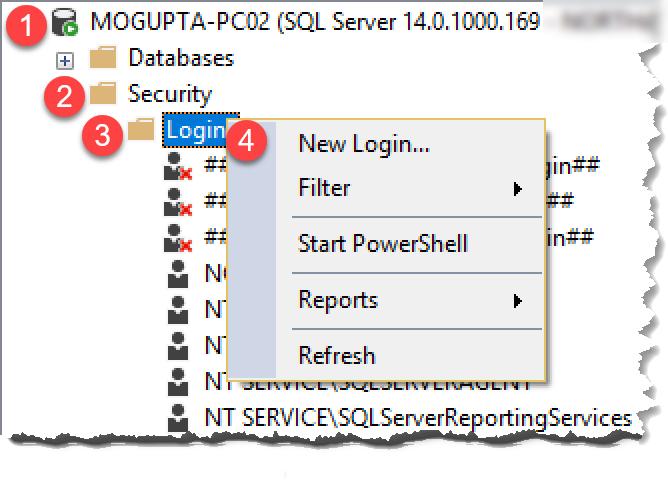
SQL Server Integration Services

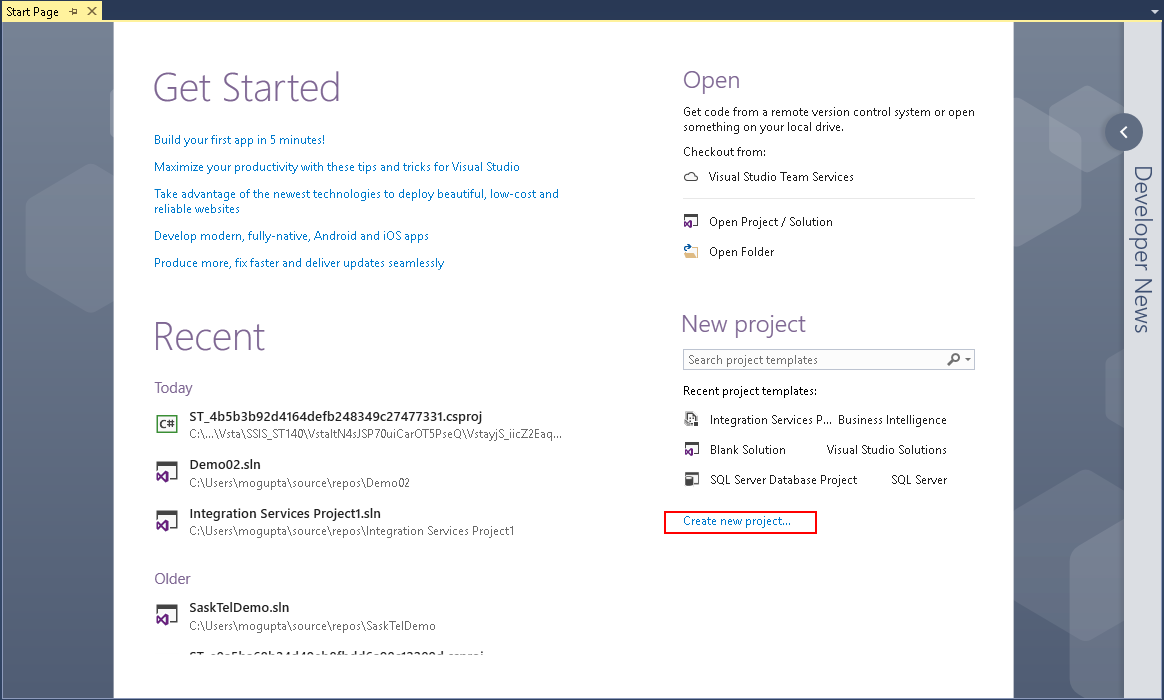
# Module 03: Connection strings

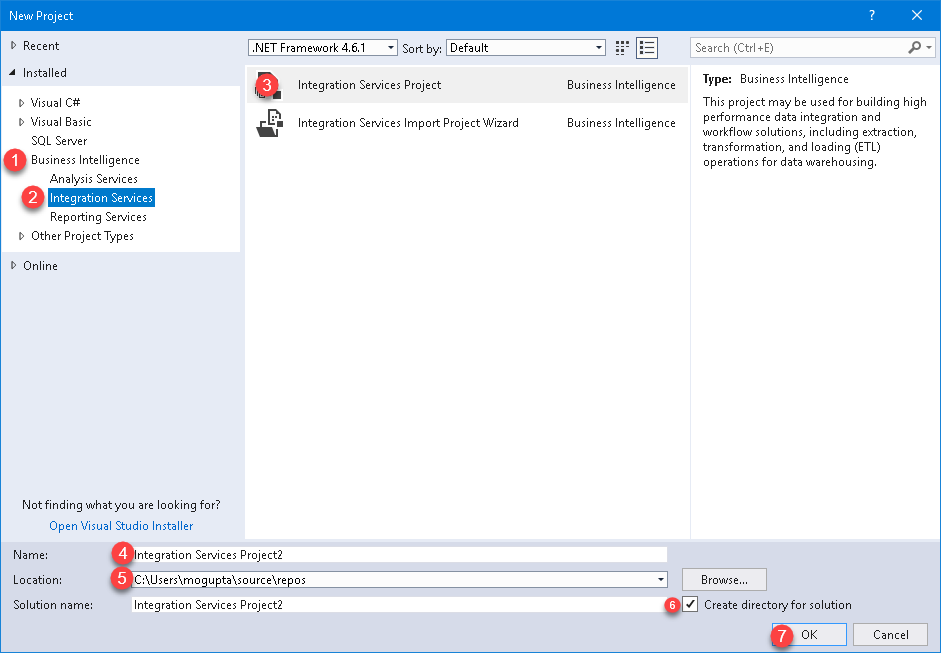
***Note: This lab will be used for future labs, please complete and save it.***

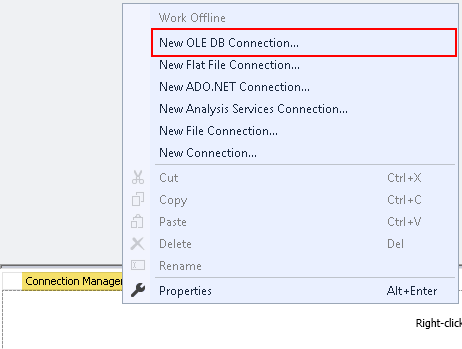
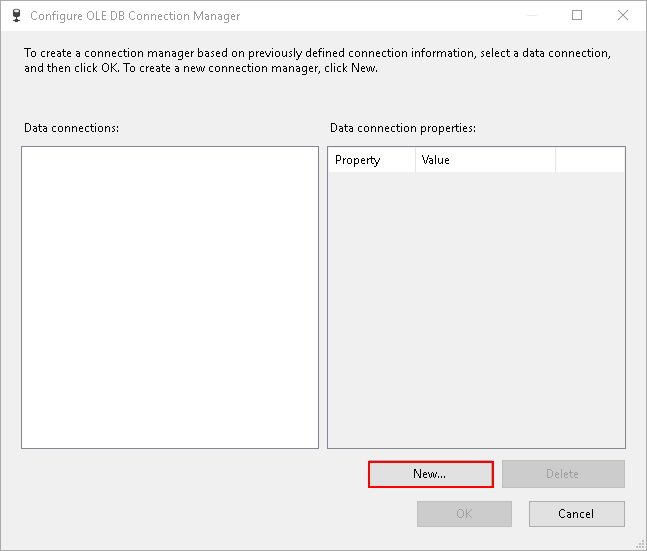
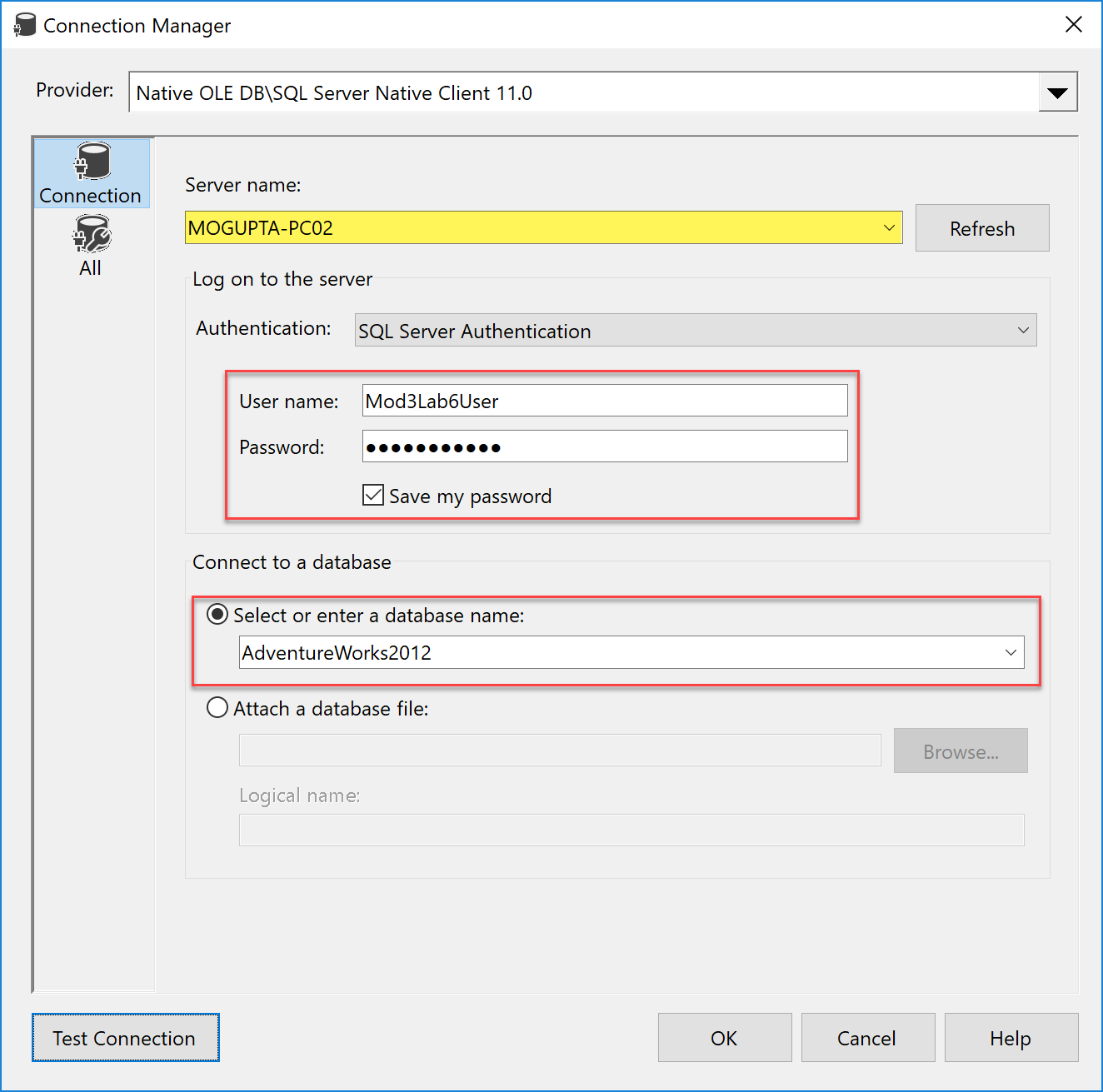
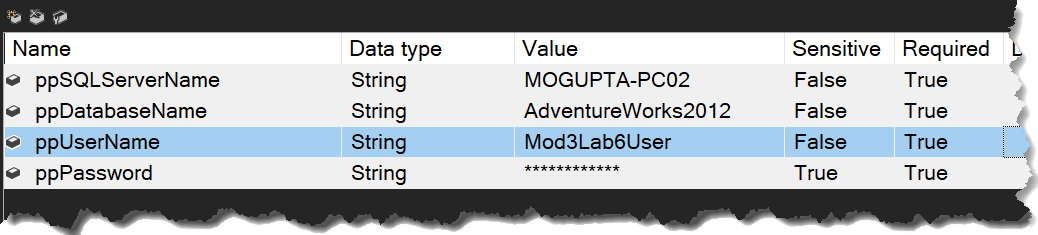
1. Before we start working in SQL Server Data Tools (SDT). Open SQL Server Management Studio (SSMS).
2. Connect to SQL Server Instance.
3. Go to Security > Logins > Right-Click > New Login.  
   
4. In Login – New dialog box select “SQL Server Authentication”. Enter in user name, passwords, and unselect “Enforce” options. Go to Server Roles and grant the user sysadmin. Click OK to create.

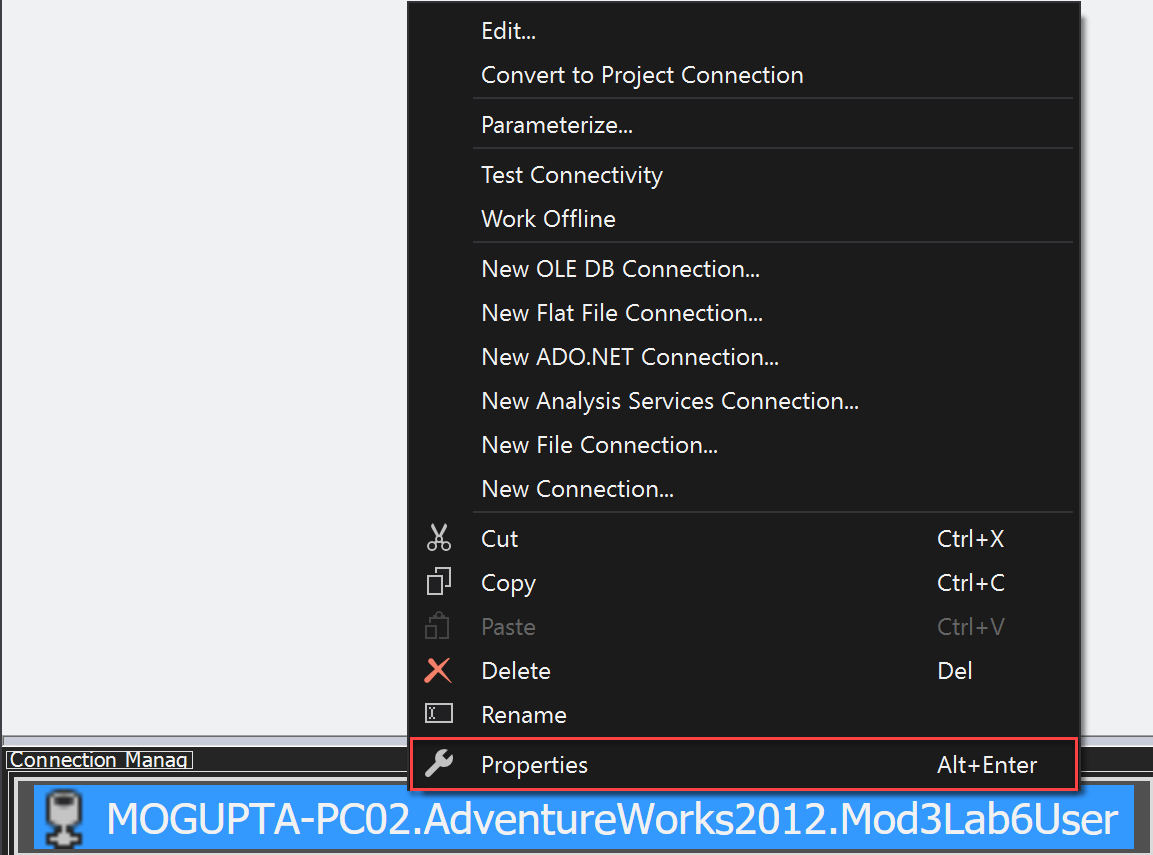
|  |  |
| --- | --- |
|  |  |

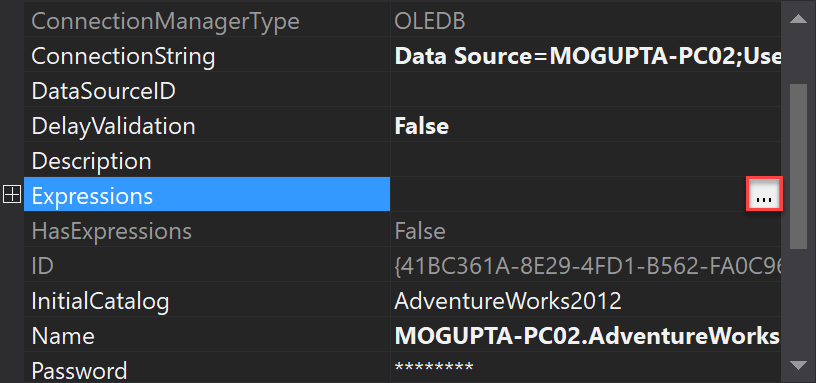
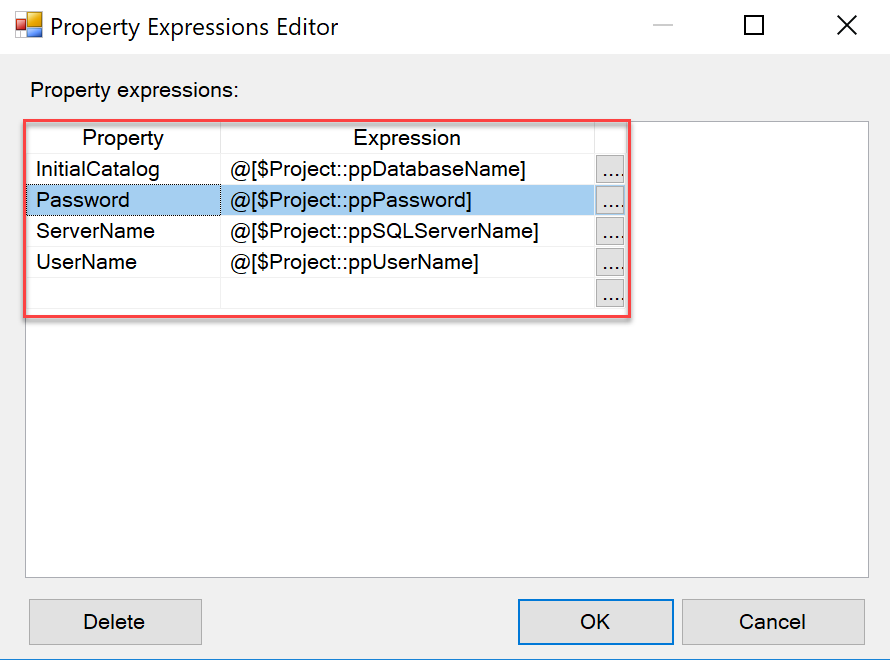
*Note: In production environment, granting sysadmin is not recommended. This is experimentation recommendation only.*

1. Launch SQL Server Data Tools (SSDT), under start menu look for Visual Studio 2017 (SSDT).
2. In the Start Page, click Create new project.  
   
3. In New Project dialog box select Business Intelligence > Integration Services > Integration Services Project. On the bottom enter in project name and location you wish to save the project. Make sure “Create directory for solution” is selected and click OK.



1. Let’s set up a connection manager to our database. In the bottom center pain under Connection Manger, right-click select New Ole-DB Connection.  
   
2. In Configure OLE DB Connect Manager, click New.  
   
3. In connection manager, type the server name, select the database “AdventureWorks2012”, change Authentication type to SQL Server Authentication. Enter in user name and password you created in step #4.  
   
4. Click OK in Configure OLE DB Connection Manager. You should see a single connection under Connection Manager.  
     
   *Hint: Rename the connection manager to OLEDB.ServerName.DatabaseName.SQLAuth. This will make it easier to identify which driver is being used for the driver.*
5. Now let’s parameterize this connection string. First, create new Parameters under project as per the below screenshot.  
     
     
   *Note: Take notice of the sensitive property under ppPassword. We set it to True, this is how we signal to SSISDB Catalog which values we need to protect. Also note it gets converted to “\*\*\*\*\*”.*
6. Right-click on connection string and go to Properties. This will open the properties dialog box in bottom-right if not already open.



1. In properties dialog box, find Expressions. Click on the ellipse beside it to define the expression for connection string.  
   
2. Select ConnectionString and click on the ellipse under expression to build the expression for connection string.  
   
3. Create an Execute SQL Task and attach it to the connection manager. Execute “Select \* FROM Person.Person”.
4. Execute package verify everything works as intended.