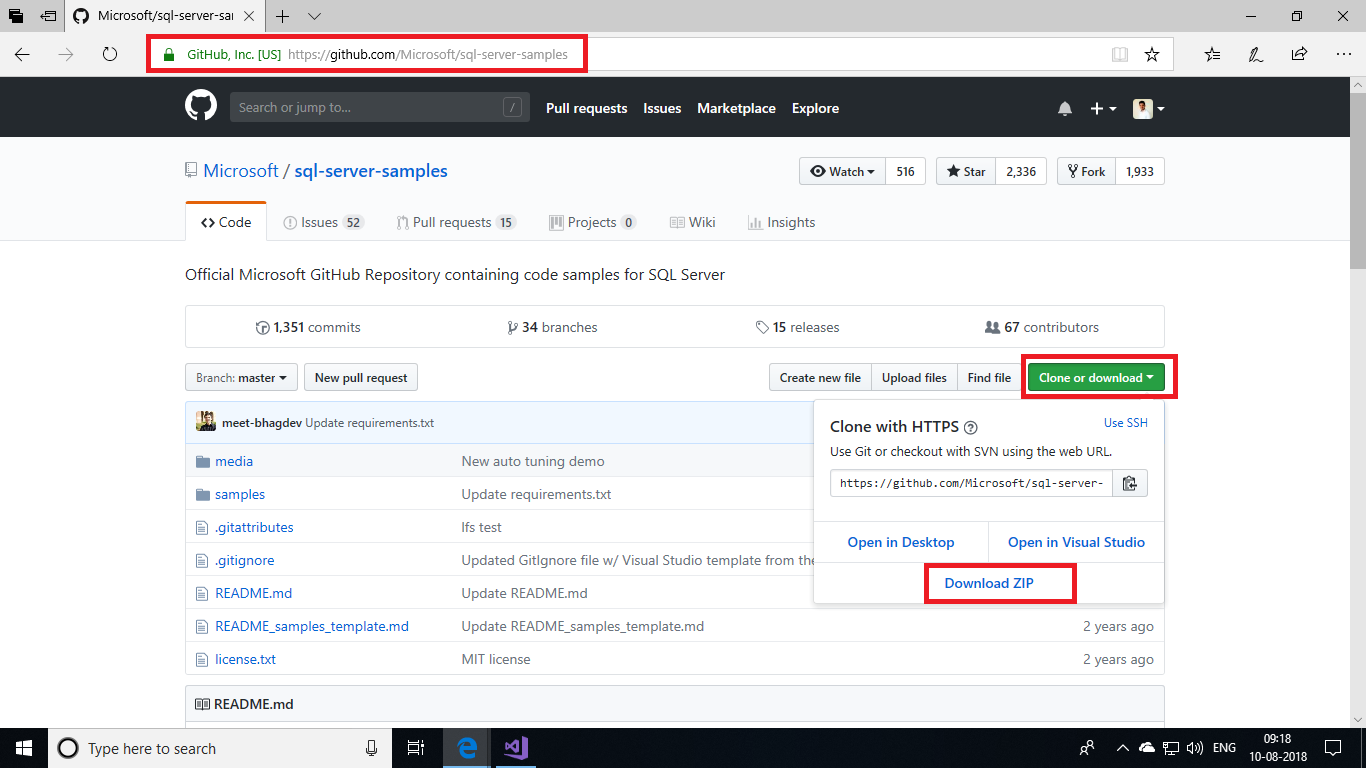


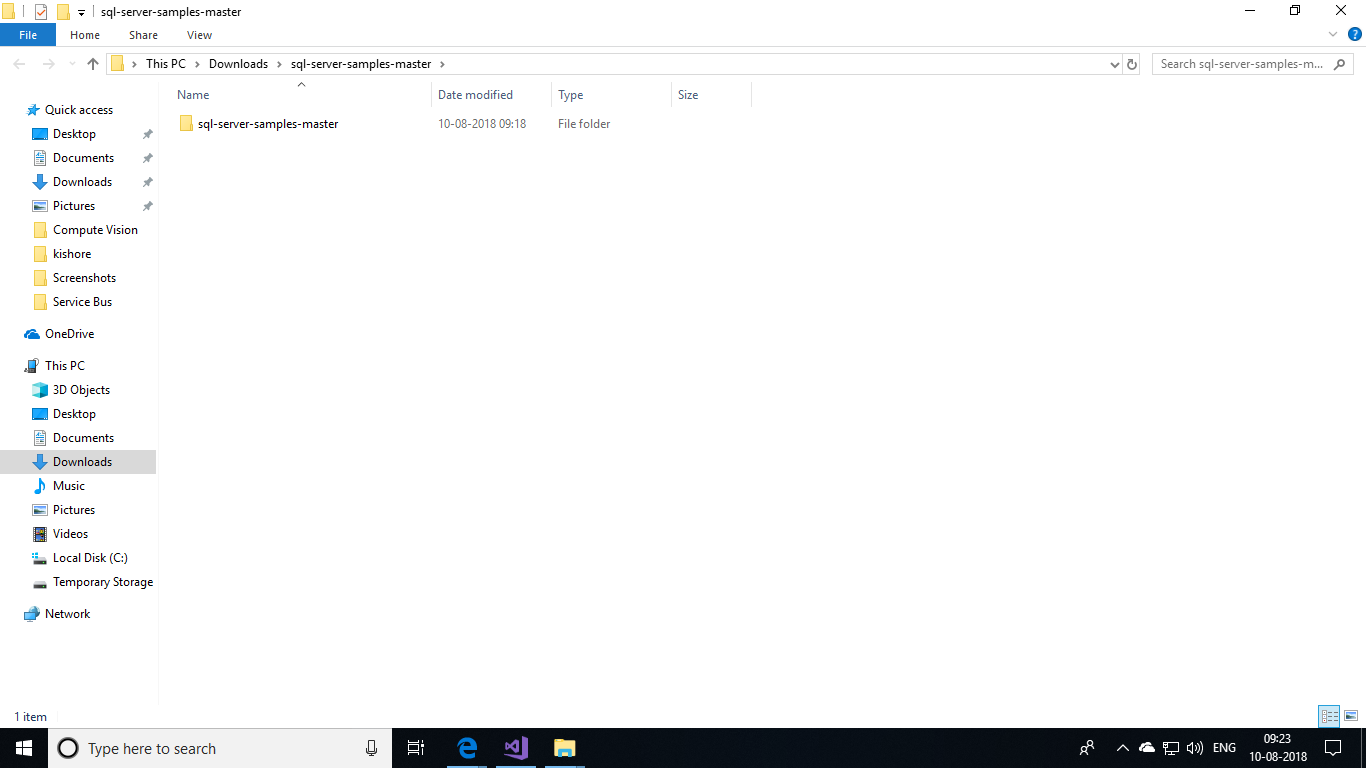


Now we have configured the SQL server with all the required features. Let’s upload a sample database into the server now.

**Adding Sample Data into SQL Database:**

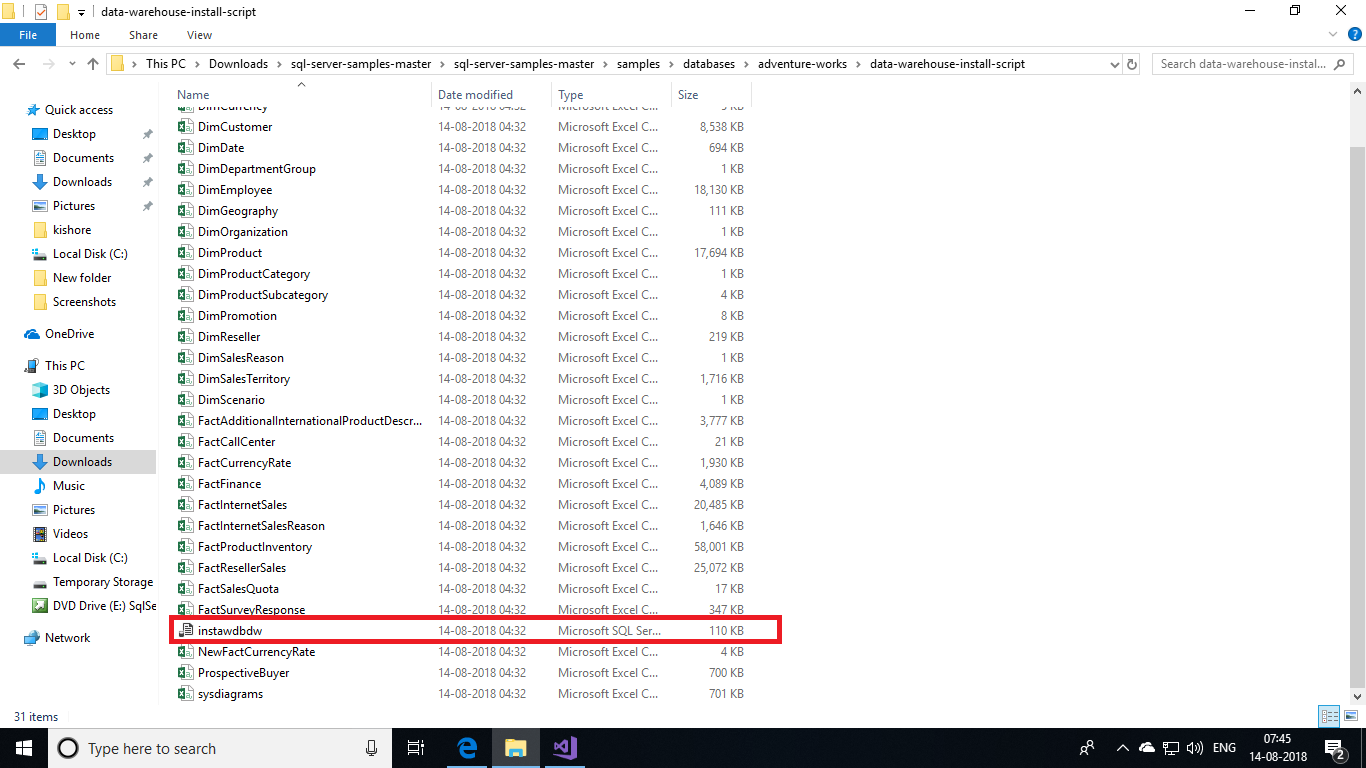
Download a sample dataset from the link - <https://github.com/Microsoft/sql-server-samples>. This dataset is in GitHub. Download it and extract it somewhere in your machine.



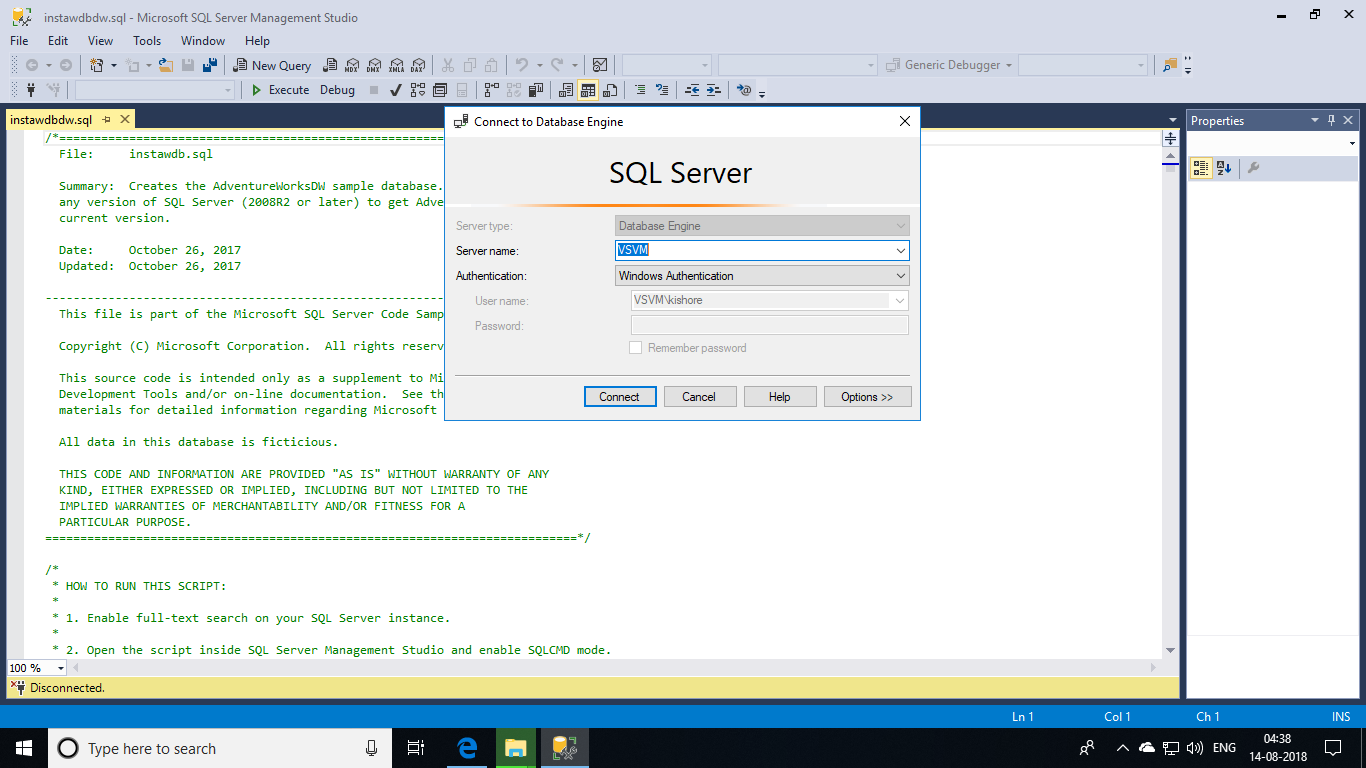


Navigate to a folder named data-warehouse-install-script by using the path **sql-server-samples-master\sql-server-samples-master\samples\databases\adventure-works\data-warehouse-install-script**

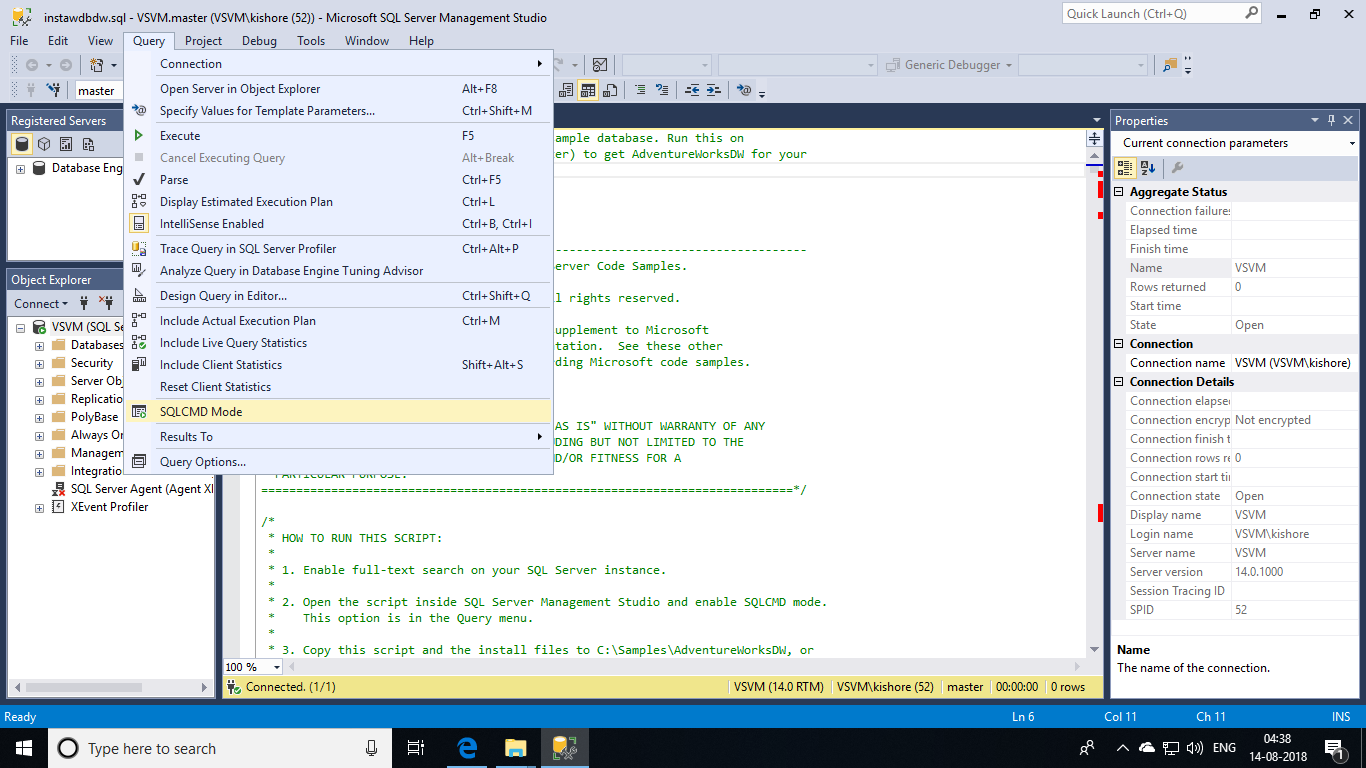
This is where the sample data is available. Now, let us add this sample data into the database. Open the file **instadwdb** which contains script to upload these data into database.



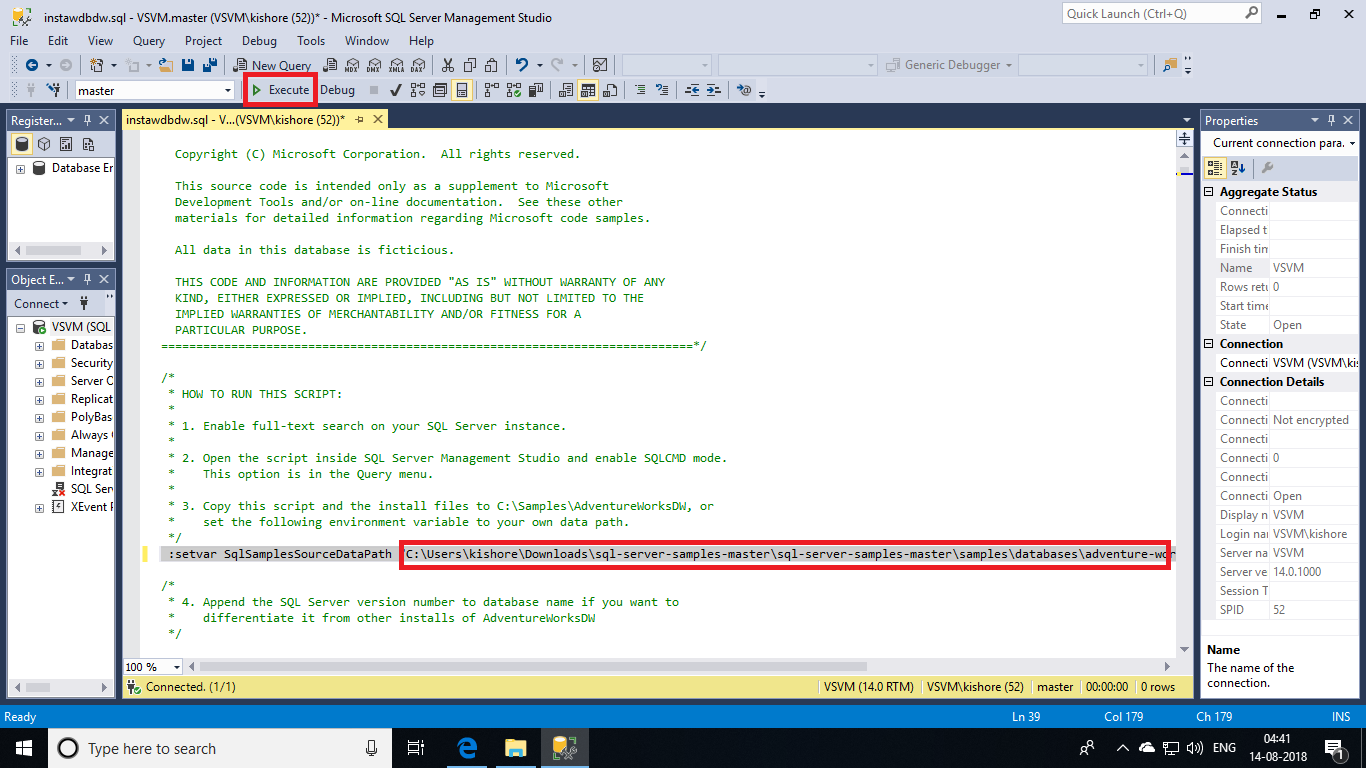
This script will now open in **SQL Server Management Studio**. There connect to your local server by entering the credentials of your SQL Server that you gave while creating the VM.



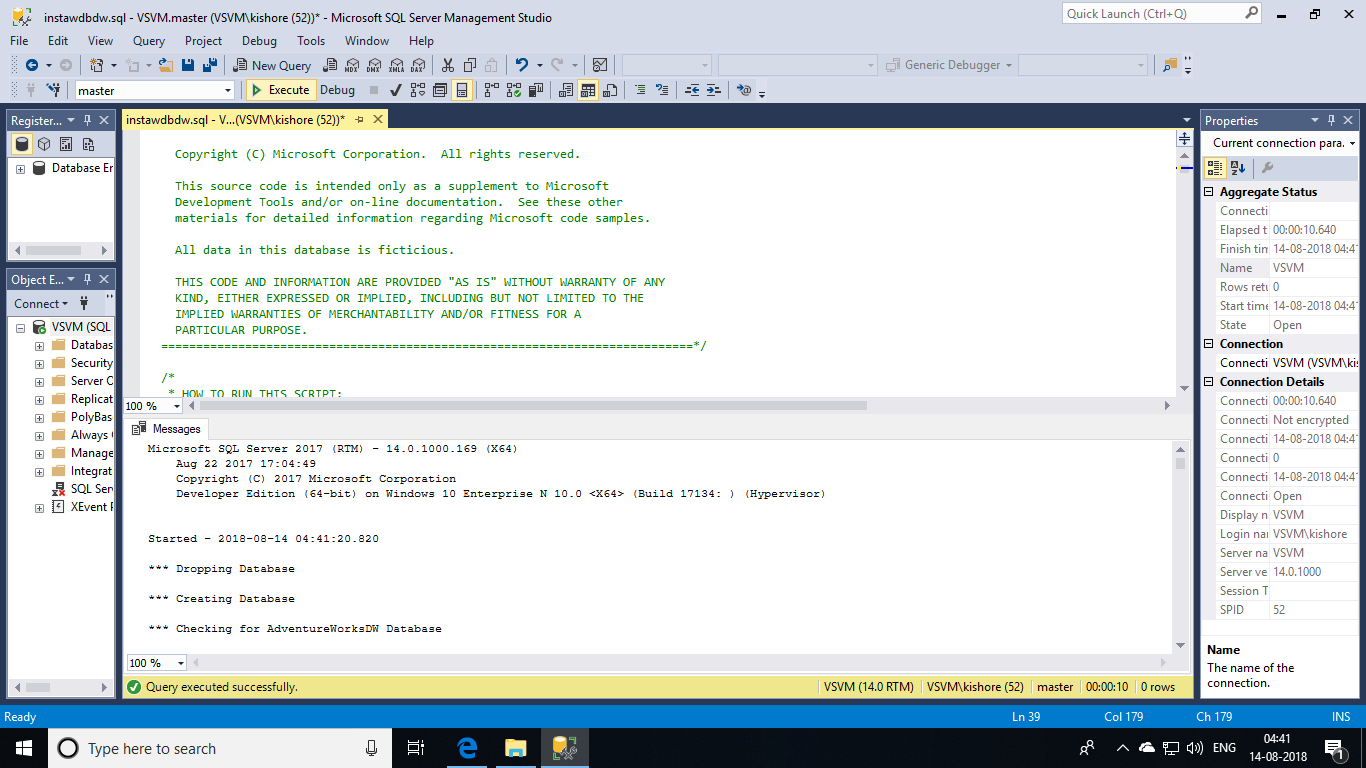
Once after the query opens, click your mouse in the query pane. This will show you Query menu at the top. Click the **Query** menu and choose **SQL CMD Mode** to execute the script.



After this, replace the path of sample data as denoted in the screenshot. Replace the path in between double quotes wit the path of **AdventureWorkLoads** you stored in your machine. Then click on execute button.



This will create you a database with data inside many tables.



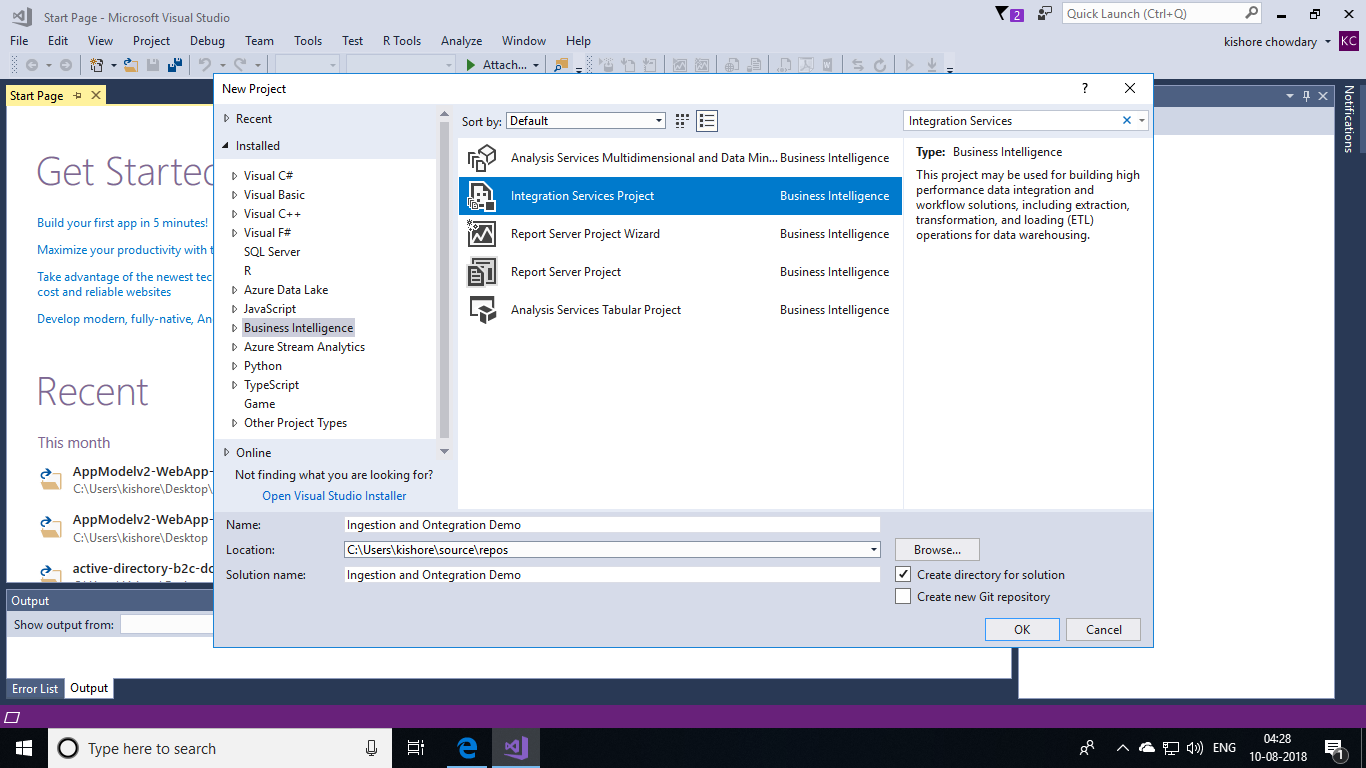
Now we have data in a local server. Let us load this data into Azure SQL Data Warehouse using **Integration Services**.

## **Creating a new Integration Services project:**

## Launch Visual Studio. On the **File** menu, select **New | Project**.

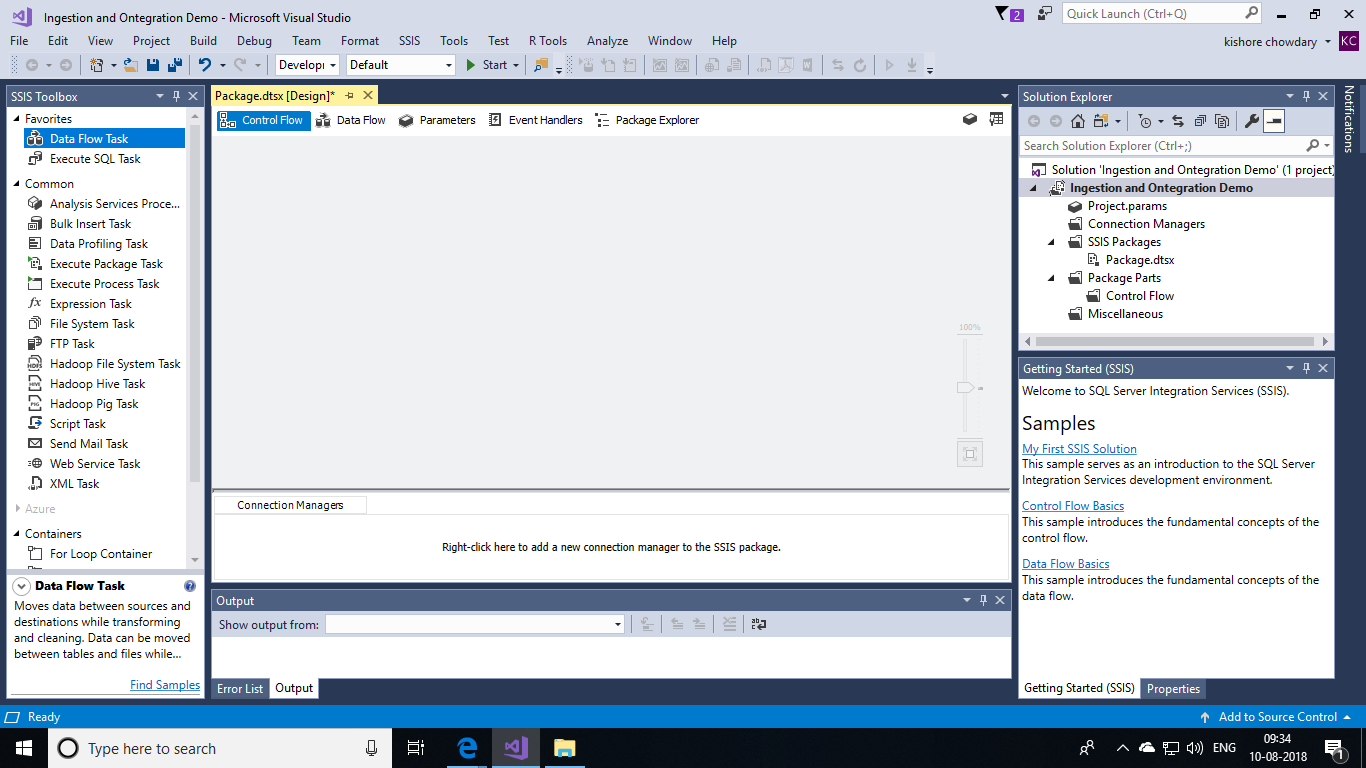


Navigate to the **Installed | Templates | Business Intelligence | Integration Services** project types. Select **Integration Services Project**. Provide values for **Name** and **Location**, and then select **OK**.



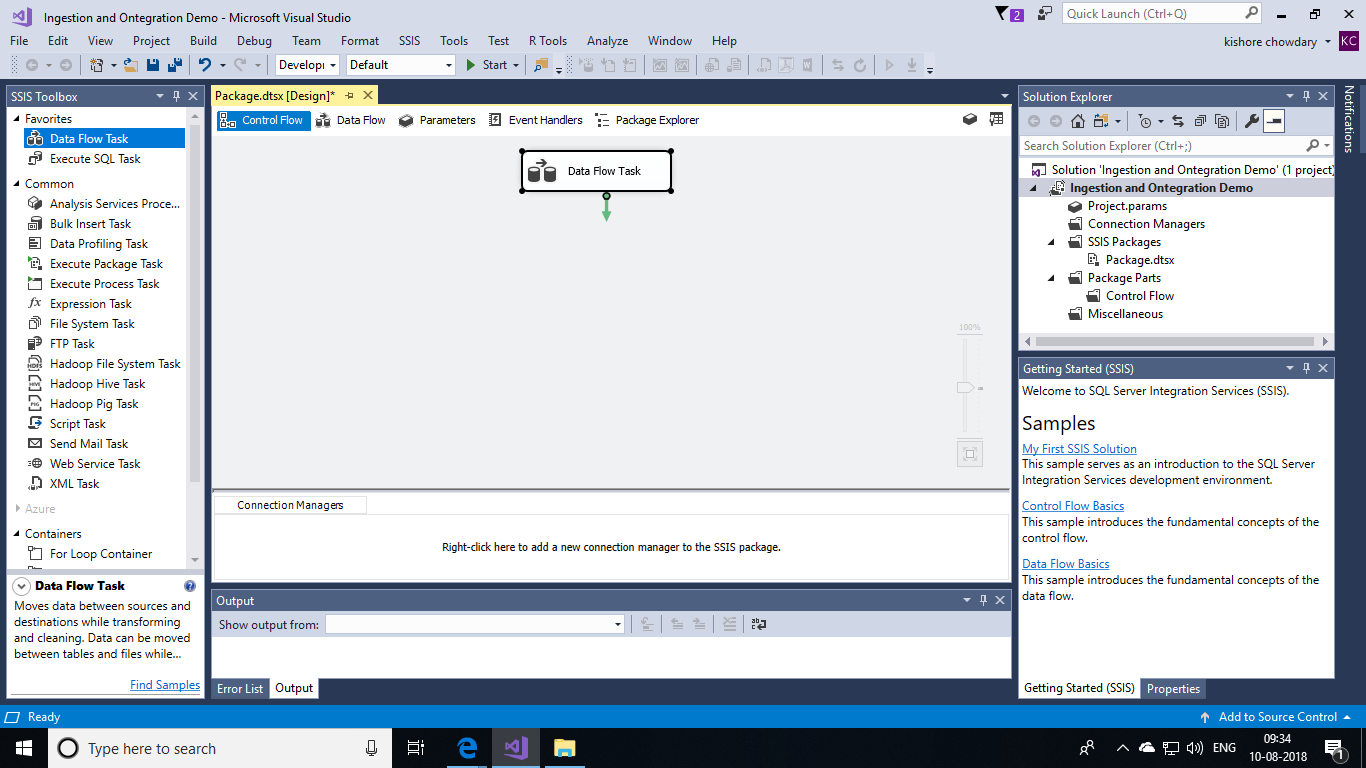
Visual Studio opens and creates a new Integration Services (SSIS) project. Then Visual Studio opens the designer for the single new SSIS package (Package.dtsx) in the project. You see the following screen areas:

* On the left, the **Toolbox** of SSIS components.
* In the middle, the design surface, with multiple tabs. You typically use at least the **Control Flow** and the **Data Flow** tabs.
* On the right, the **Solution Explorer** and the **Properties** panes.

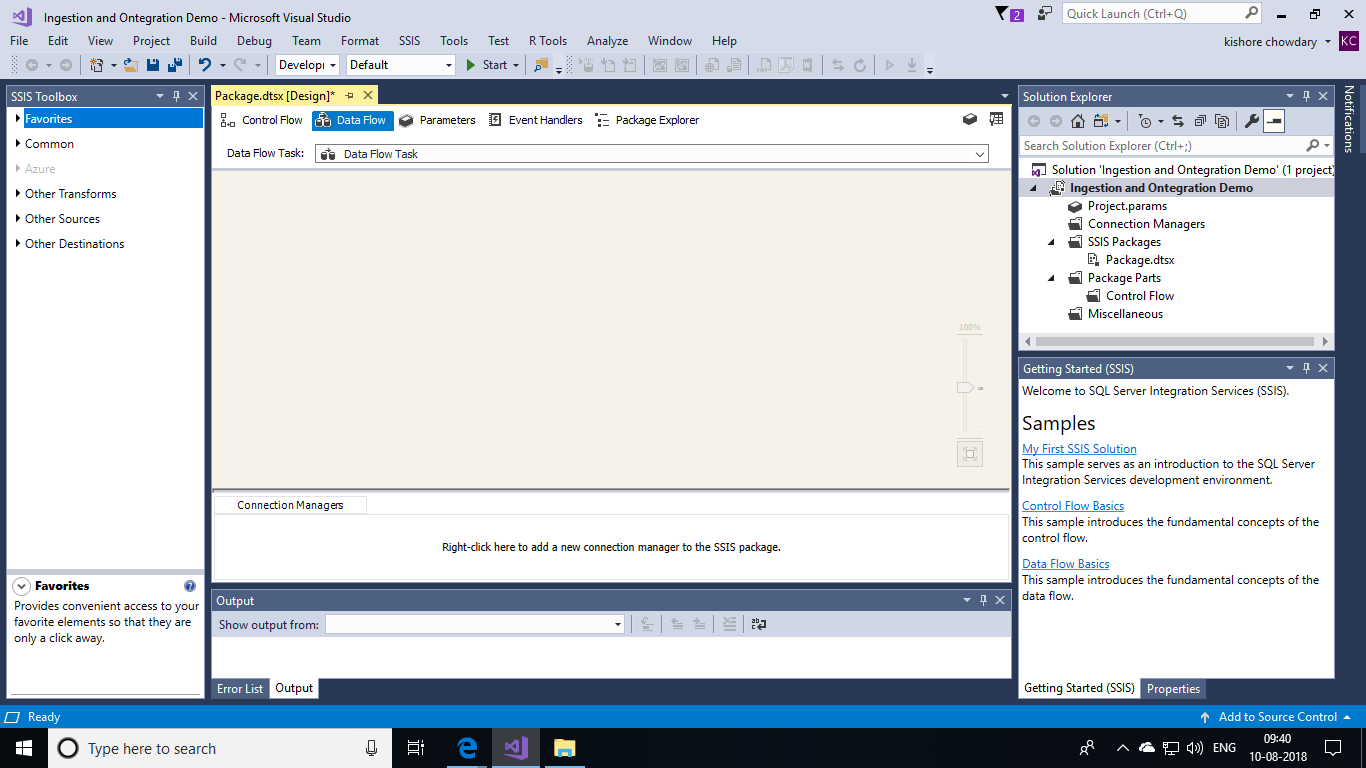


**Creating Data Flow:**

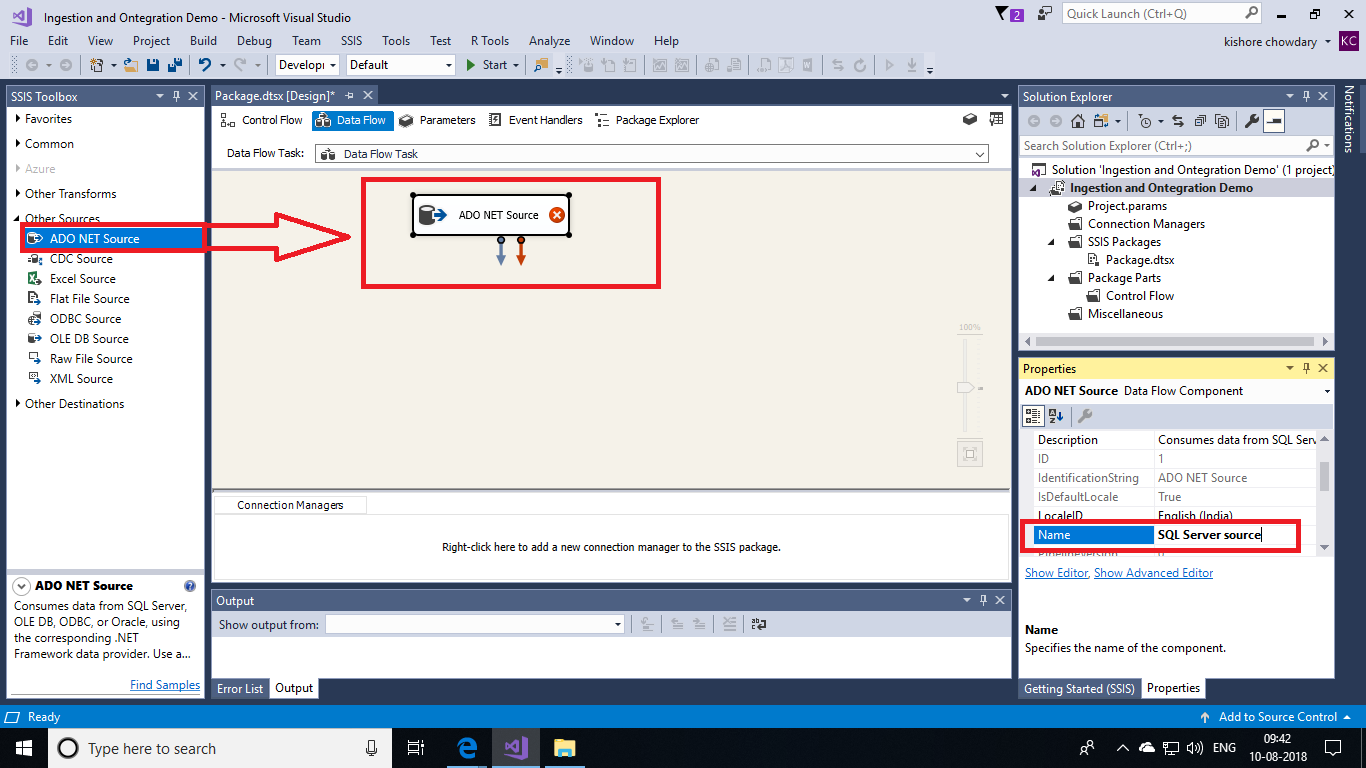
Drag a **Data Flow Task** from the Toolbox to the centre of the design surface (on the **Control Flow** tab).



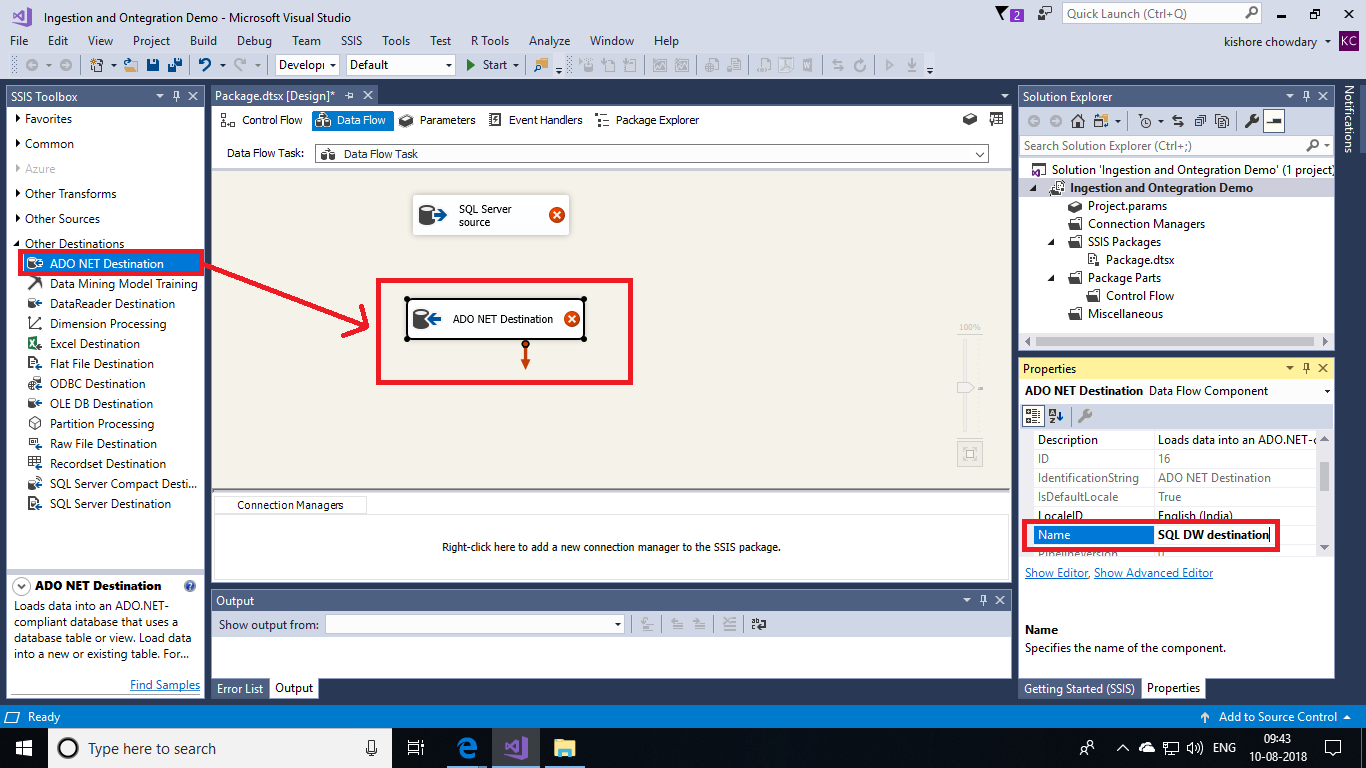
Double-click the Data Flow Task to switch to the Data Flow tab.



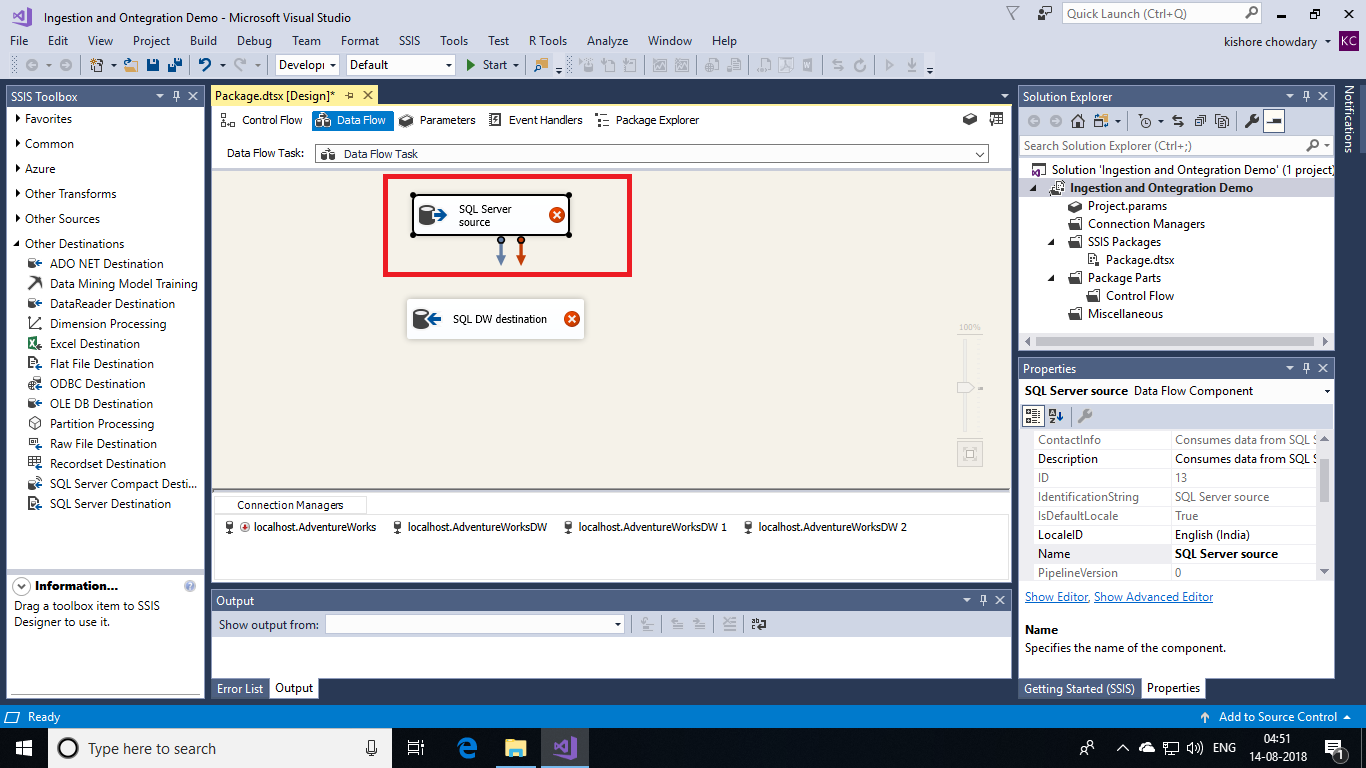
From the Other Sources list in the Toolbox, drag an ADO.NET Source to the design surface. With the source adapter still selected, change its name to **SQL Server source** in the **Properties** pane.



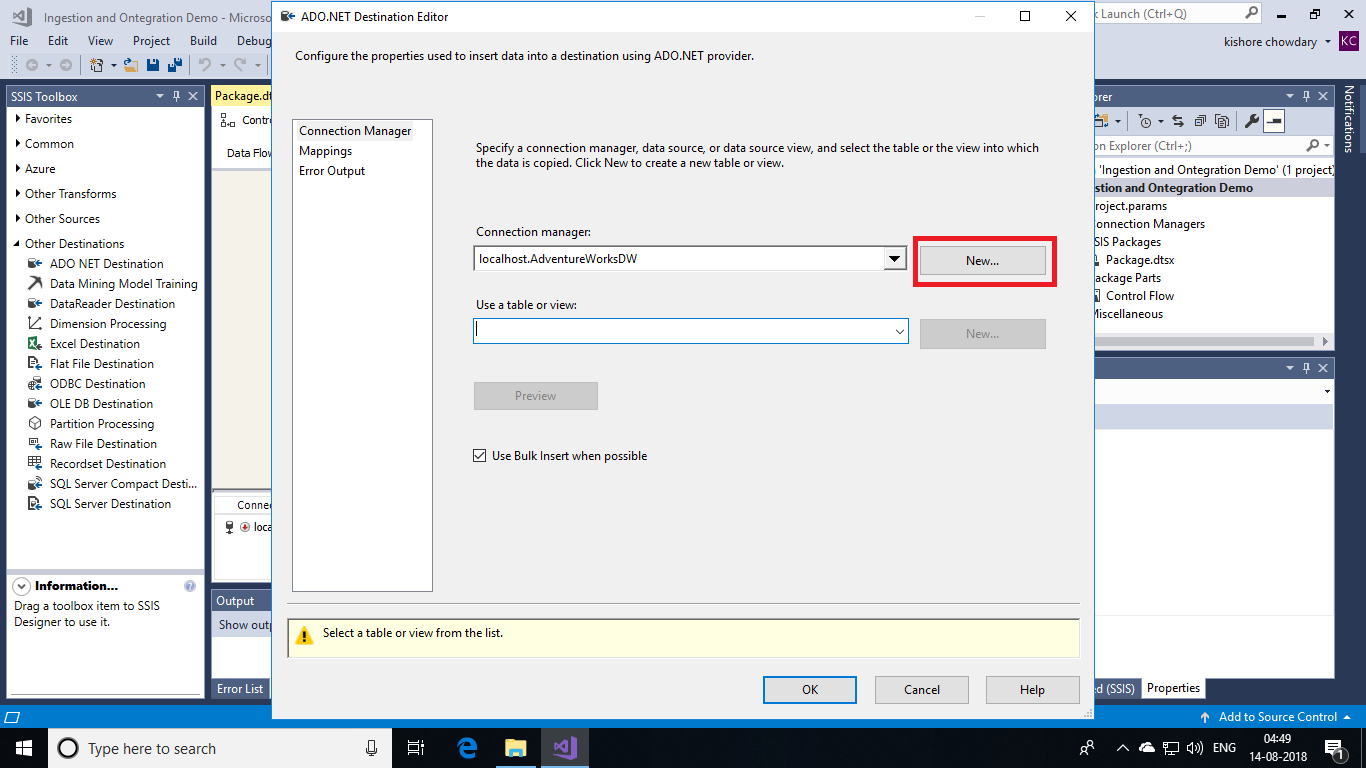
From the Other Destinations list in the Toolbox, drag an ADO.NET Destination to the design surface under the ADO.NET Source. With the destination adapter still selected, change its name to **SQL DW destination** in the **Properties** pane.



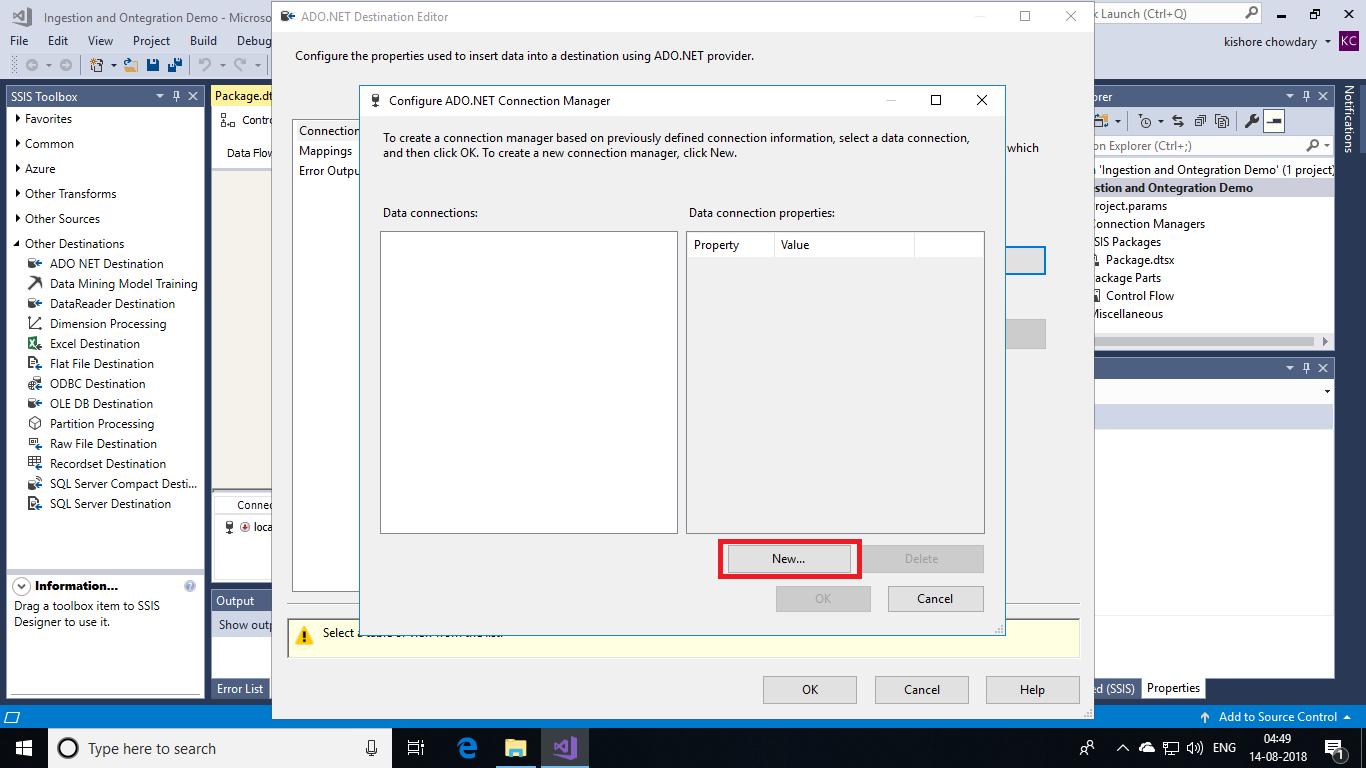
Now, let us setup the source data and destination location to store the data. Double-click the source adapter to open the **ADO.NET Source Editor.**



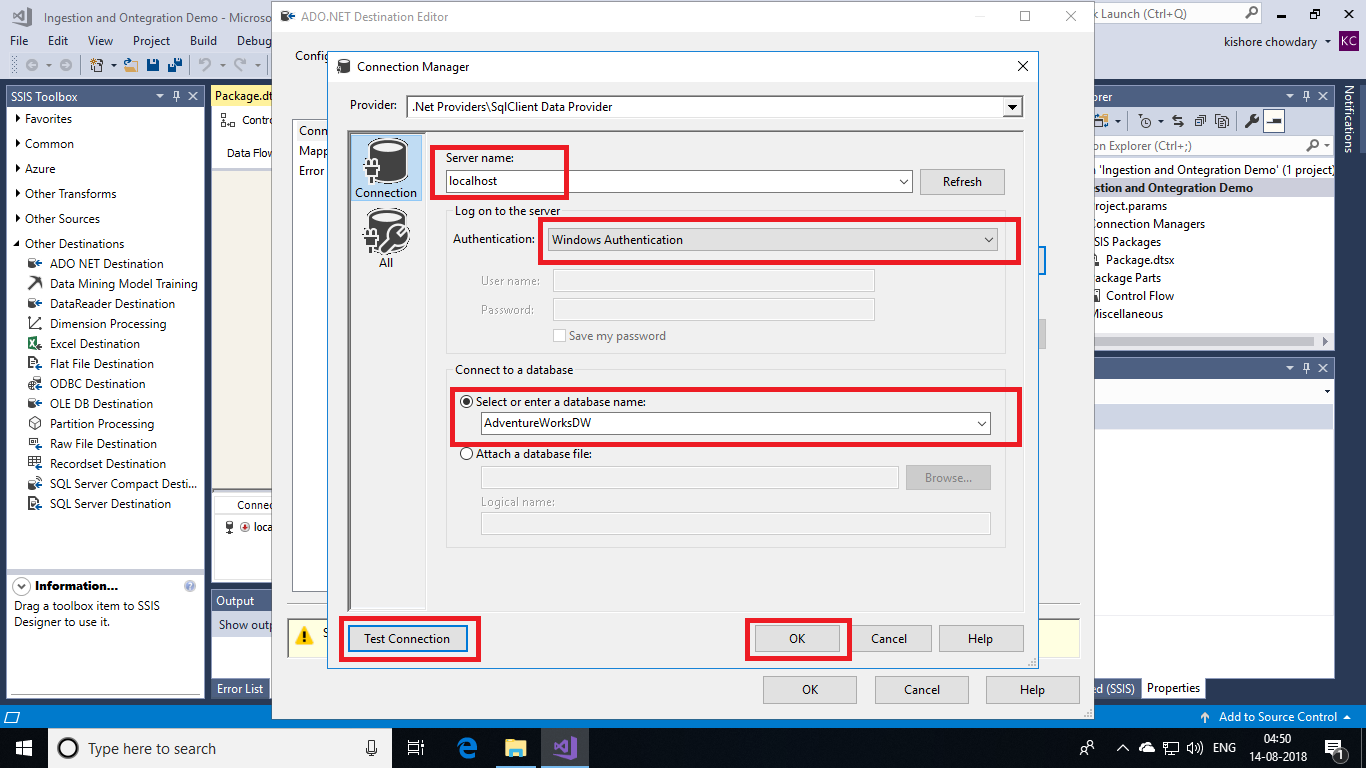
This will open the editor and displays a window to load data into Source. There, click on **New** button to choose a database in the SQL Server.



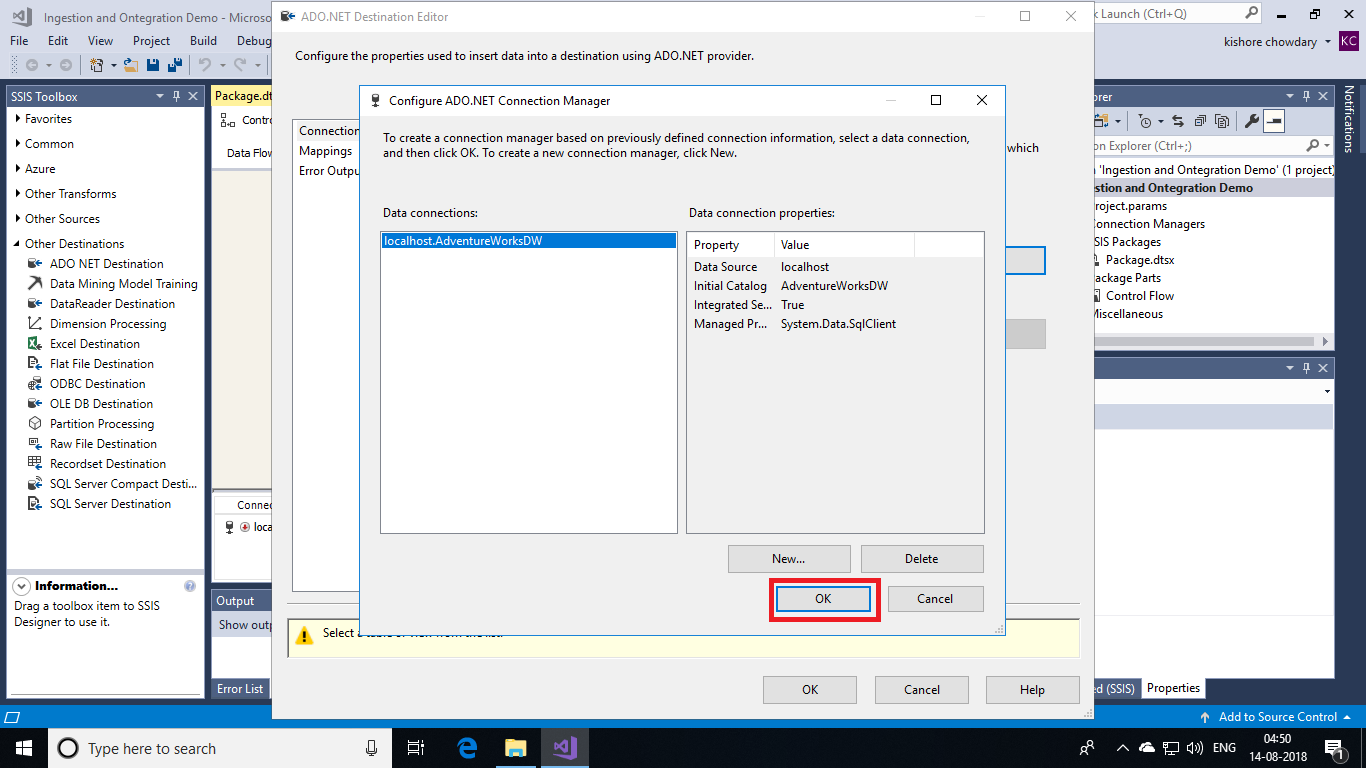
Now, you will get a connection manager. Click **New** button to select the database.



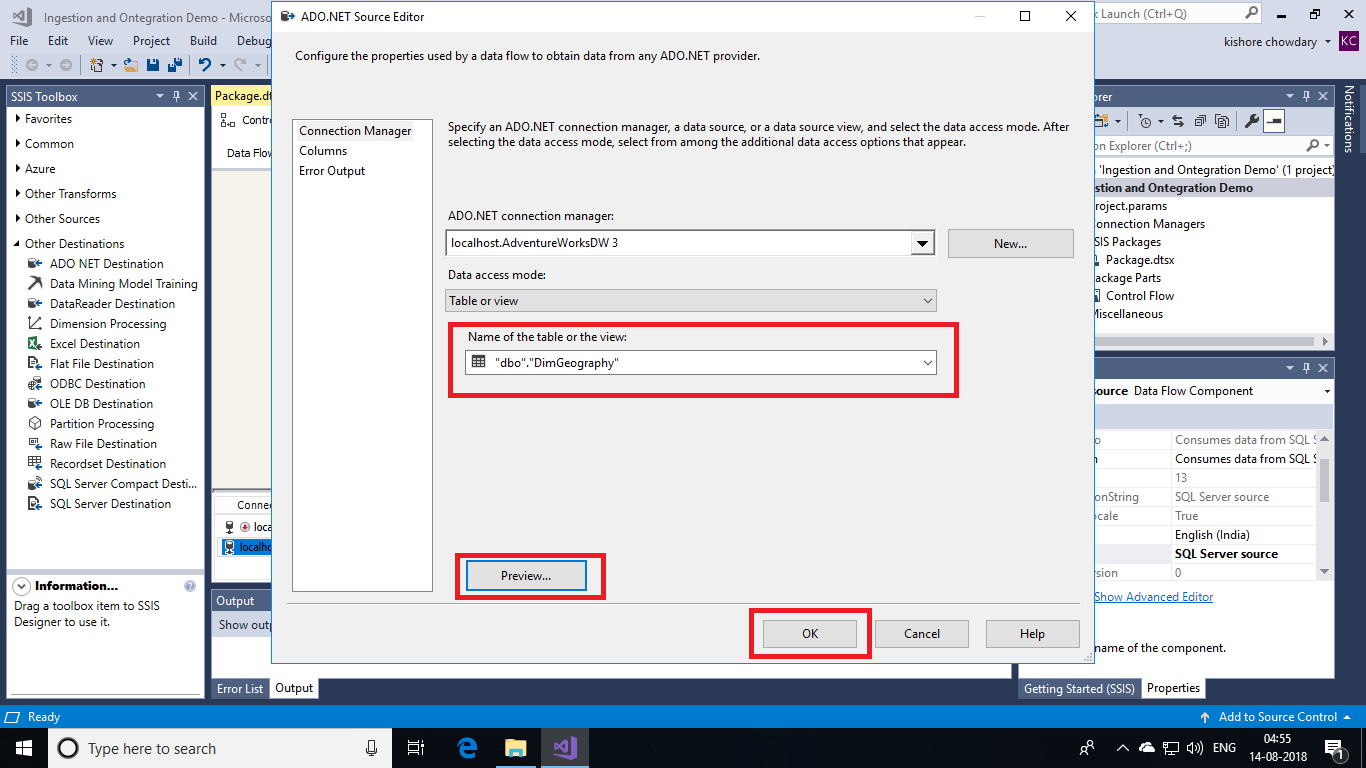
In the place of server name, enter **localhost.** For authentication, choose **Windows Authentication** and then select **AdventureWorksDW**. Test the connection and click **Ok**.



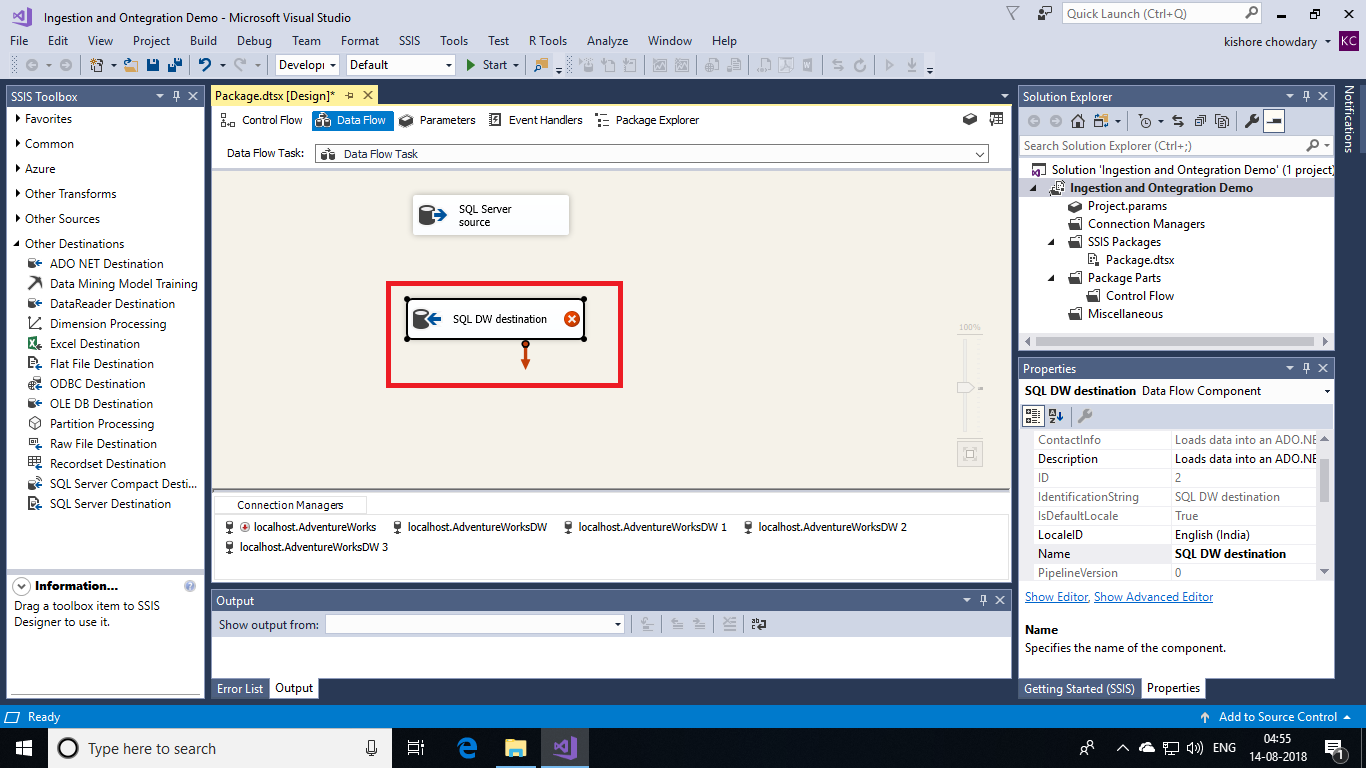
Again click on the Ok button.



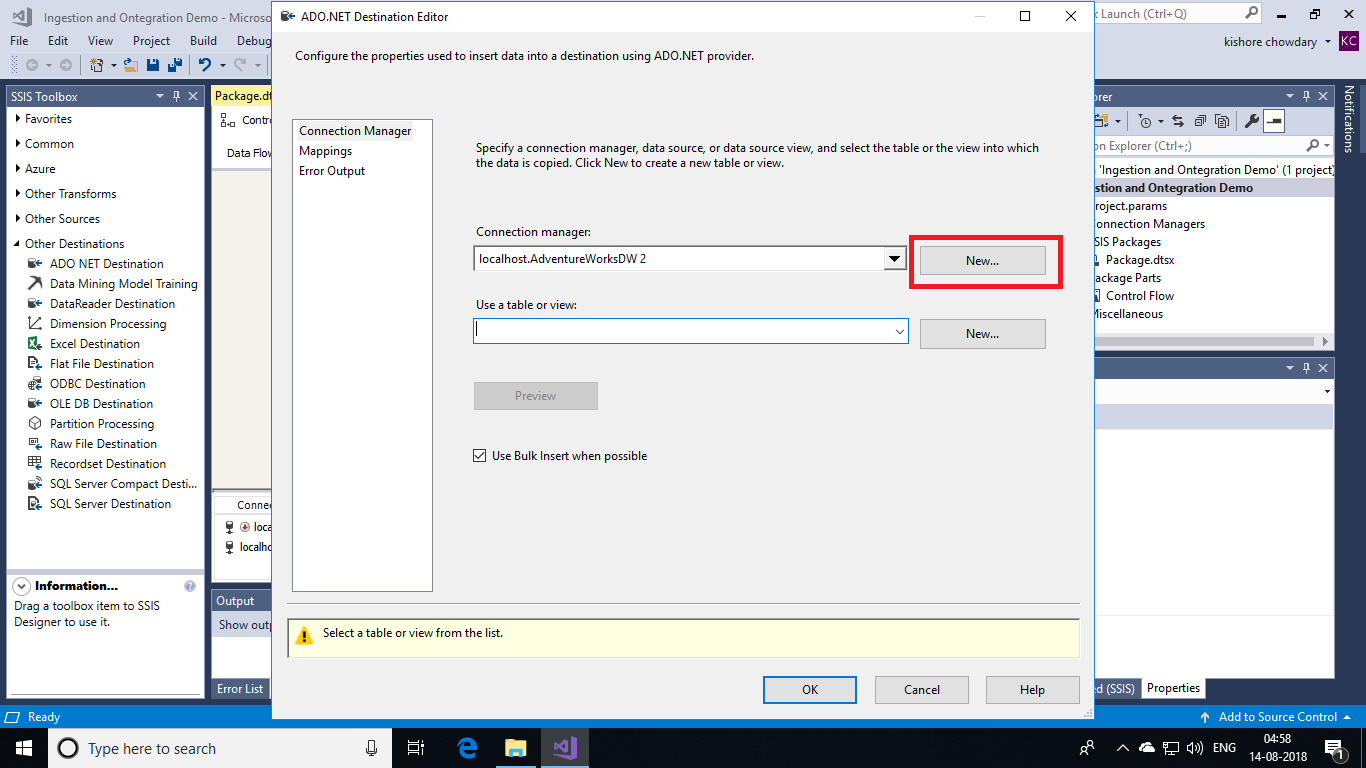
After choosing database, select the table that you wish to load into SQL Data Warehouse. Click on **Preview** button to view the table. Finally, click **Ok** button.



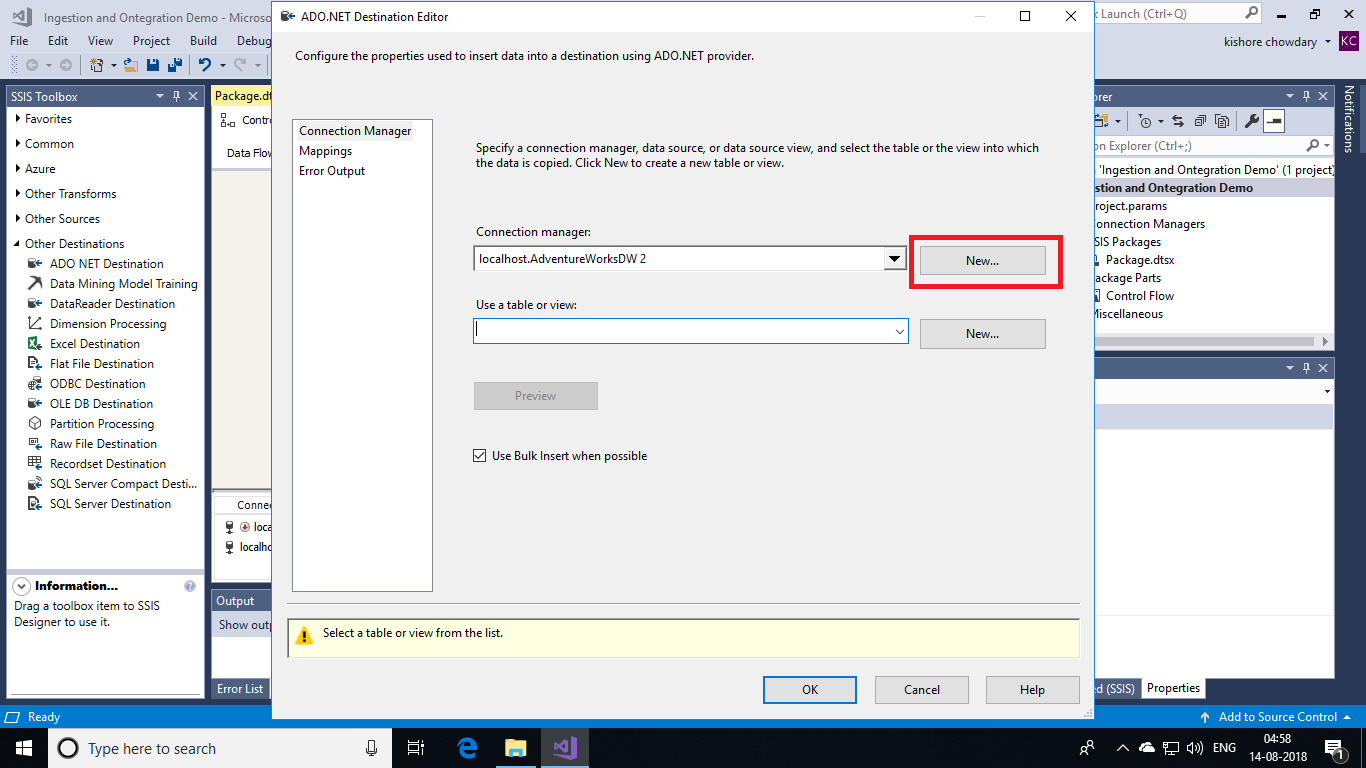
Now, double click the **Destination Adapter** to open the editer.



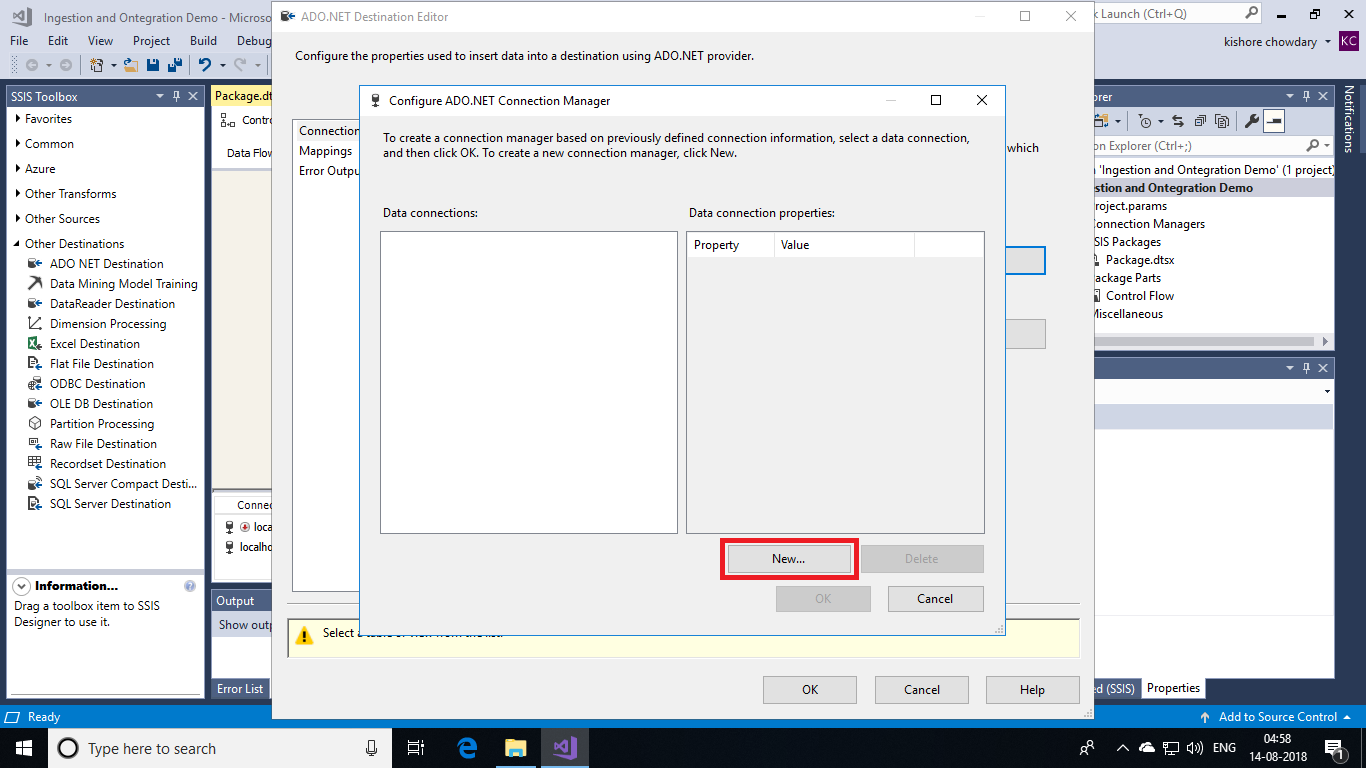
Double-click the destination adapter to open the **ADO.NET Destination Editor**.



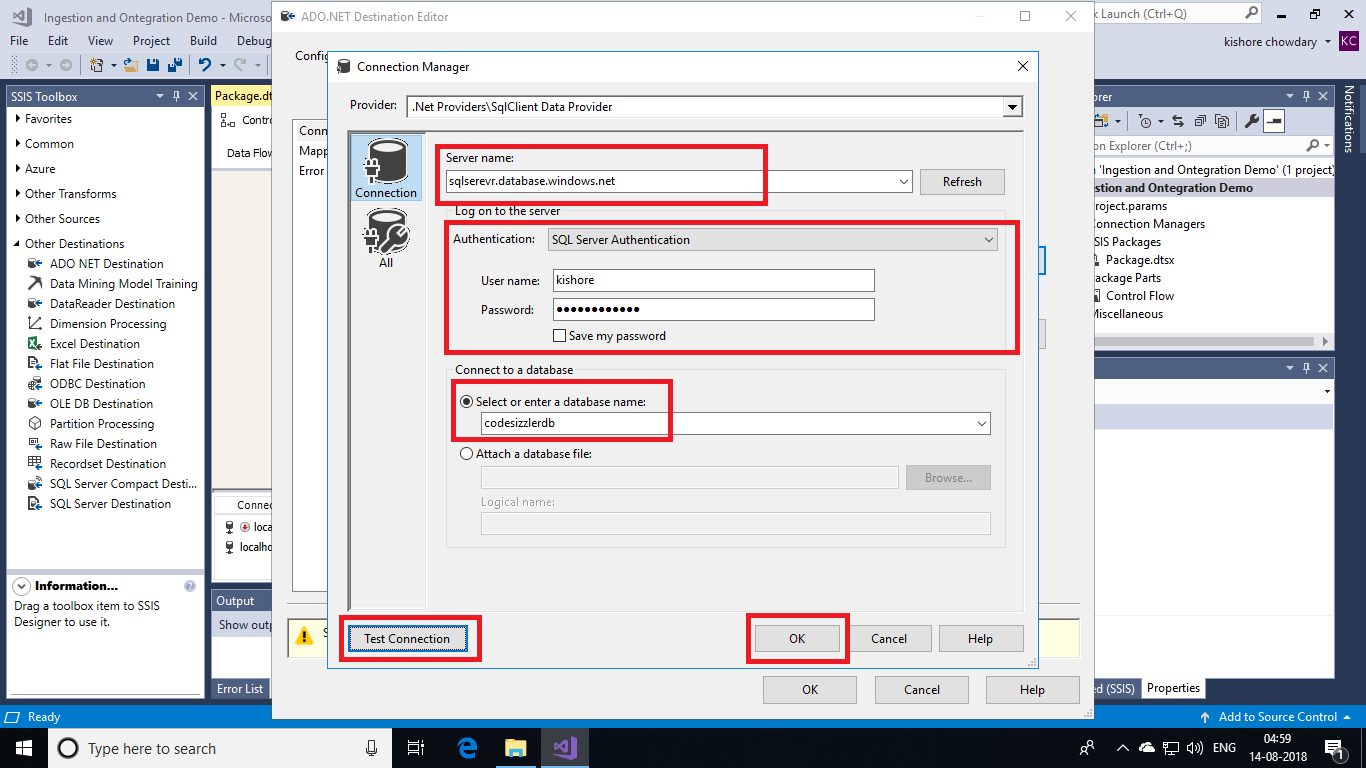
On the **Connection Manager** tab of the **ADO.NET Destination Editor**, click the **New** button next to the **Connection manager** list to open the **Configure ADO.NET Connection Manager** dialog box and create connection settings for the Azure SQL Data Warehouse database into which this tutorial loads data.



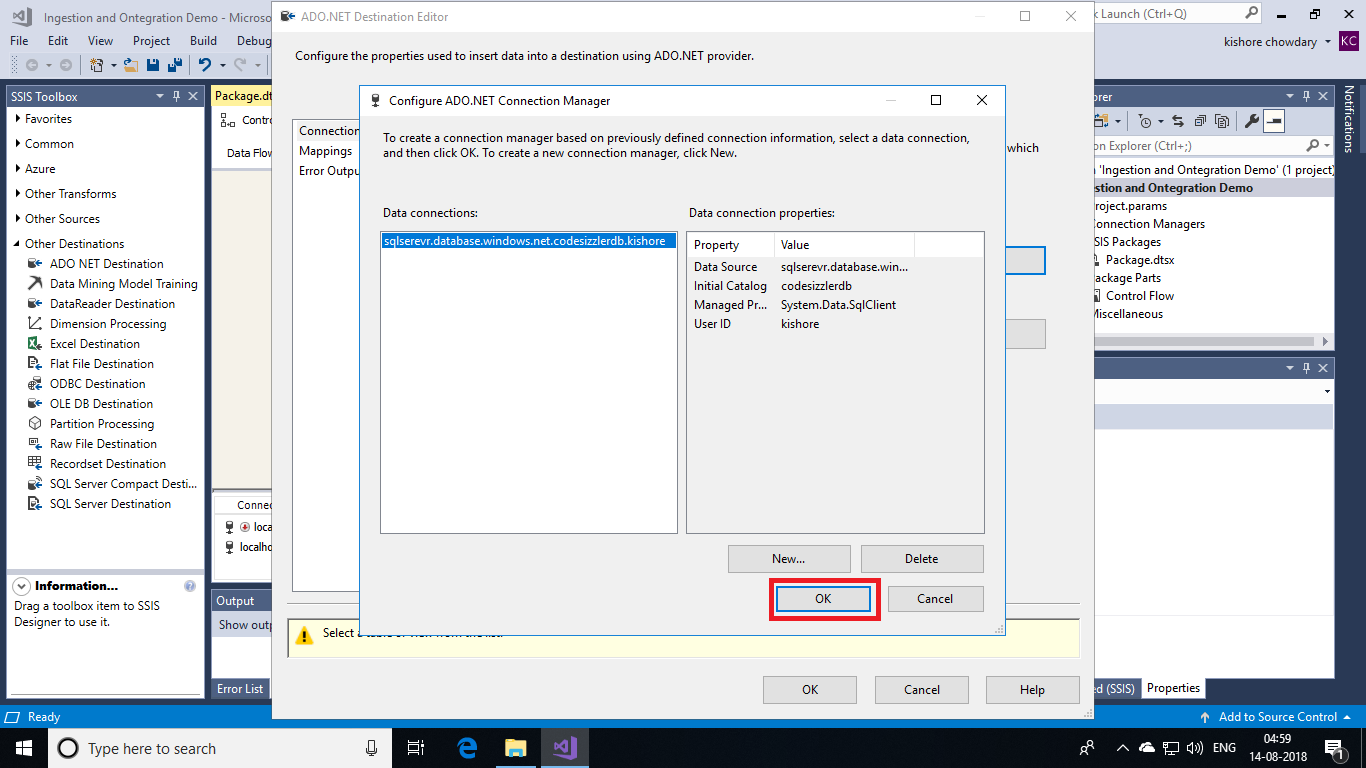
In the **Configure ADO.NET Connection Manager** dialog box, click the **New** button to open the **Connection Manager** dialog box and create a new data connection.



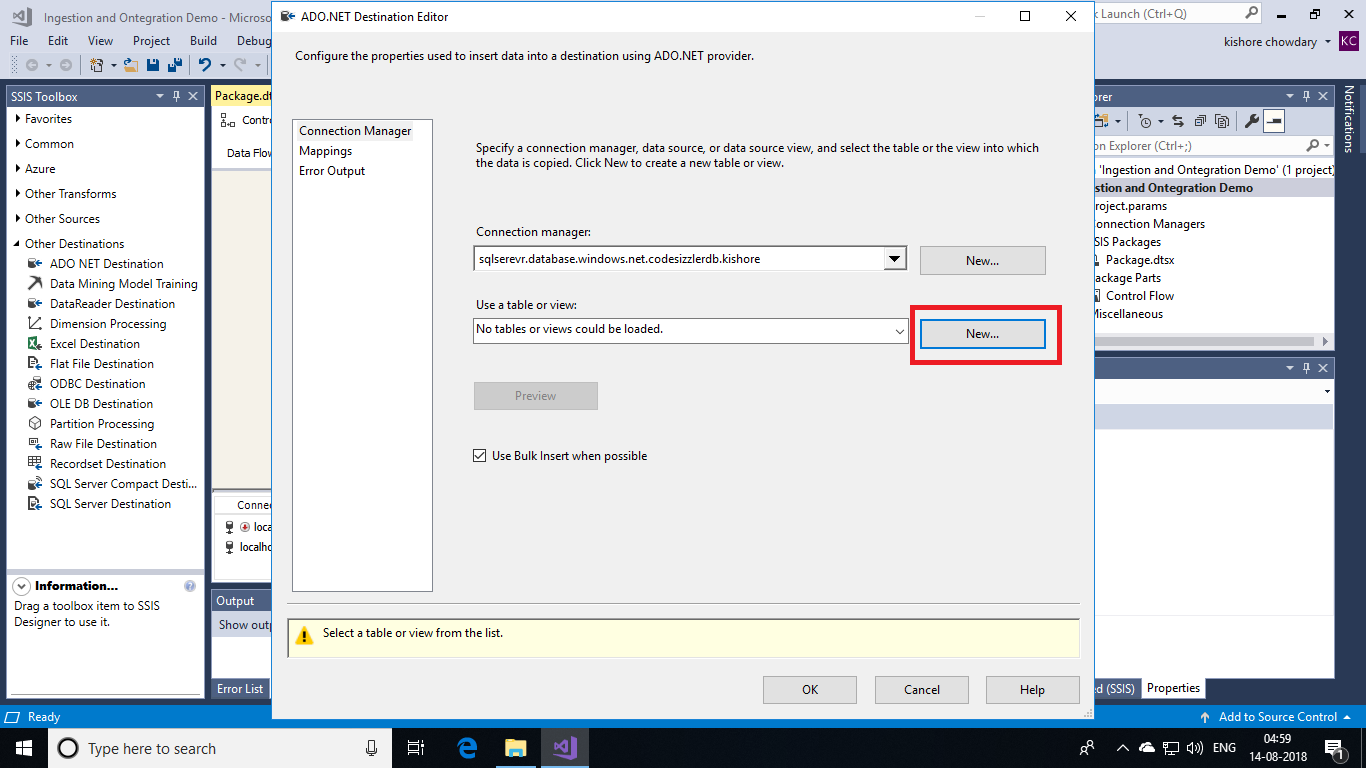
In the place of server name, enter the DNS of your **SQL Server Data Warehouse**. Choose **SQL Server** authentication and enter the user name and password of SQL Server. Choose the database and test your connection. Now, click **Ok** button.



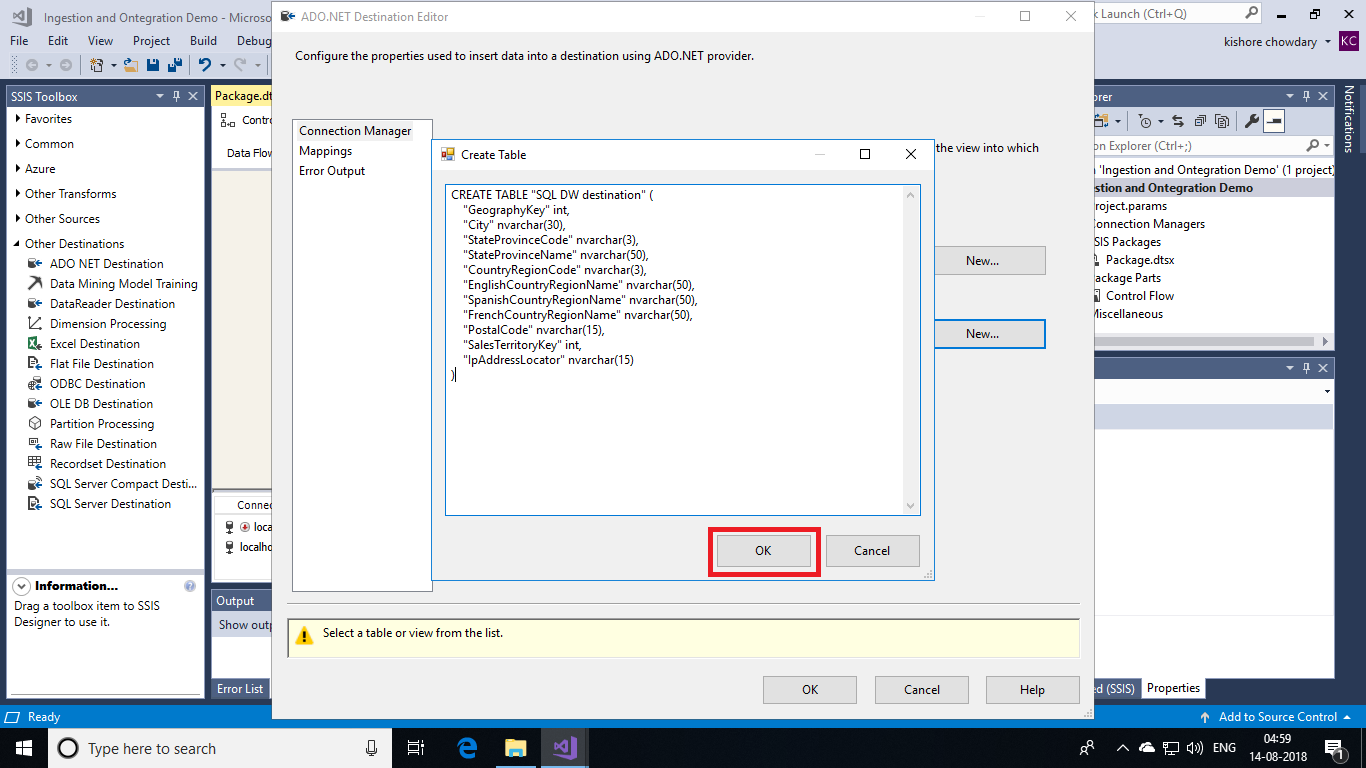
Once again click on **ok** button.



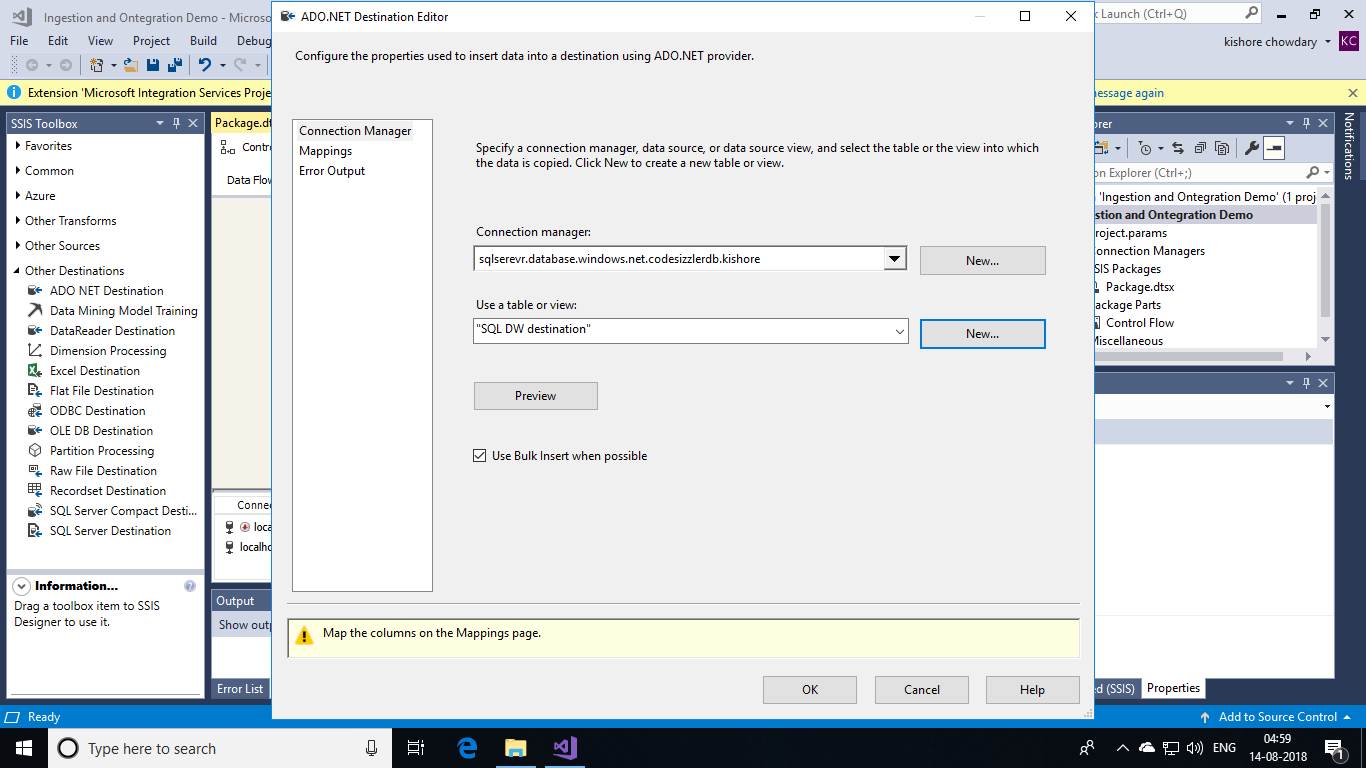
Click on **New** button to create a table with same schema for the purpose of copying data into the SQL Warehouse Database.



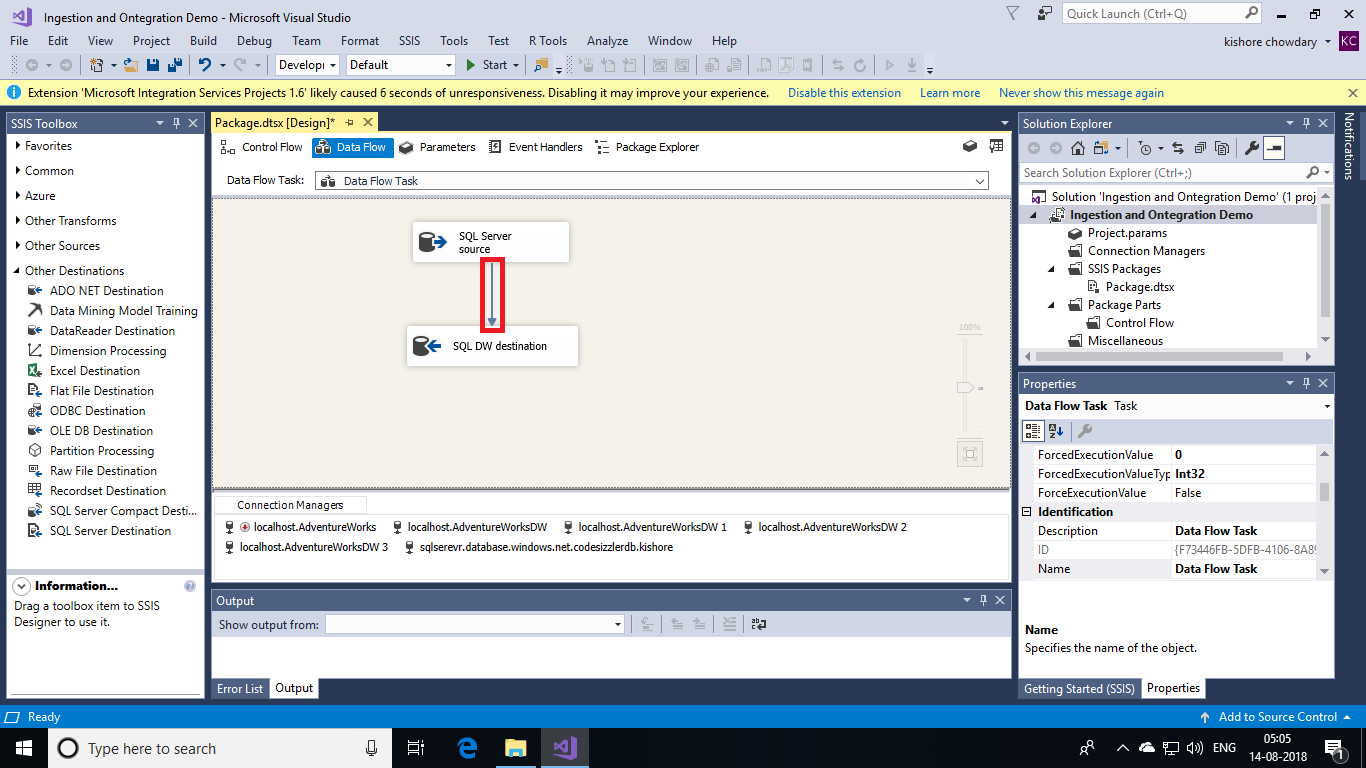
As shown below, a query will be generated. Click on **Ok** button so that table gets created in Azure SQL Data Warehouse.



Next, click the **Mappings** option to map your data with schema. Click **ok** again.



After finishing all these configurations, connect the blue line of source adapter with destination adapter. Then click on start button to execute your script.



On successful deployment, your data in SQL database will be loaded into the SQL Datawarehouse for analytics. You can now use the uploaded data for analytics.

