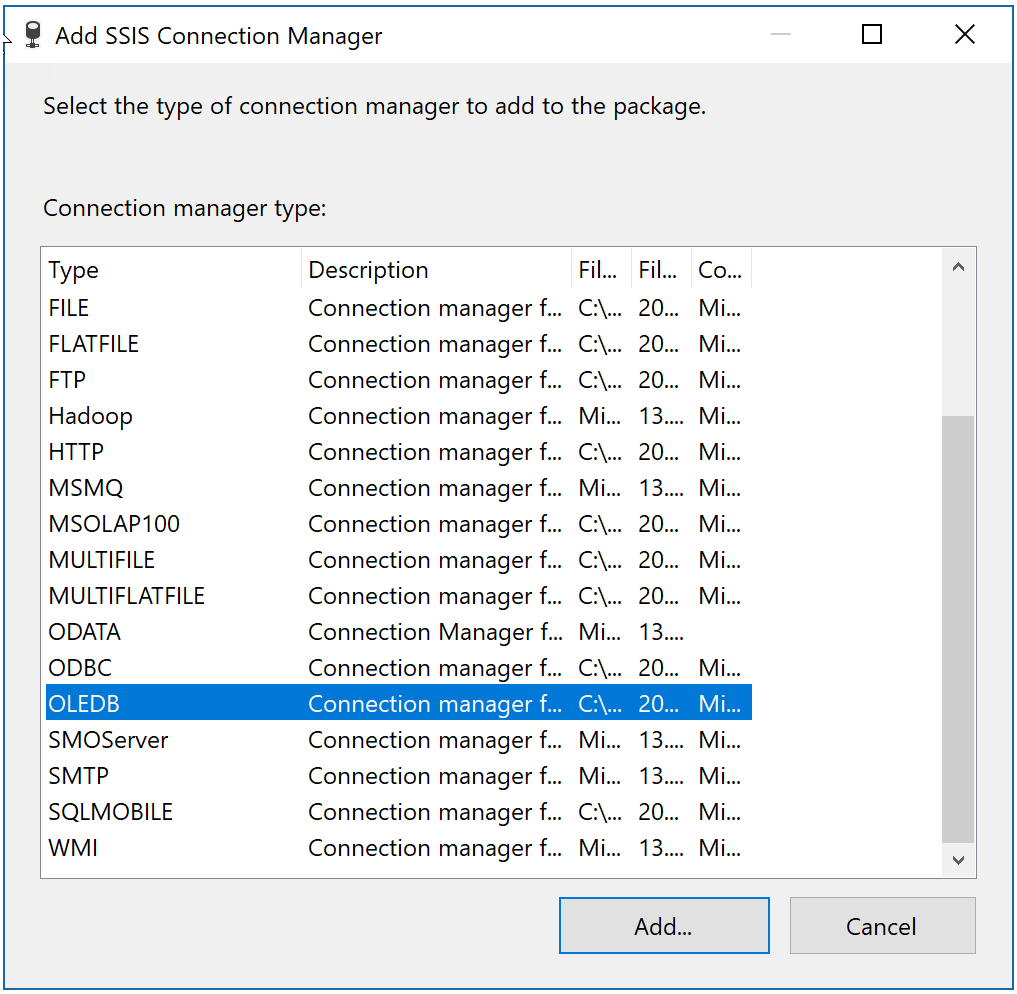
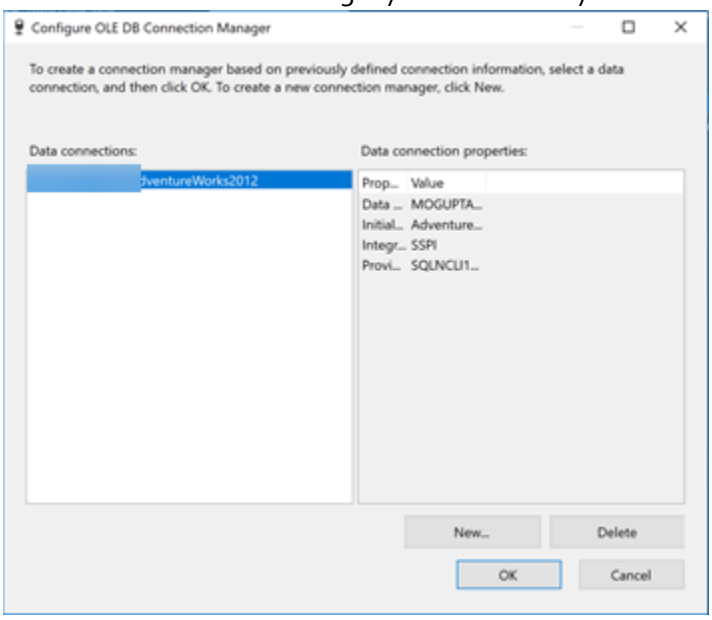
SQL SERVER INTEGRATION SERVICES

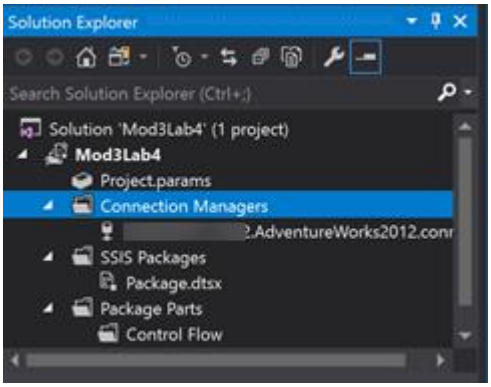
MODULE 03 – LAB 04: CONTROL FLOW: SQL EXECUTE TASK (STORED PROCEDURE)

1. Launch Visual Studio 2019.
2. Create a new Integration Services project.
3. Create a new Shared Connection Manager.
4. Select OLEDB and click Add.



5. Select the connection Manager you have already created in previous Labs. And click OK.

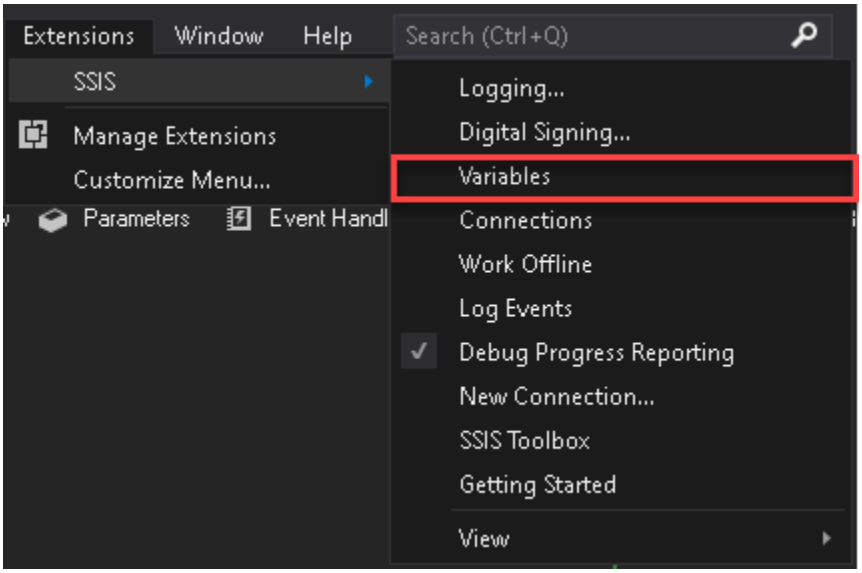




6. In SQL Server Management Studio (SSMS), open and run the StoredProcedure_Sales_uspGetTerritorySalesYTD.sql script to create a simple stored procedure that'll be need in later steps (make sure to update the database name).

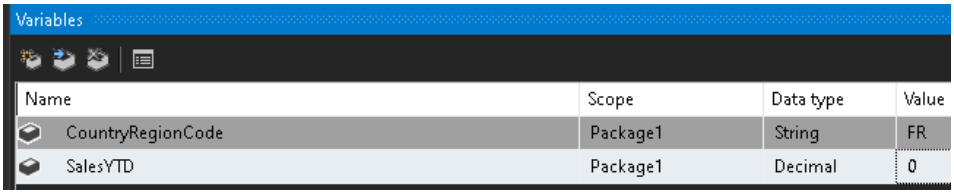
```
USE AdventureWorks2012 --You may need to change D
GO

IF OBJECT_ID('sales.uspGetTerritorySalesYTD') IS
    DROP PROCEDURE Sales.uspGetTerritorySalesYTD;
GO

CREATE PROCEDURE Sales.uspGetTerritorySalesYTD
  @CountryRegionCode VARCHAR(50)
, @SalesYTD REAL OUTPUT
```

7. Verify script runs correctly by testing it. Open and run the StoredProcedure_Sales_uspGetTerritorySalesYTD_Test.sql script (make sure to update the database name).
8. Next, using the Variables to call the stored procedure with input and output parameters. Capture the numbered result set is returned by the stored procedure and then display the result.
9. In the SSIS Menu select Variables.

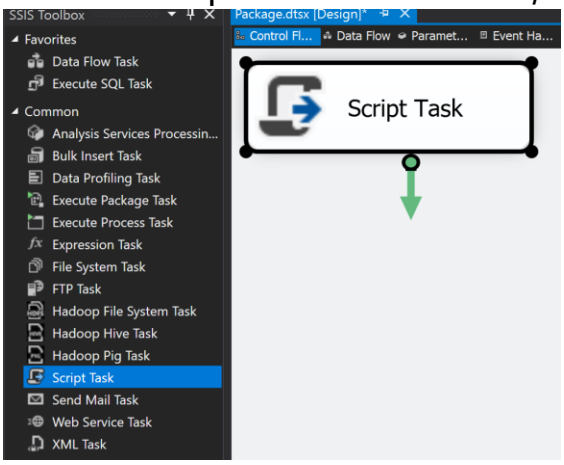


10. Click on the New Variable icon.

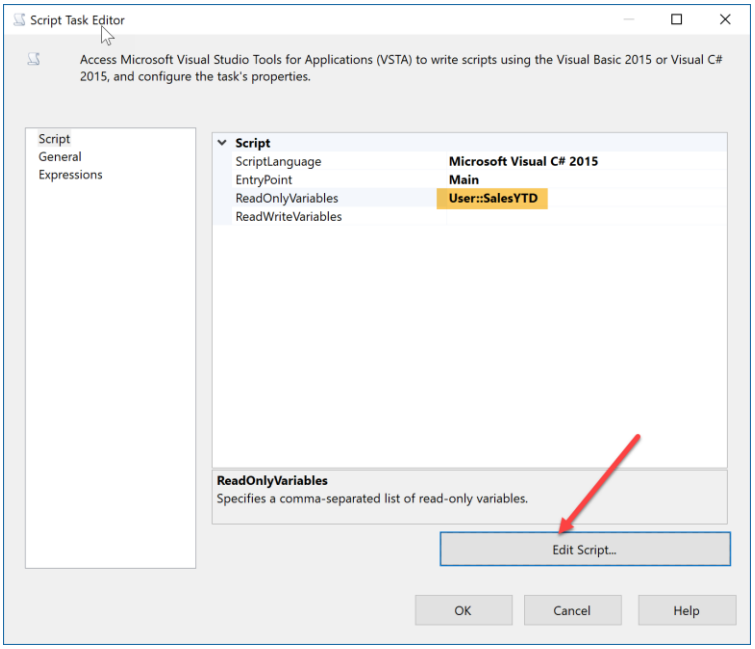
11. Rename the Variable to CountryRegionCode of String Data Type, with a default value of FR. Rename second variable to SalesYTD of Decimal data type.



12. Double-click **Script Task** from SSIS Toolbox, to add onto the Control Flow canvas.



13. Set Script Task's ReadOnlyVariables property to User::SalesYTD in the Script Task Editor, then click on Edit Script.



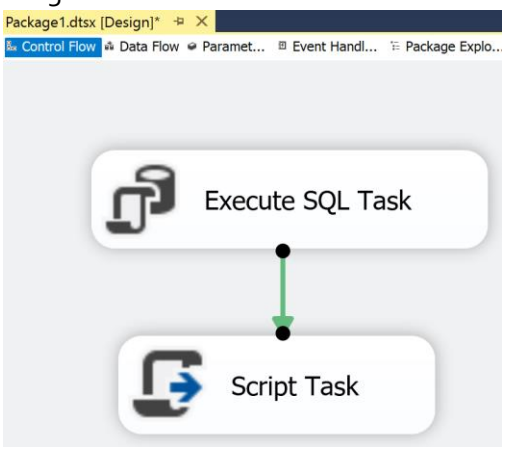
14. In the VstaProjects window, Add following code line into the main() block.

```
MessageBox.Show(Dts.Variables["User::SalesYTD"].Value.ToString());
```

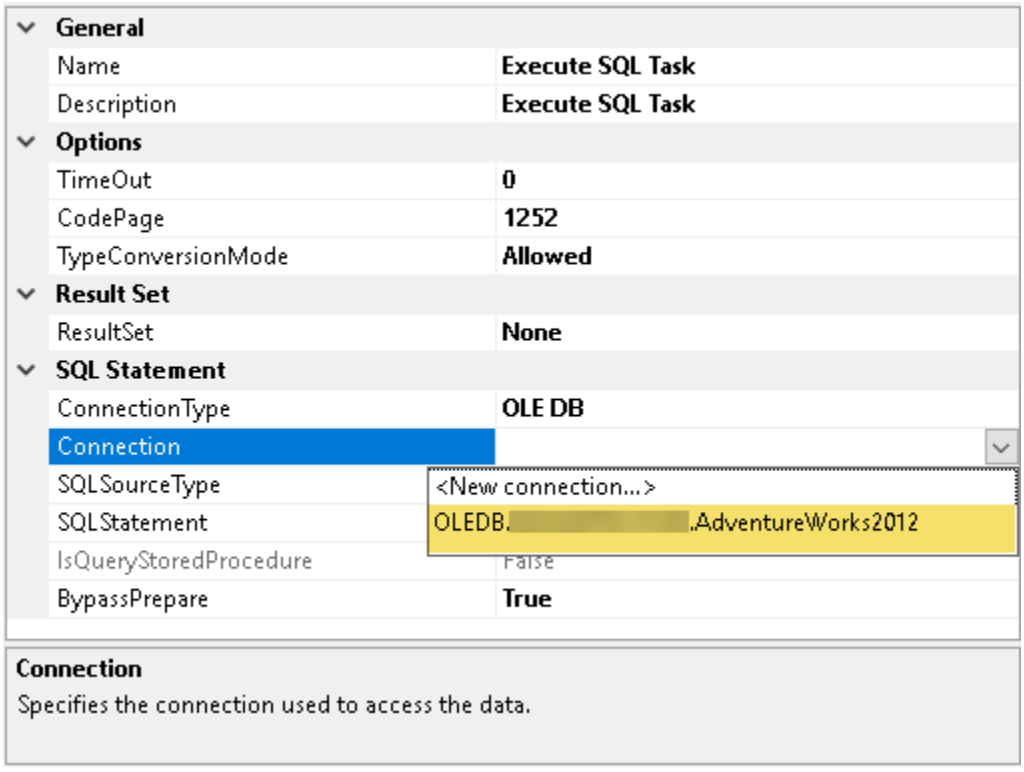
```
public void Main()
{
    // TODO: Add your code here

    Dts.TaskResult = (int)ScriptResults.Success;
    MessageBox.Show(Dts.Variables["User::SalesYTD"].Value.ToString());
}
```

15. Close the VstaProjects (it will automatically save the code), and then click on OK to close the Script Task Editor Dialog.
16. Double-click on **Execute SQL Task** from SSIS toolbox, to add onto Control Flow canvas.

17. Drag Green arrow from Execute SQL Task to Script Task.

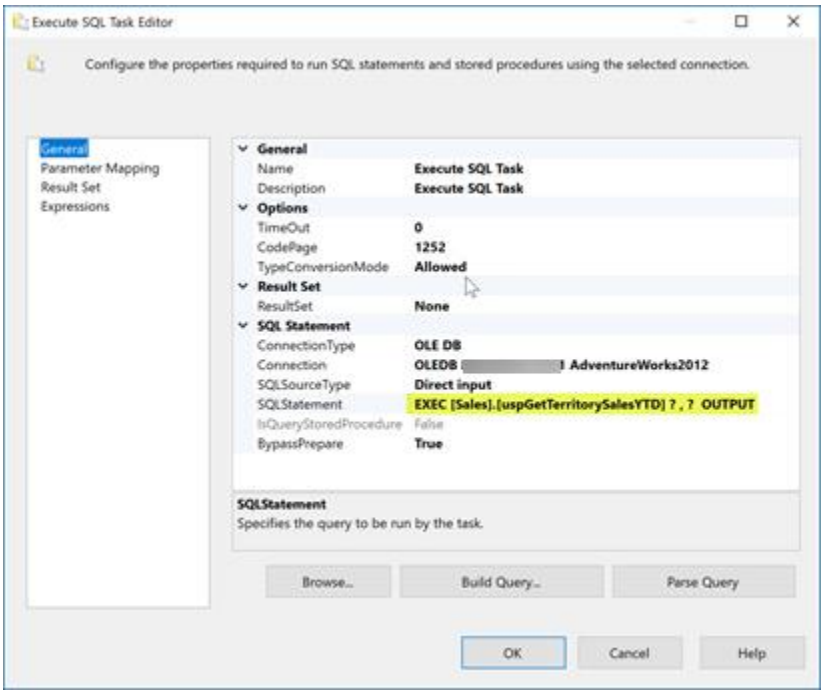


18. Double click on the Execute SQL Task on the Control Flow canvas and Set the Connection.



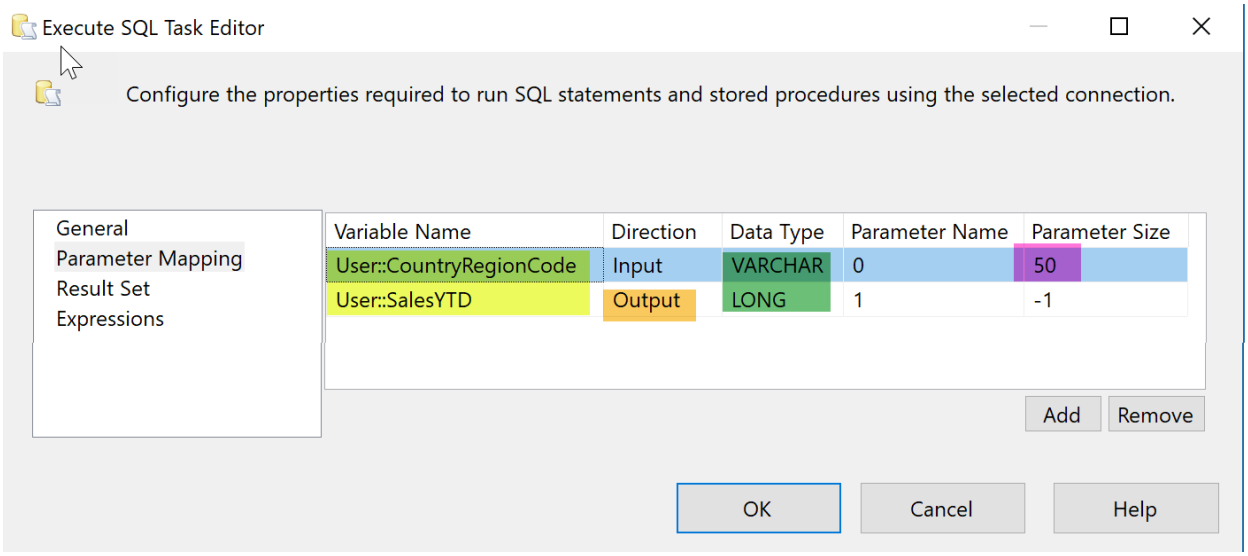
19. Enter the stored procedure all in the SQLStatement property.

```
EXEC [Sales].[uspGetTerritorySalesYTD] ? , ?  OUTPUT
```



*Note: Notice we are using the "**?**" (question mark) as markers for the parameters. Each driver has different requirements. Review the document, [Execute SQL Task – Parameter names and markers](#).*

20. Next, we must define how do stored procedure parameters map to variables in SSIS package. Click on the Parameter Mapping and click Add to add two parameters as in the below screenshot.



Note: The Parameter Name start from index `0 for OLDEB connections`.

21. Click OK to close the Execute SQL Task Editor dialog.
22. Execute the Package and notice the SaleYTD was successfully retrieved.