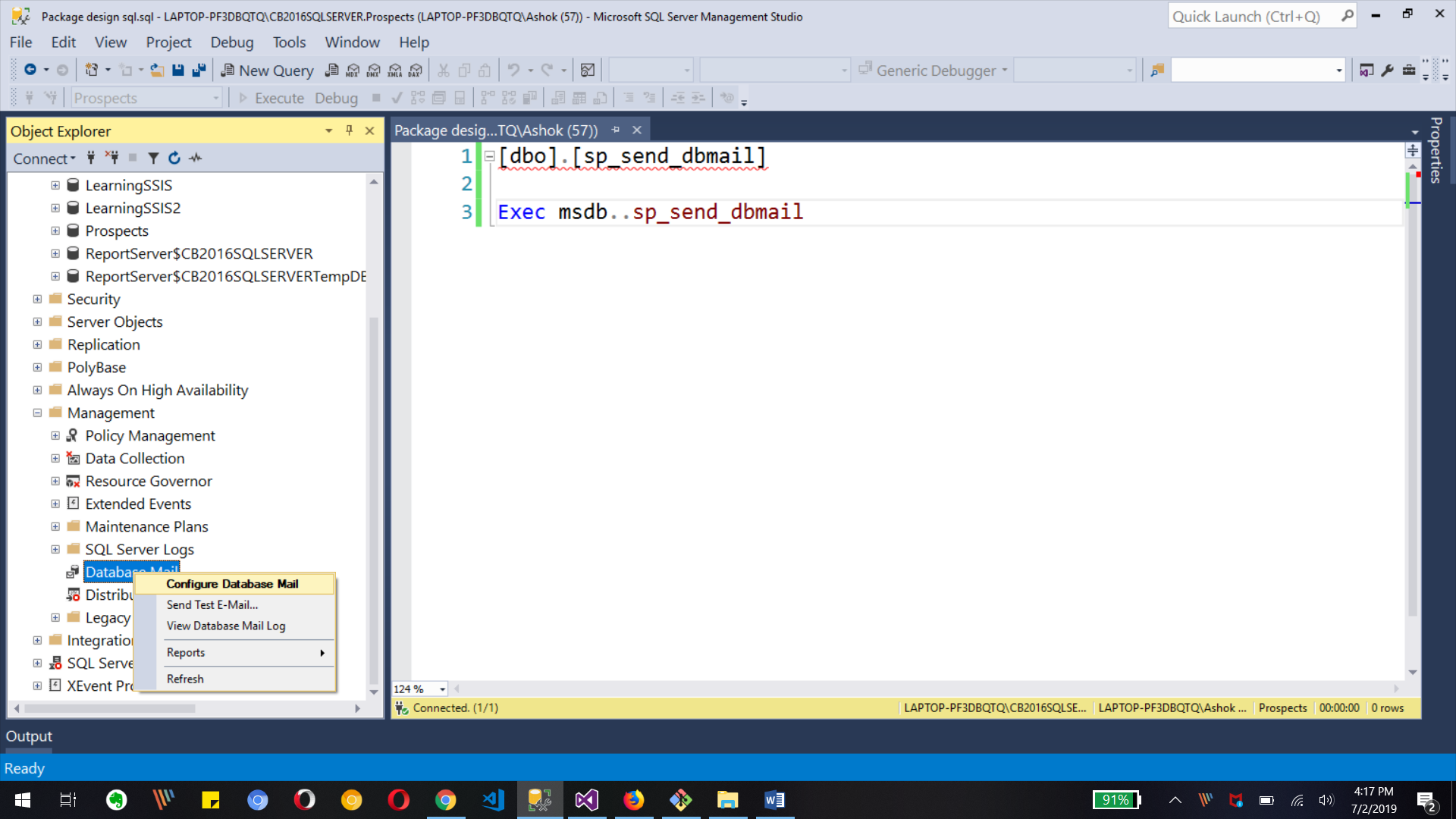
/\*

Ashok

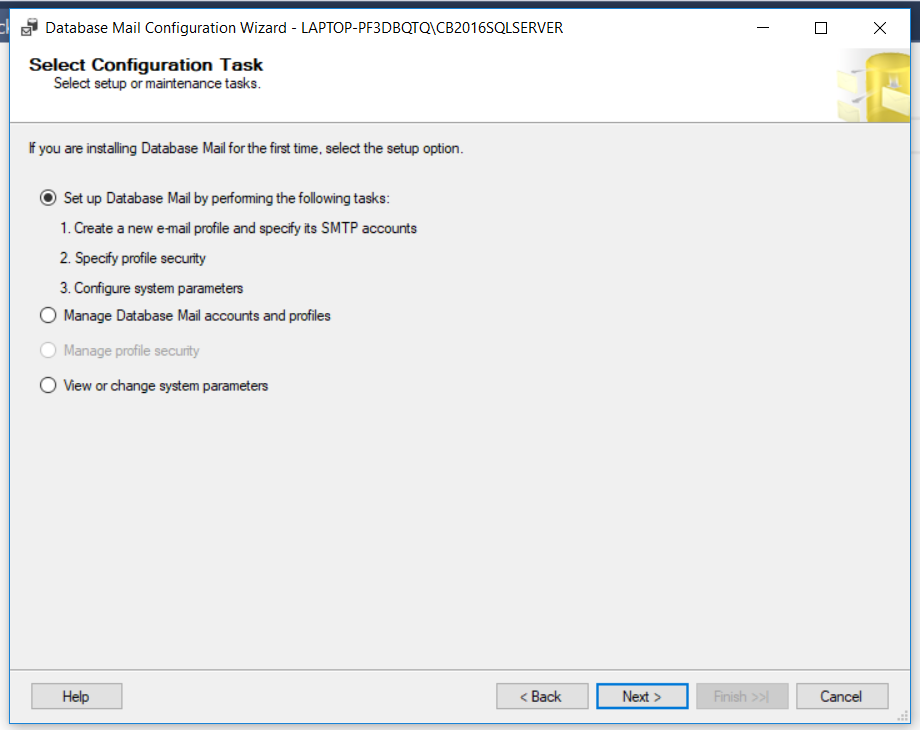
Package Design Consideration Lab

7.2.19\*/

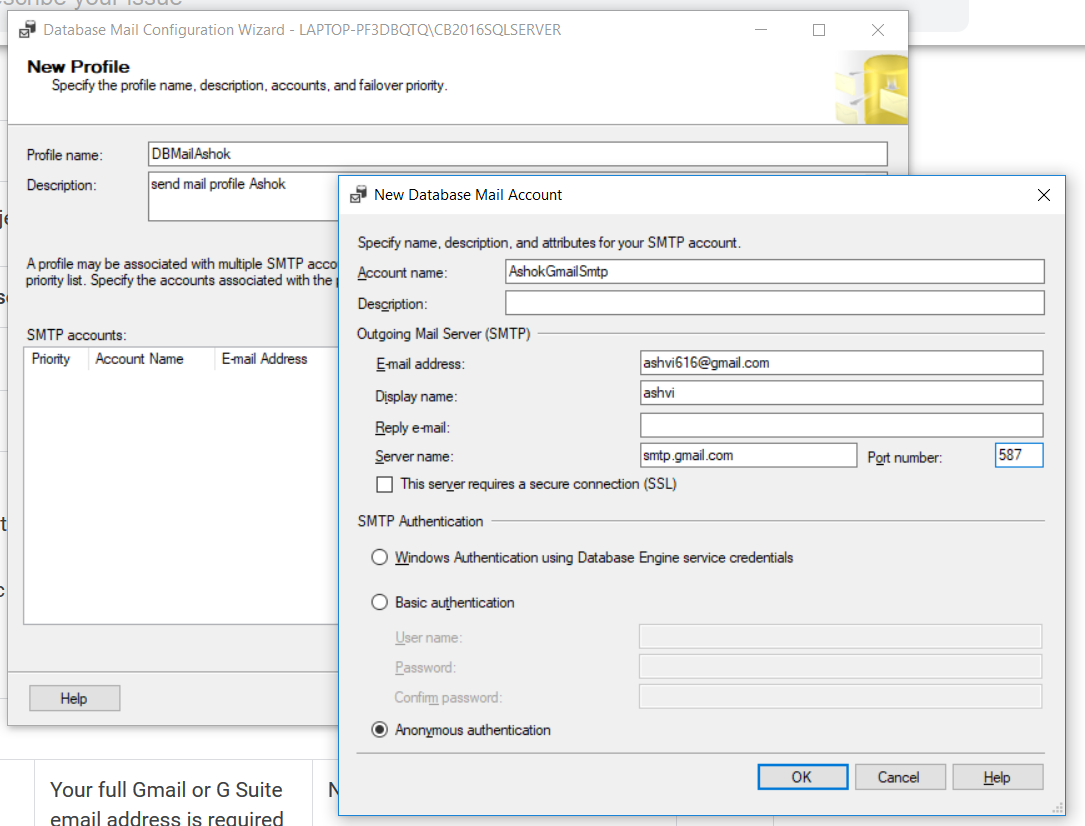
1. Configure database mail on your SQL instance.



Creating DB mail feature with right click on it to configure database mail as first time use



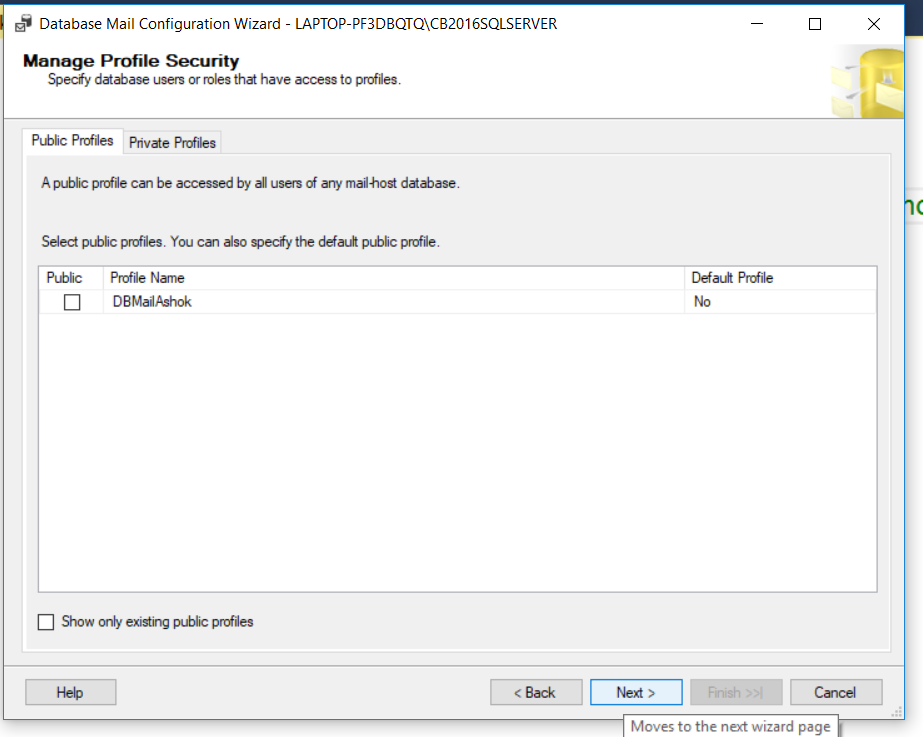
Above image will be shown for configuration so need to select set up DB mail first check box



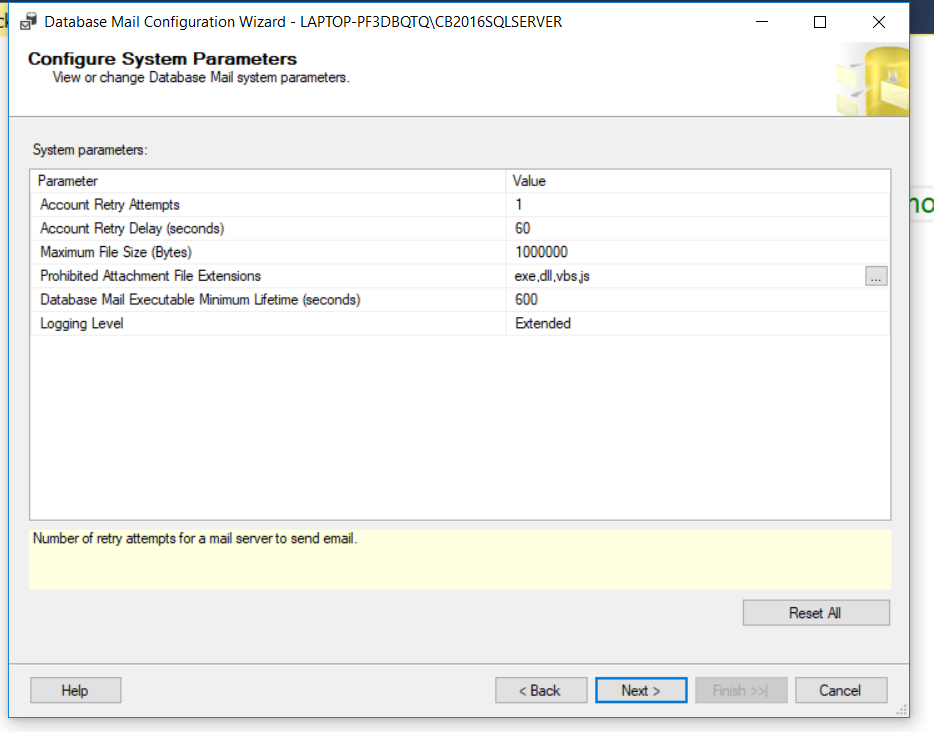
As shown in the above image we need to provide profile name and SMTP account detail so click add in it and fill the New database mail account detail as per above with your email address and password



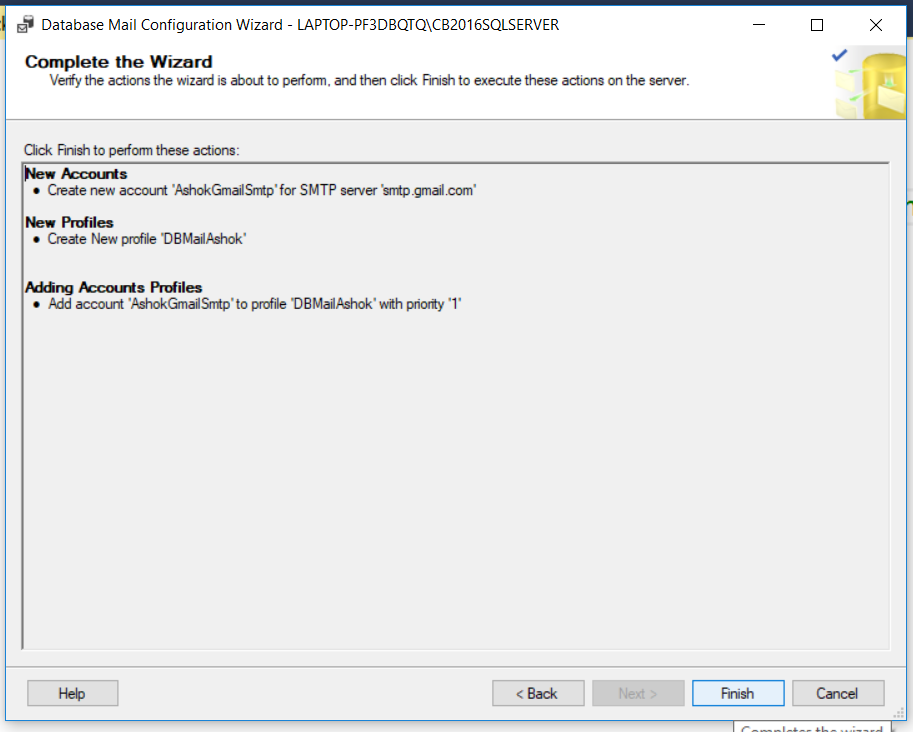
Final look of first step in configuration for profile and SMTP account. Click next now.

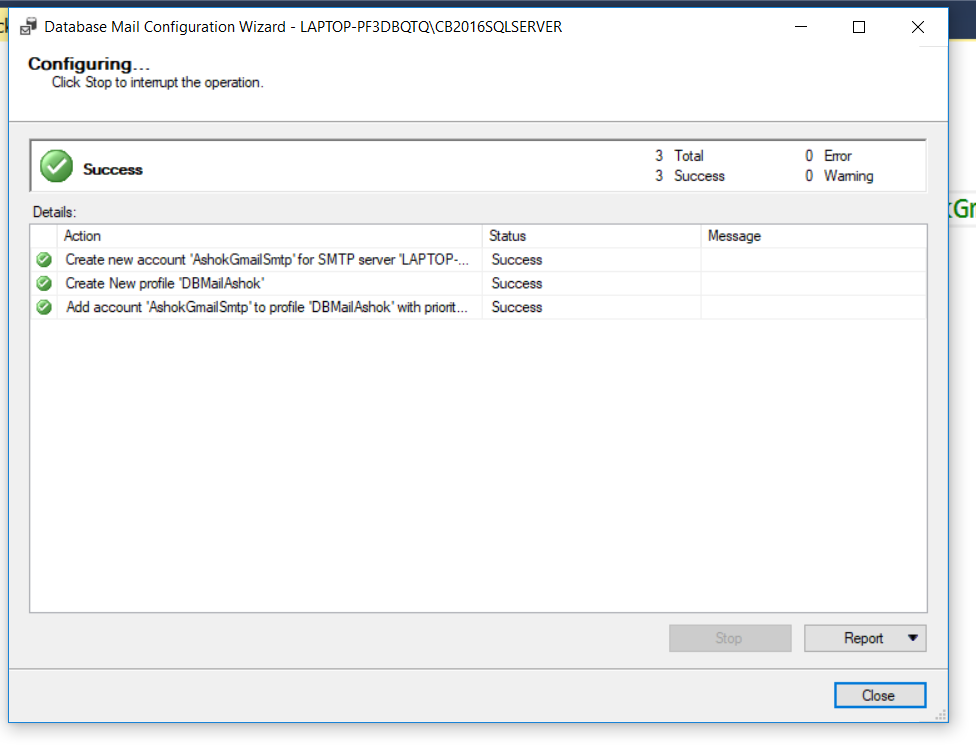


We need to manage the security of profile as public or private. Click next



Now this is final Parameters of configuration. And click next for final step



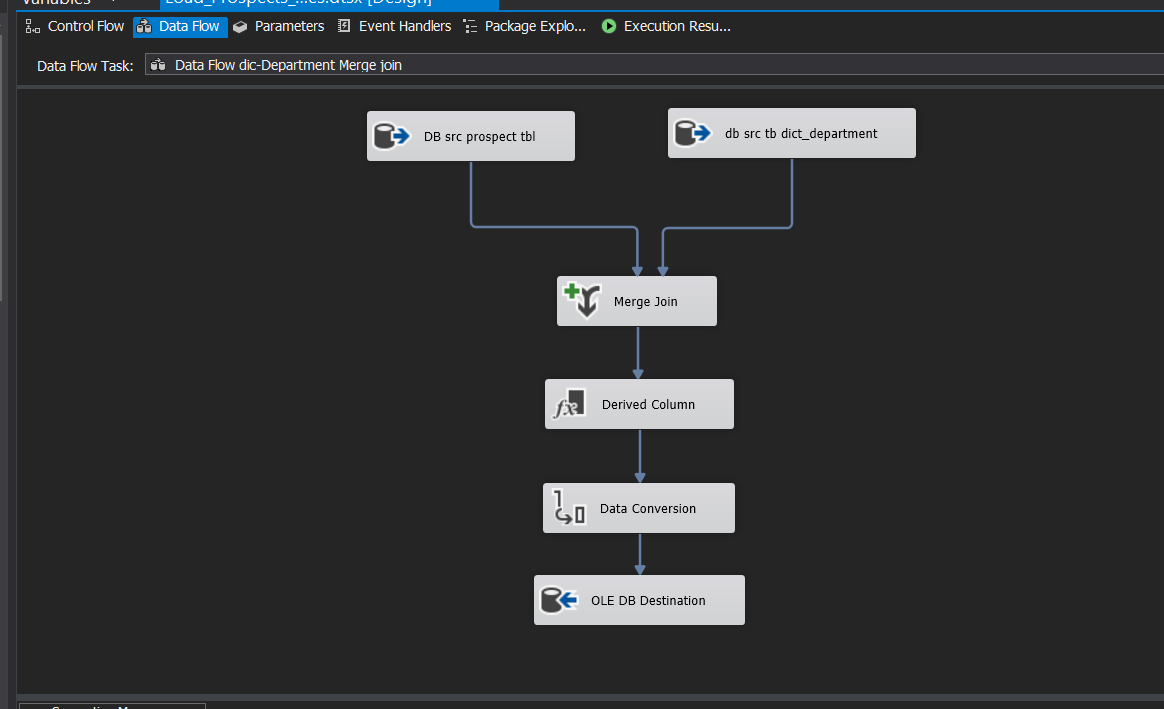


This is finish and final step for configuring database mail in SSMS/sql instance.

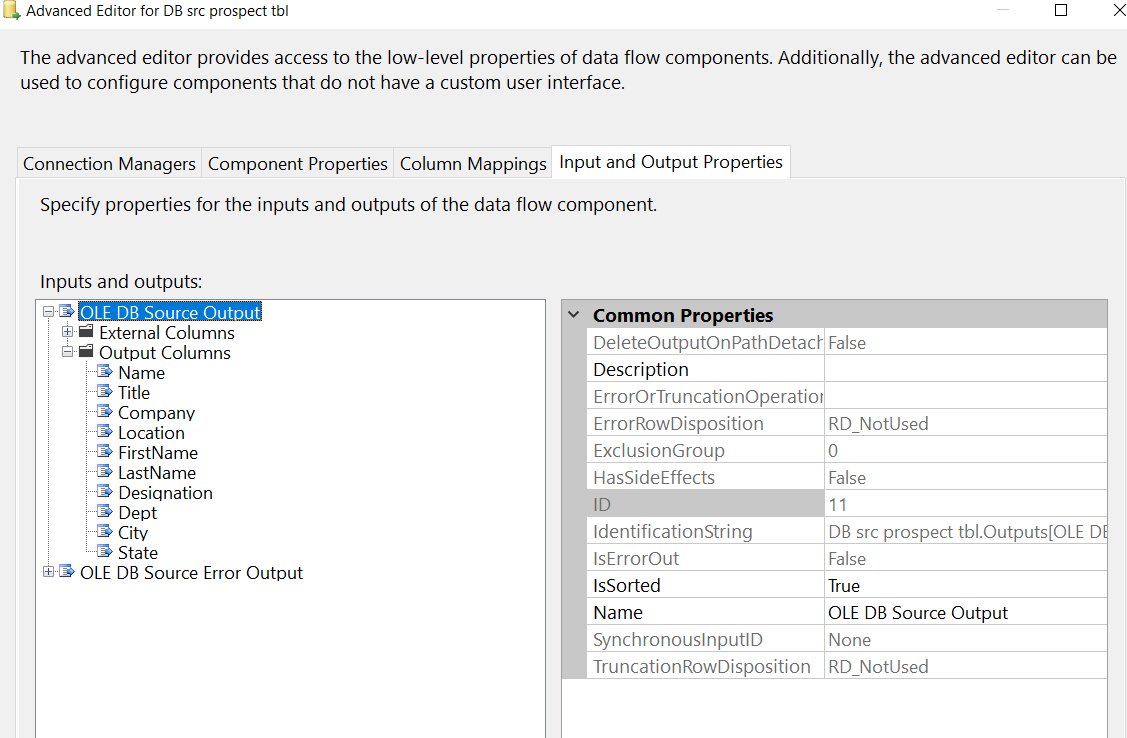
2. Modify Load\_Prospects\_Dictionaries.dtsx to include the following

a. Use Merge Join while loading Title and Department dictionaries instead of Lookup

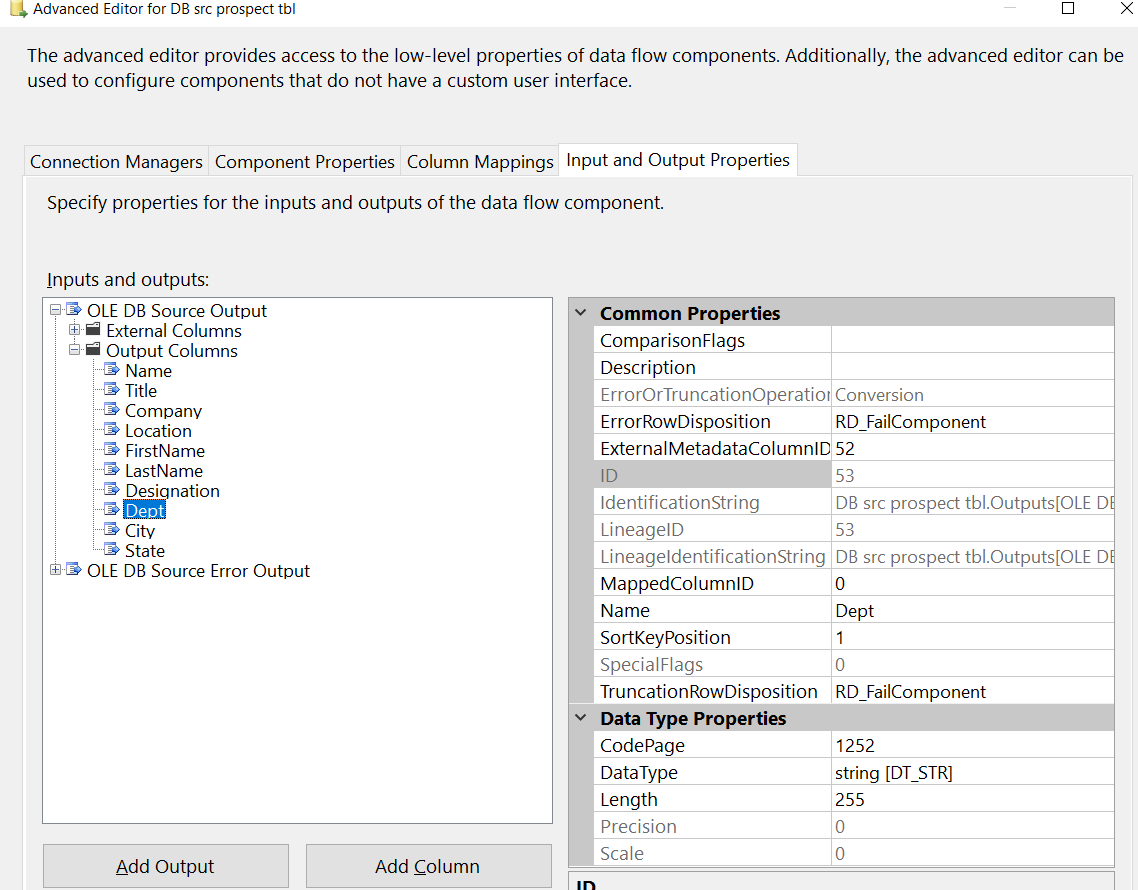
component.



The above image is for Dictioanry\_deparment table with merge join. We need two separate table source one from prospect and another from department to join together.

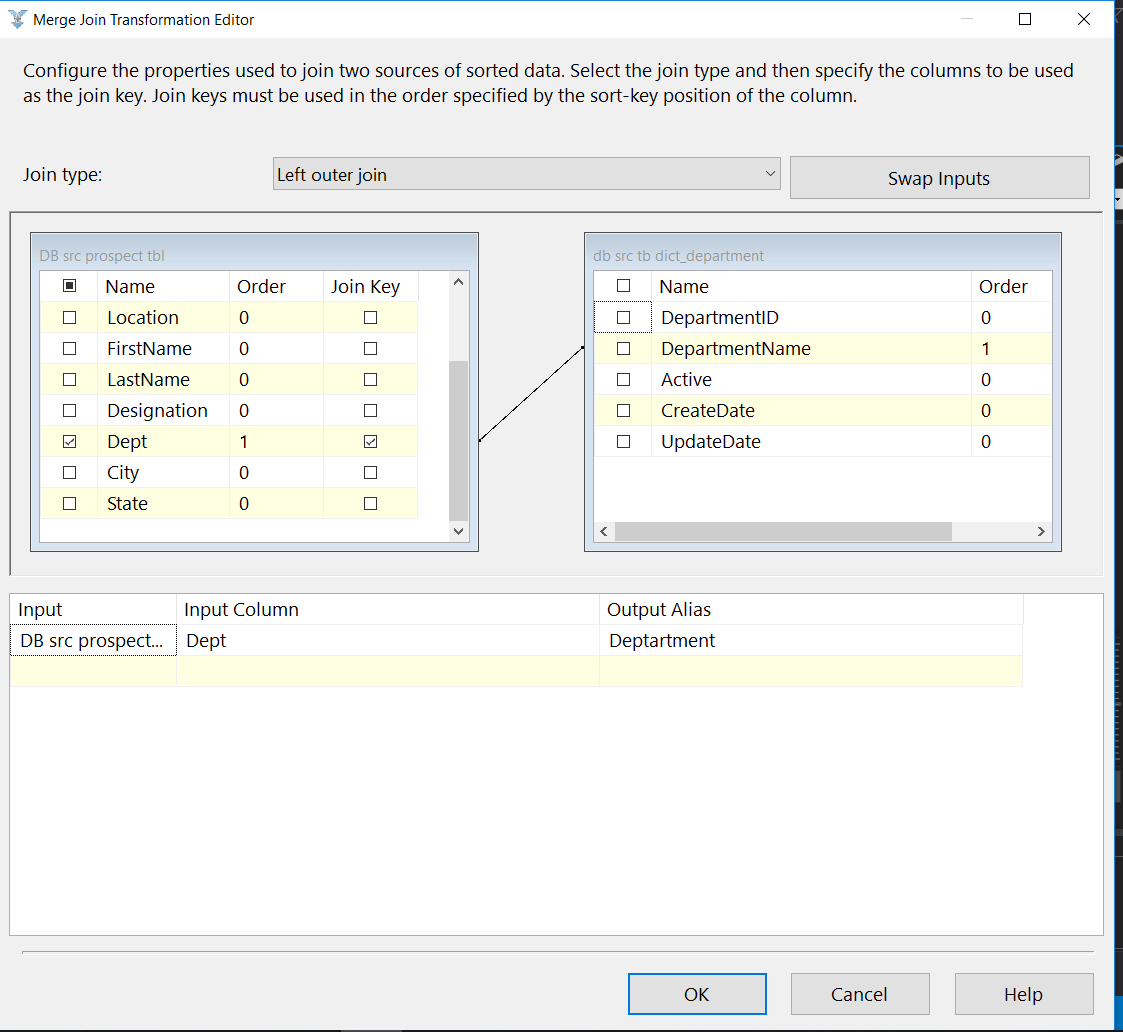


We need to sort the coming data from source tables so we have to set isSorted to true in both source’s output advance Editor.

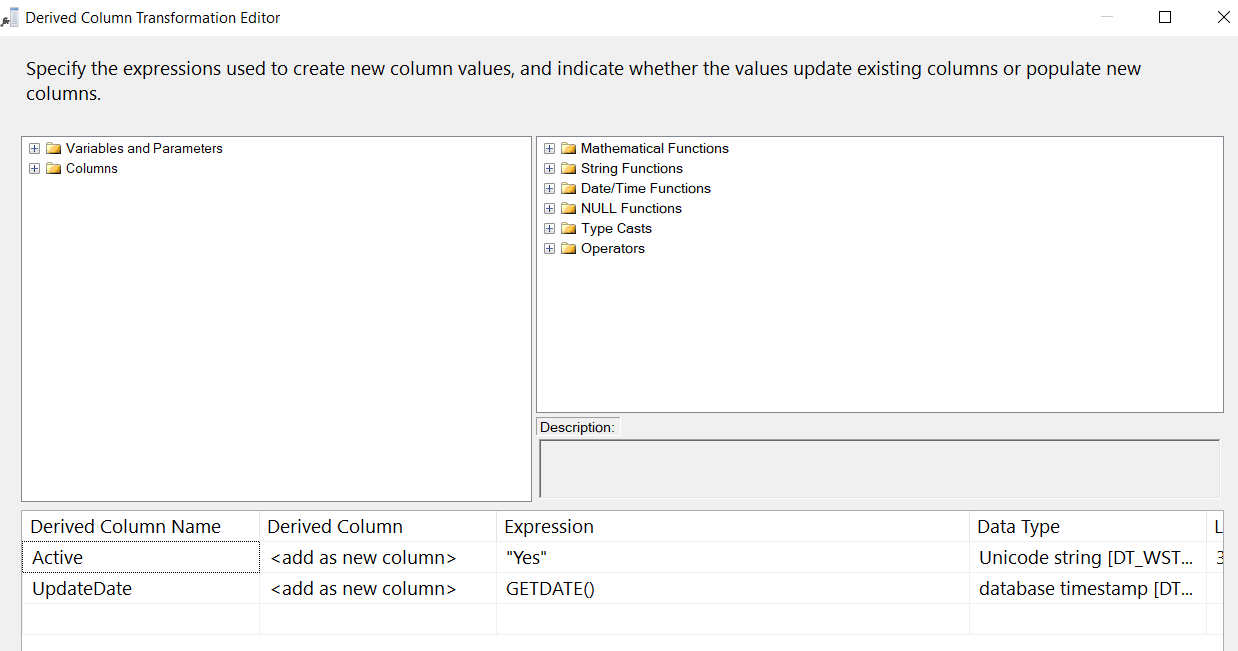


and We have to set sortkeyporition to any specific digit to any one column(Dept) for sorting reference.

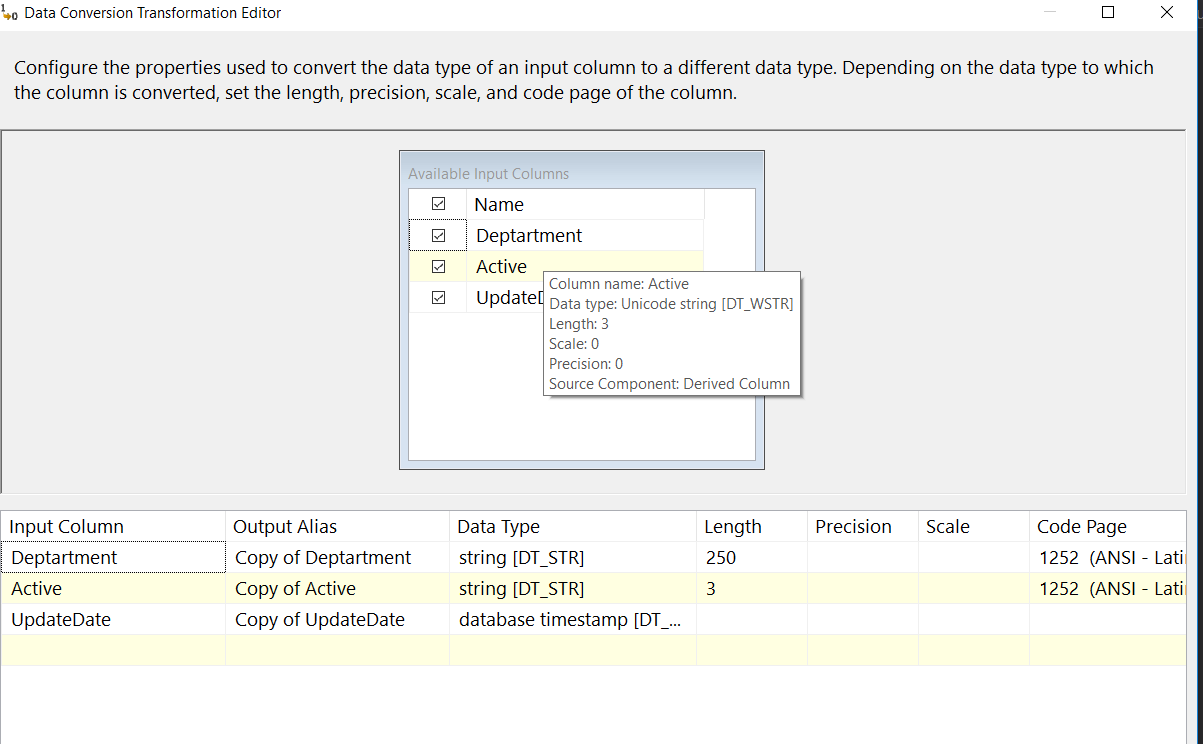
Also we have to set same isSorted to true and sorkkeyPosition to same digit in another source so both tables can be sorted and join together in merge join.



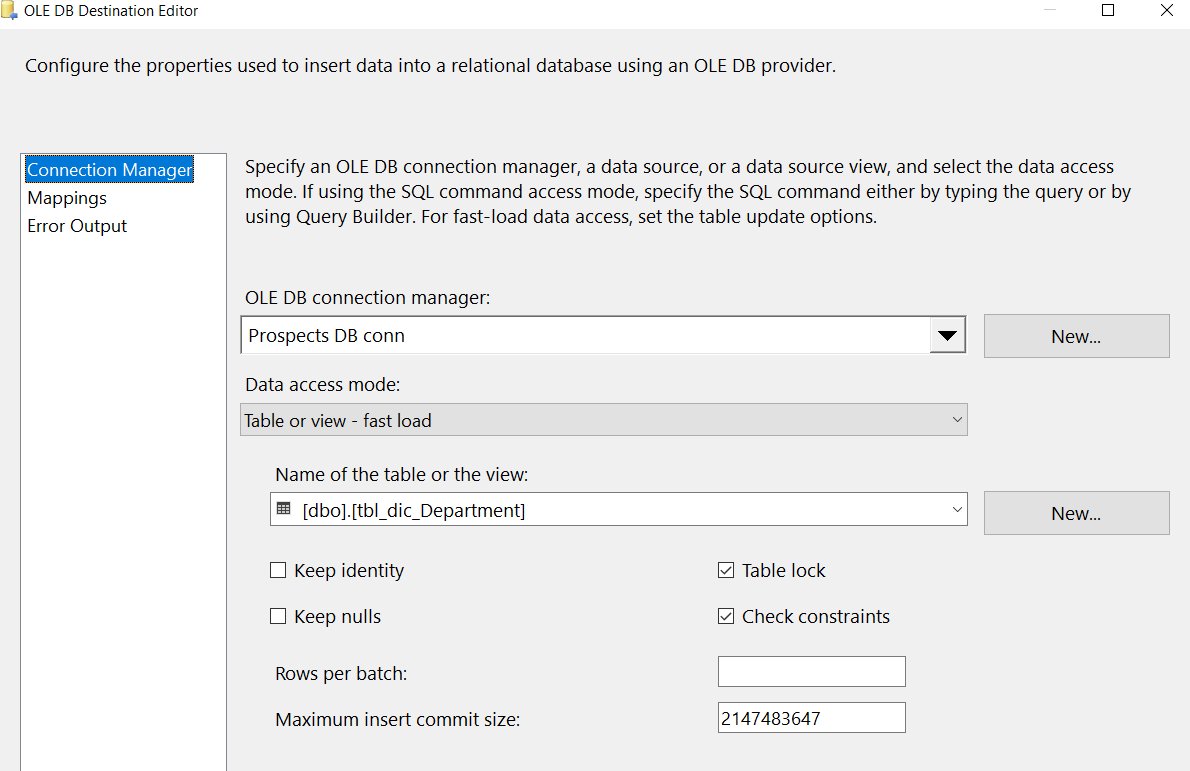
Above image shows the Merge join transformation editor where we need to select join type as left and mapping column from left and right sources as shown above where we can see dept and departmentName and we need only data from Dept column so we selected that column from left tbl.



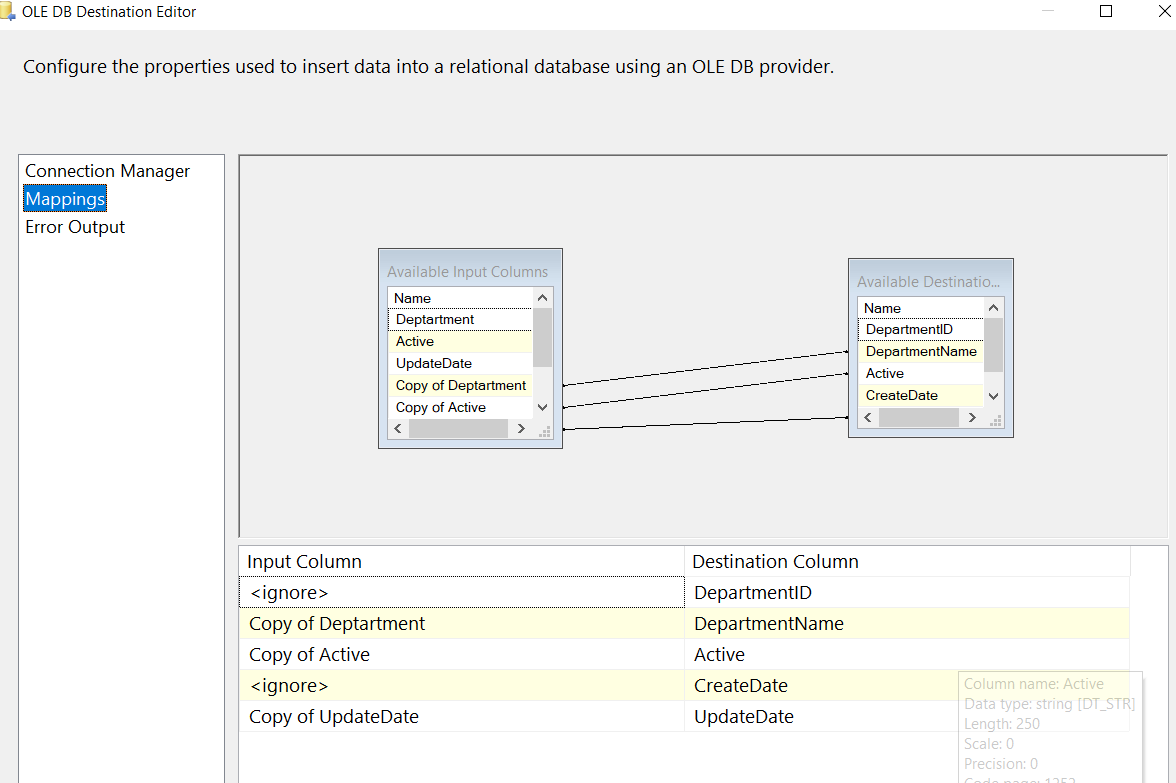
Now we need to get two column data for Active and UpdateDate to insert into DB table so we use derived column transformation and set value as shown.



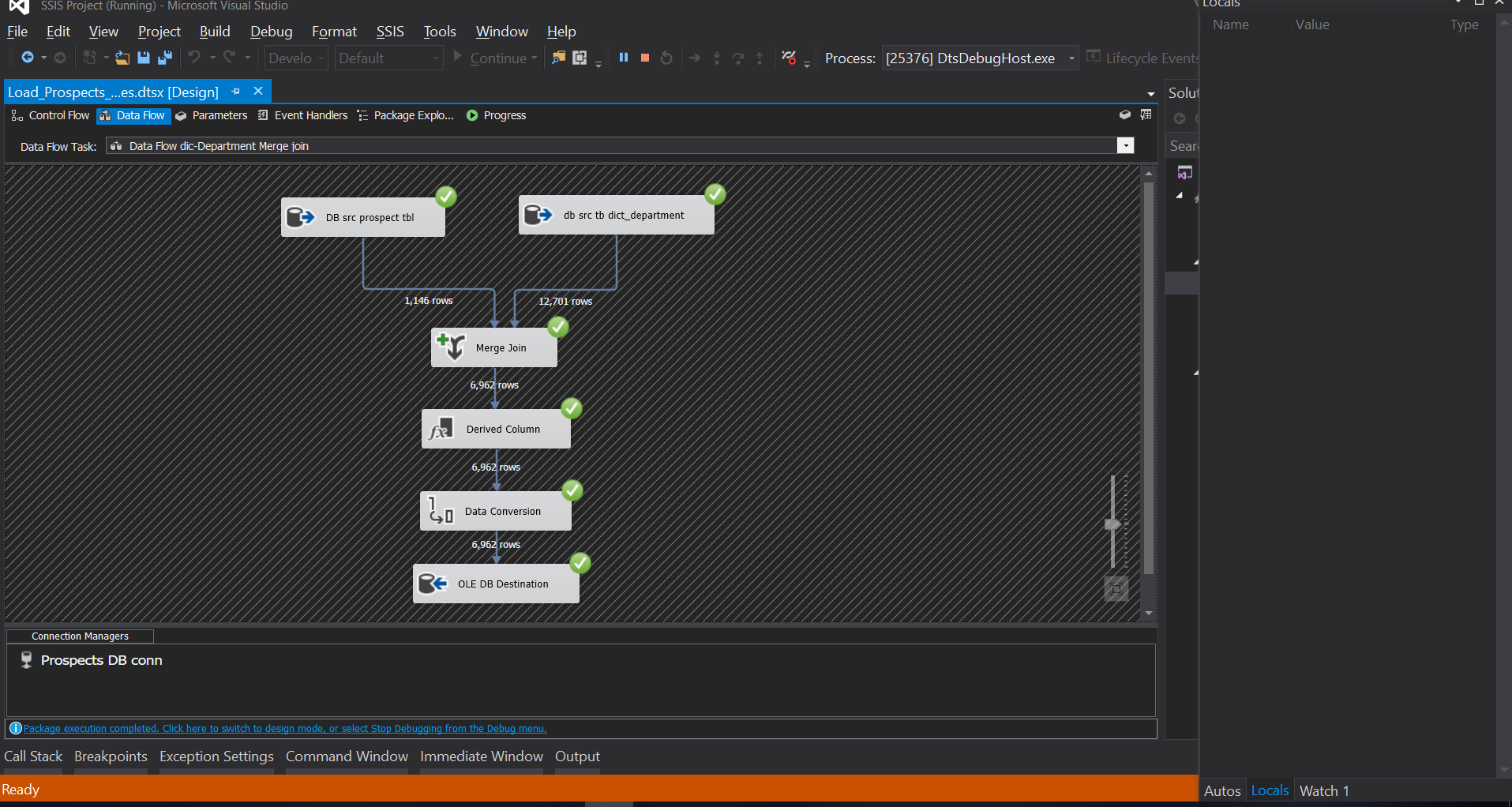
Also we need to change the data type for derived columns and input columns so we use data conversion transformation to set input columns data type as shown in above image.



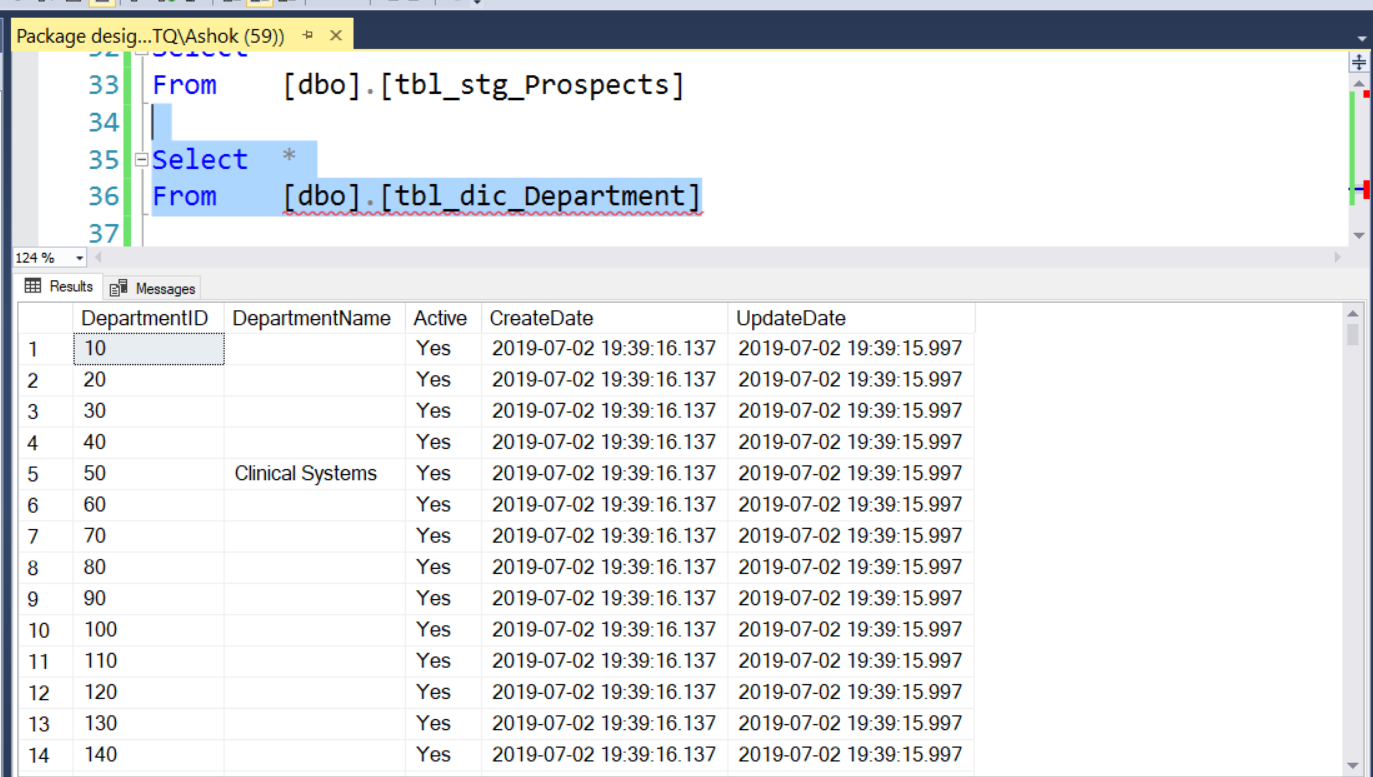
Above image is for Ole db destination editor for loading destination table with input columns from up stream. We have Department table selected for it.



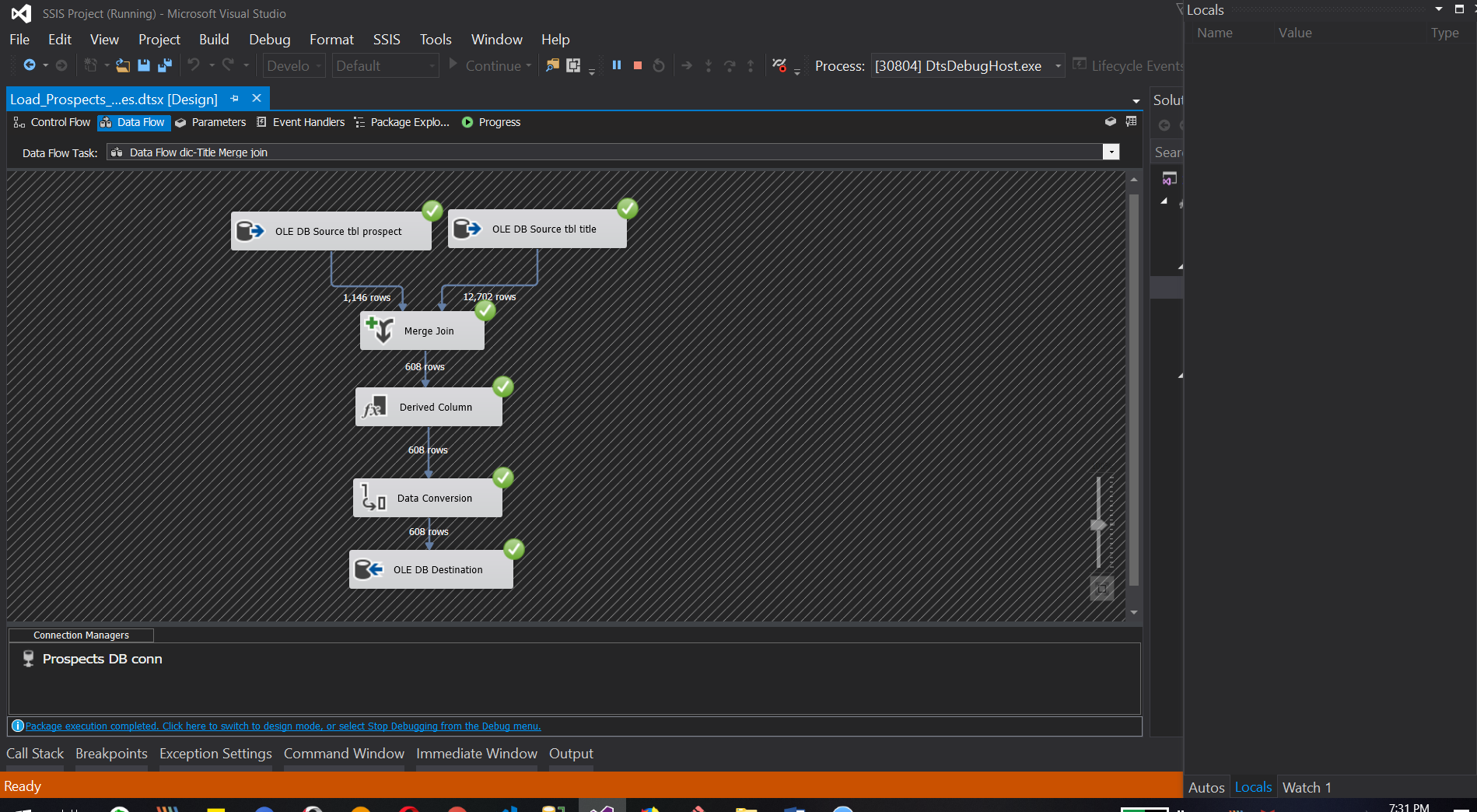
Now we need to map input columns and destination columns shown as above image.



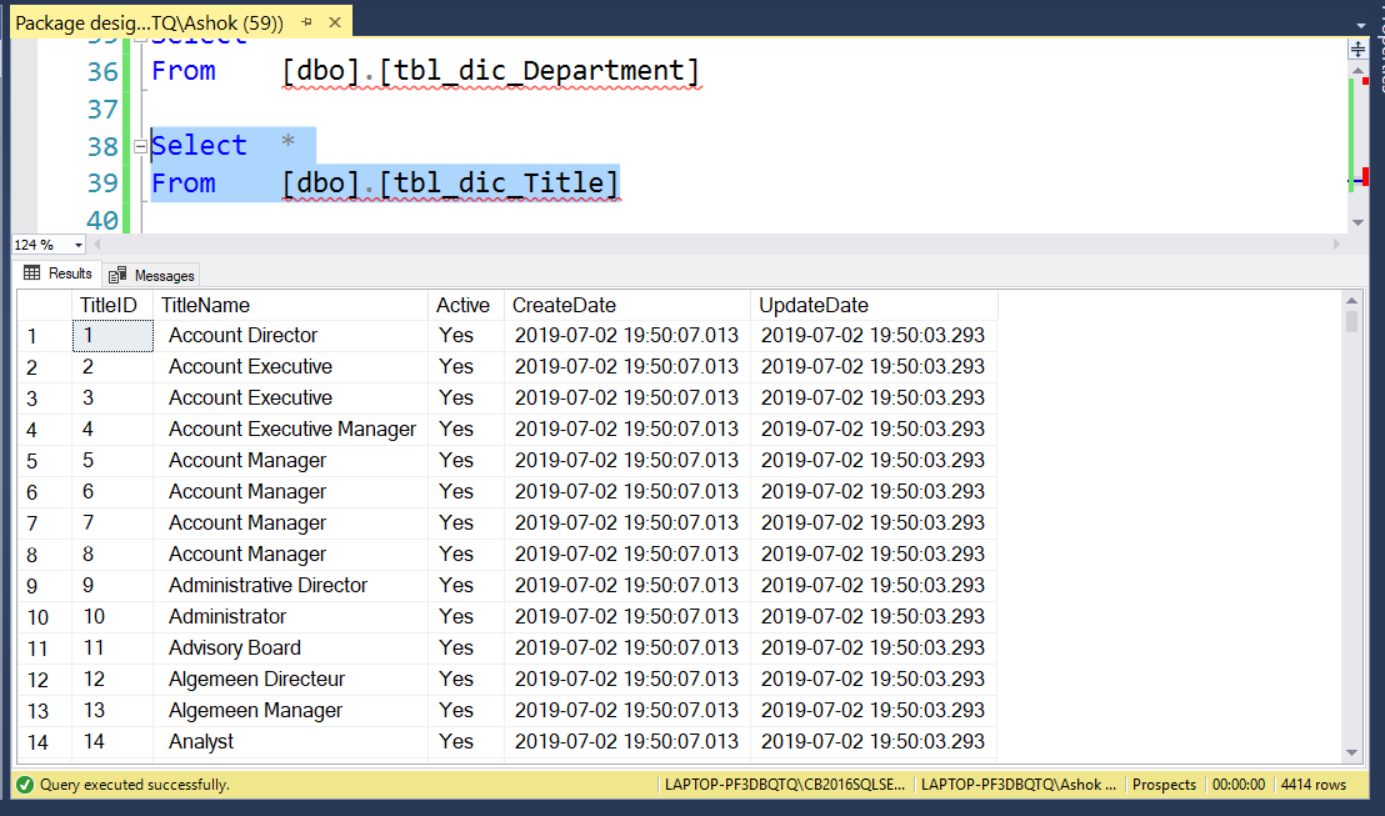
Lets run the package and see the output in ssis.



Final insert for table from SSIS as shown in above SSMS output.



Now we can do same process which was done for Dictionary\_department table with merge join for dictionary\_title table on titleName column to insert data into SSMS sql Title table.



The above output is for dictionary\_Title table in SSMS for inserted data with the help of ssis package.

b. Include related tasks on the control flow tab of all the packages in sequence containers

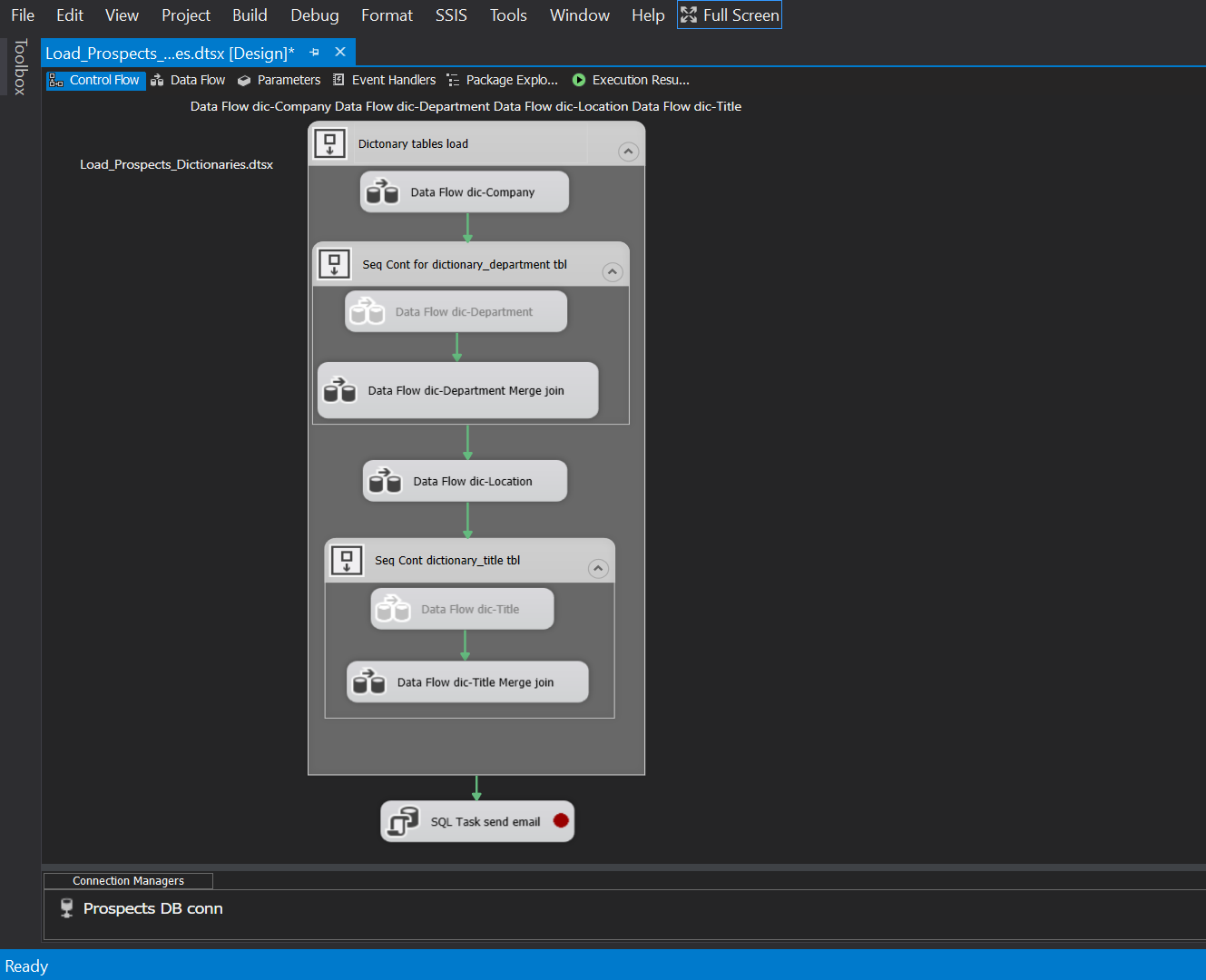
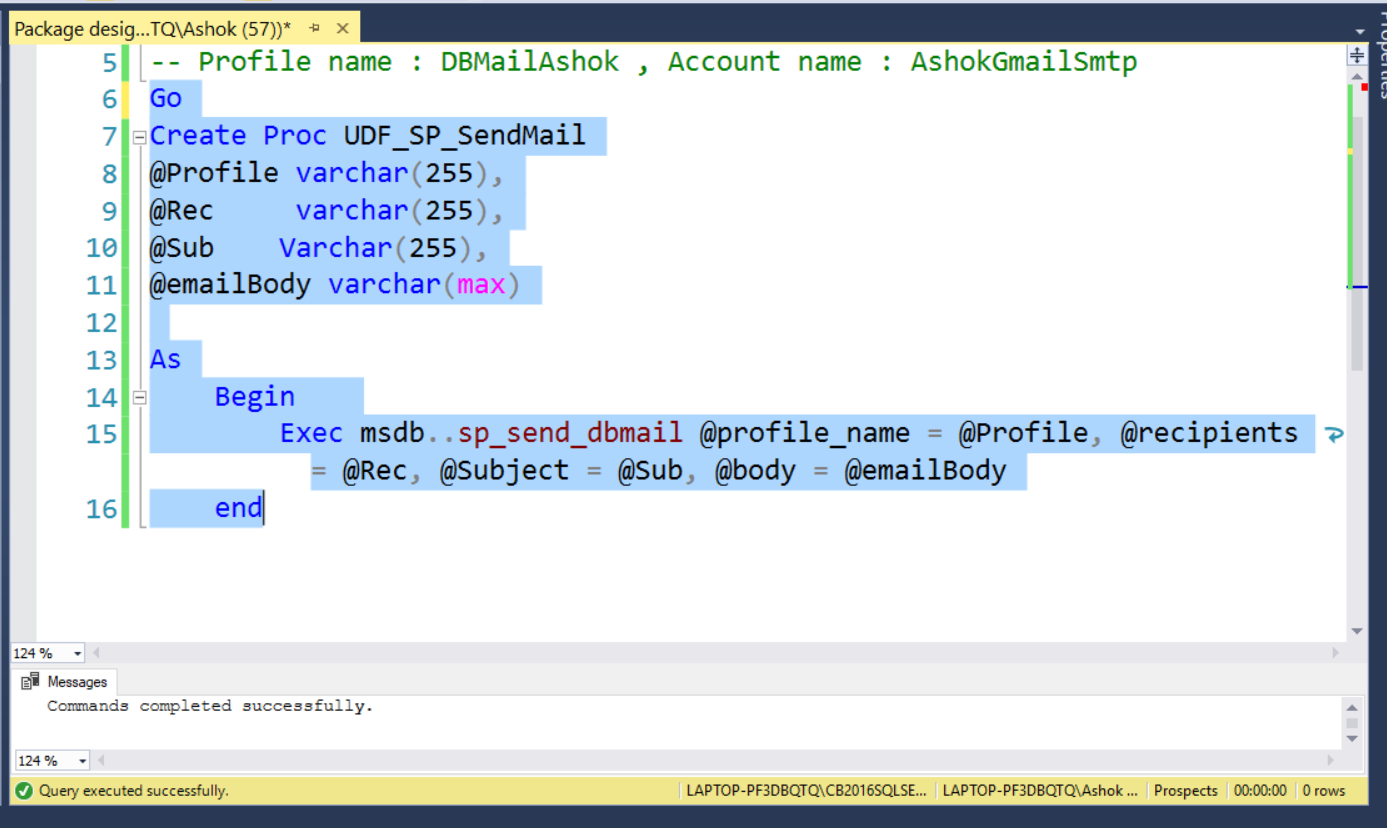


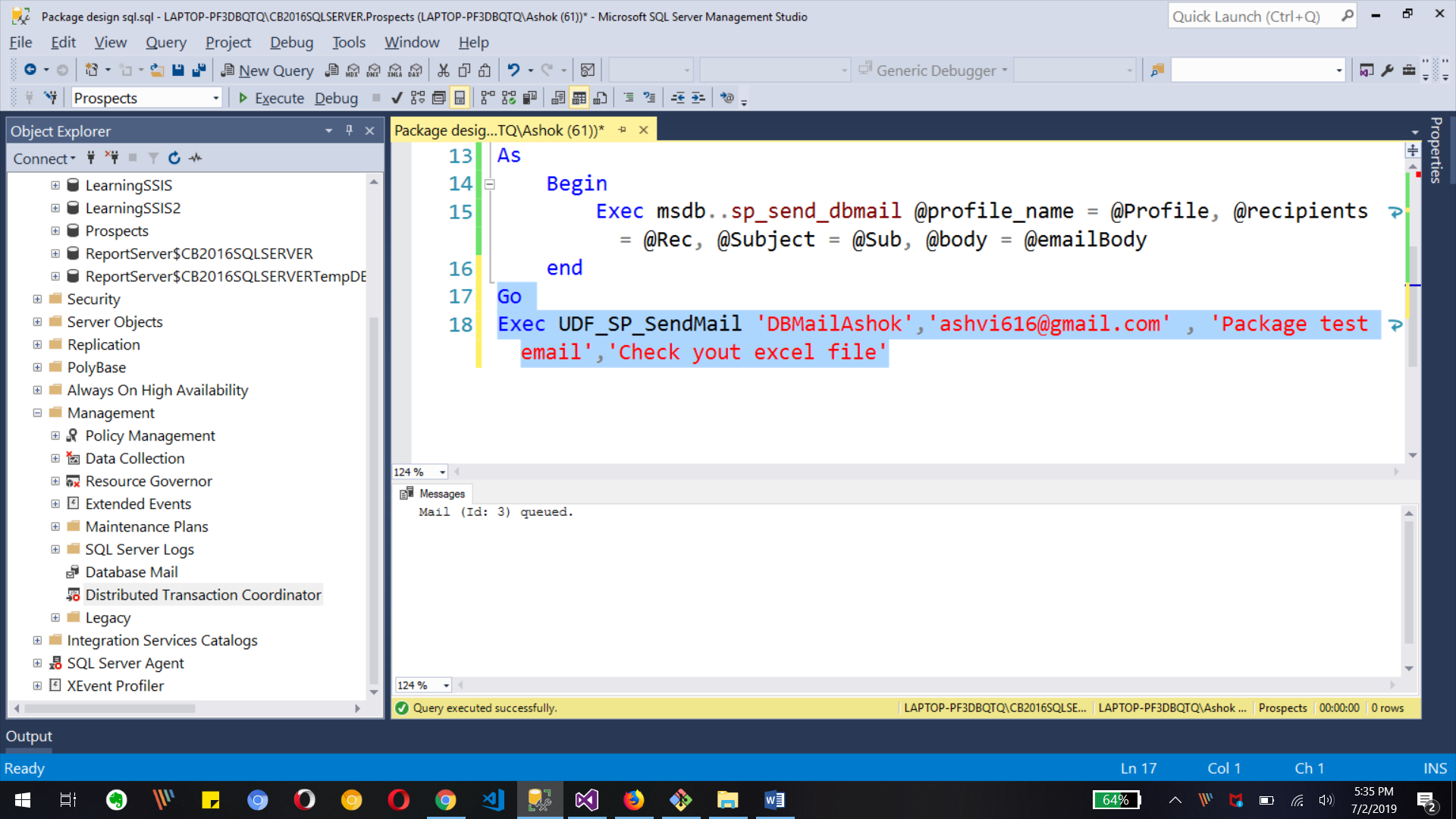
Image shows the related tasks are set inside of Control flow for better execution and understanding.

c. Implement database mail in your package to send out emails in the event of package

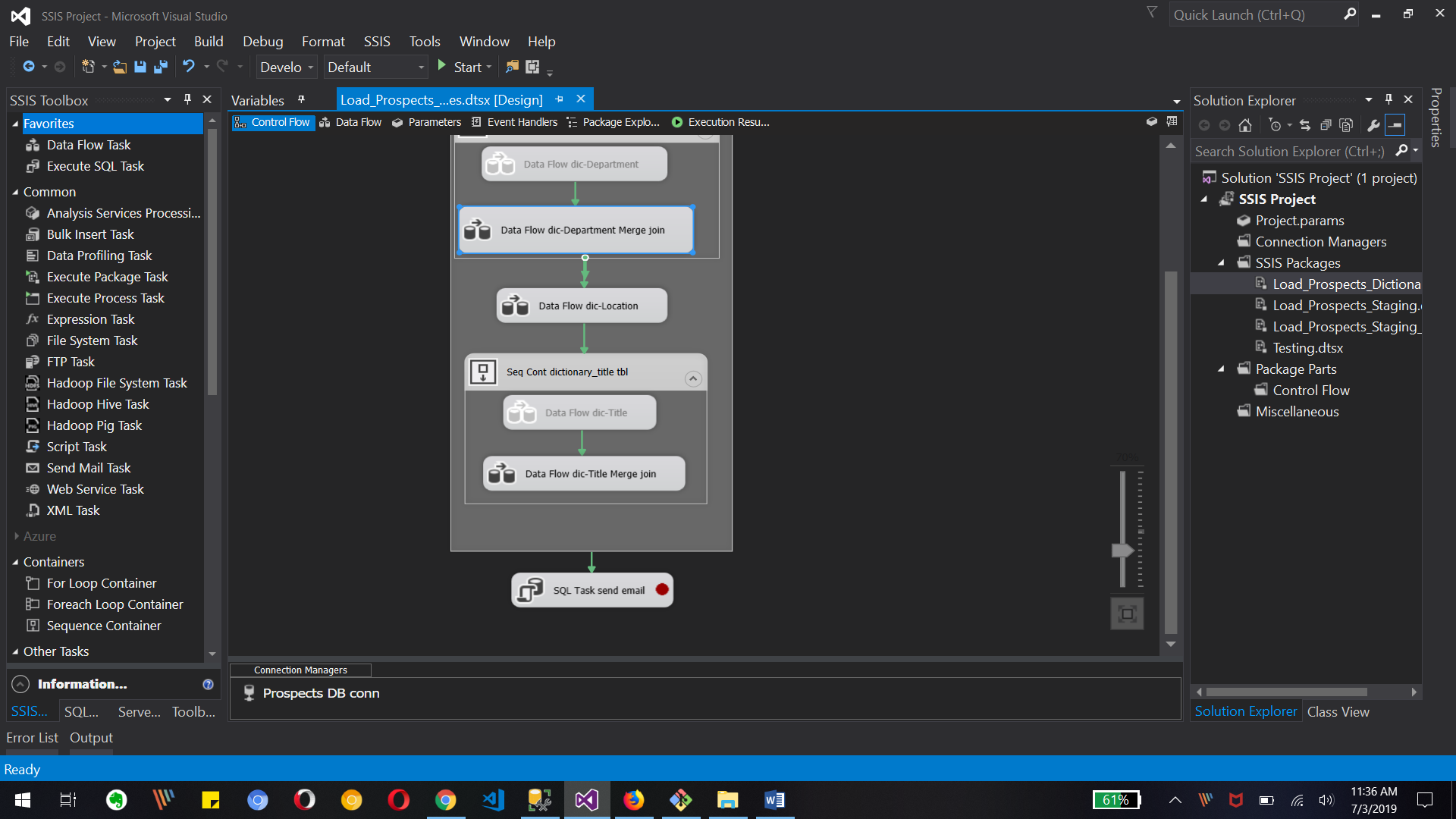
failure, successful file load and error files



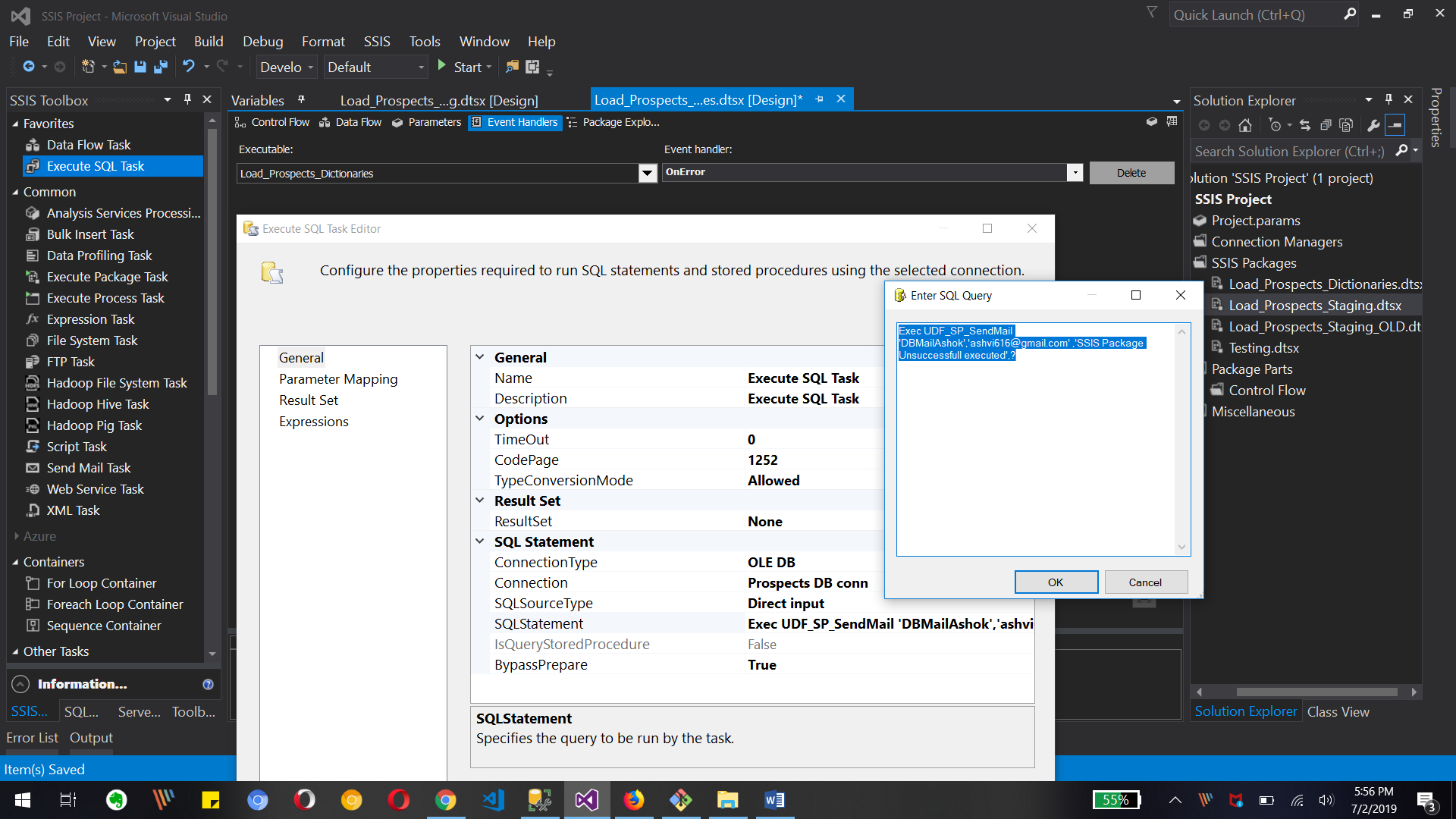
--- To implement database mail in package we have to create user stored procedure(UDF\_SP\_SendMail) which will use system stored procedure(sp\_send\_dbmail) for database send mail as shown in image.



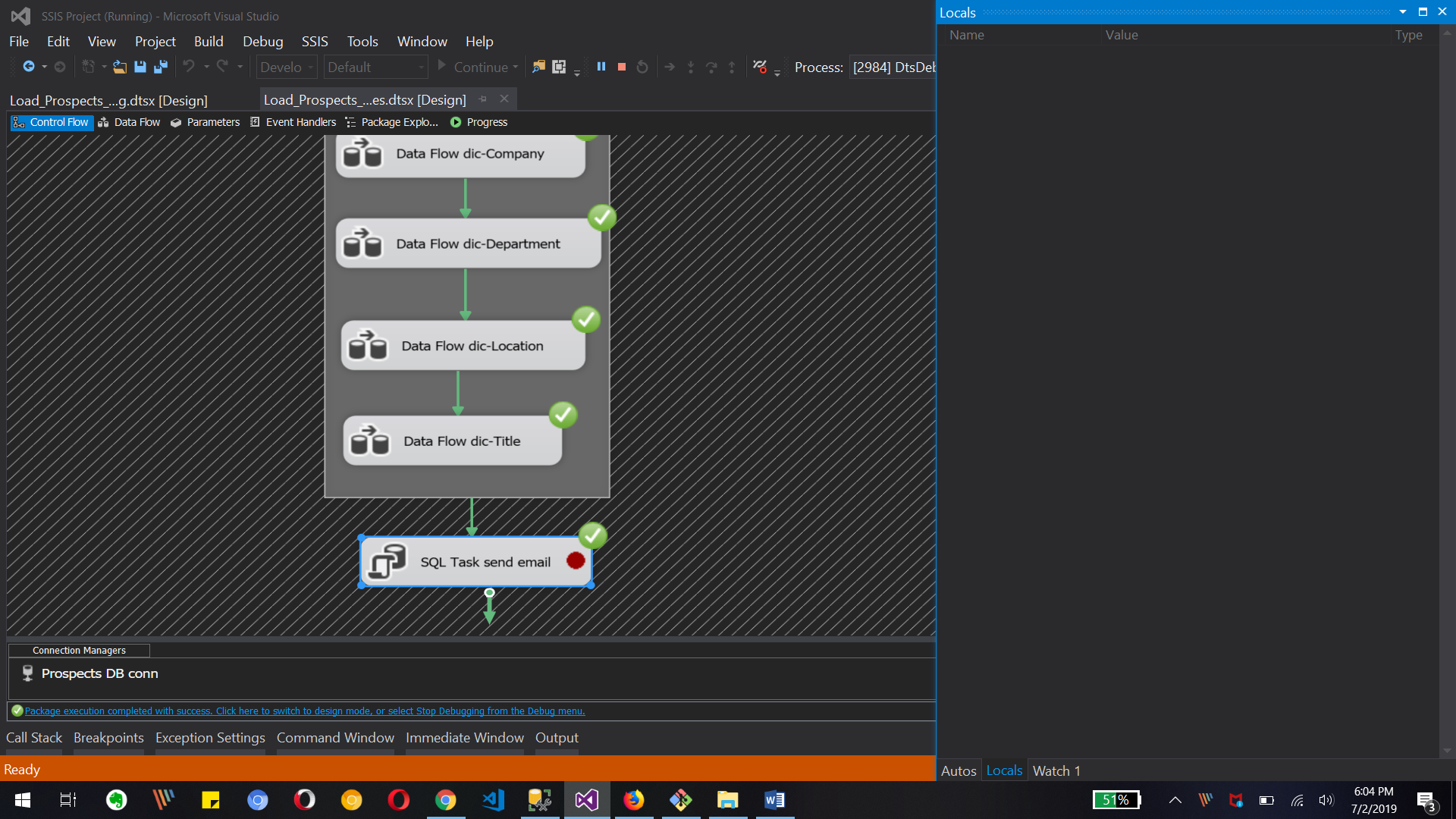
Also we can test the store procedure as shown in above image with message Mail queued.



To implement send mail we use Execute sql task with query at the end of all process as shown in image.

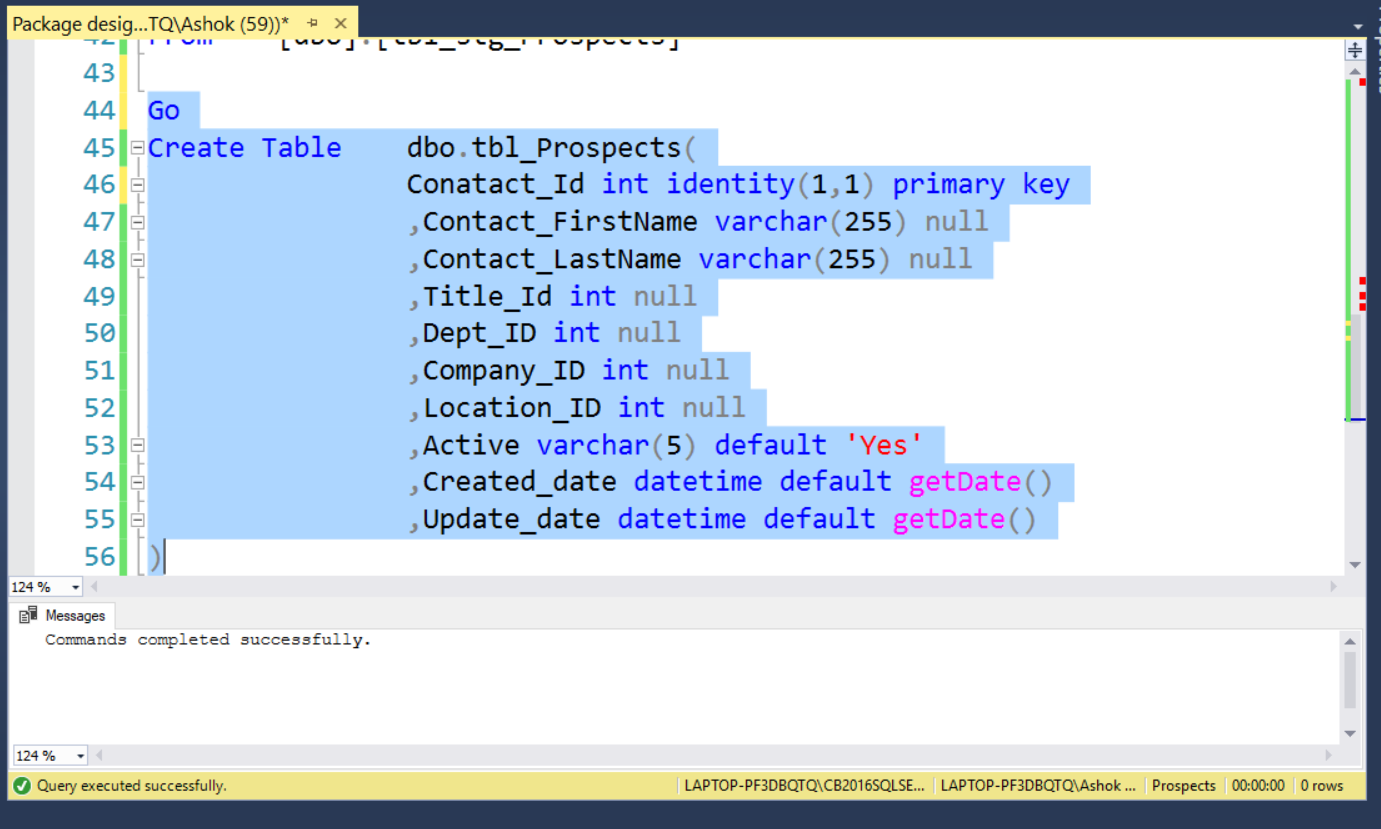


We can edit execute sql task with sql statement/query as shown in above image to call SSMS/sql server database stored procedure to send mail.



The above image shows the execution of whole package with send mail task.

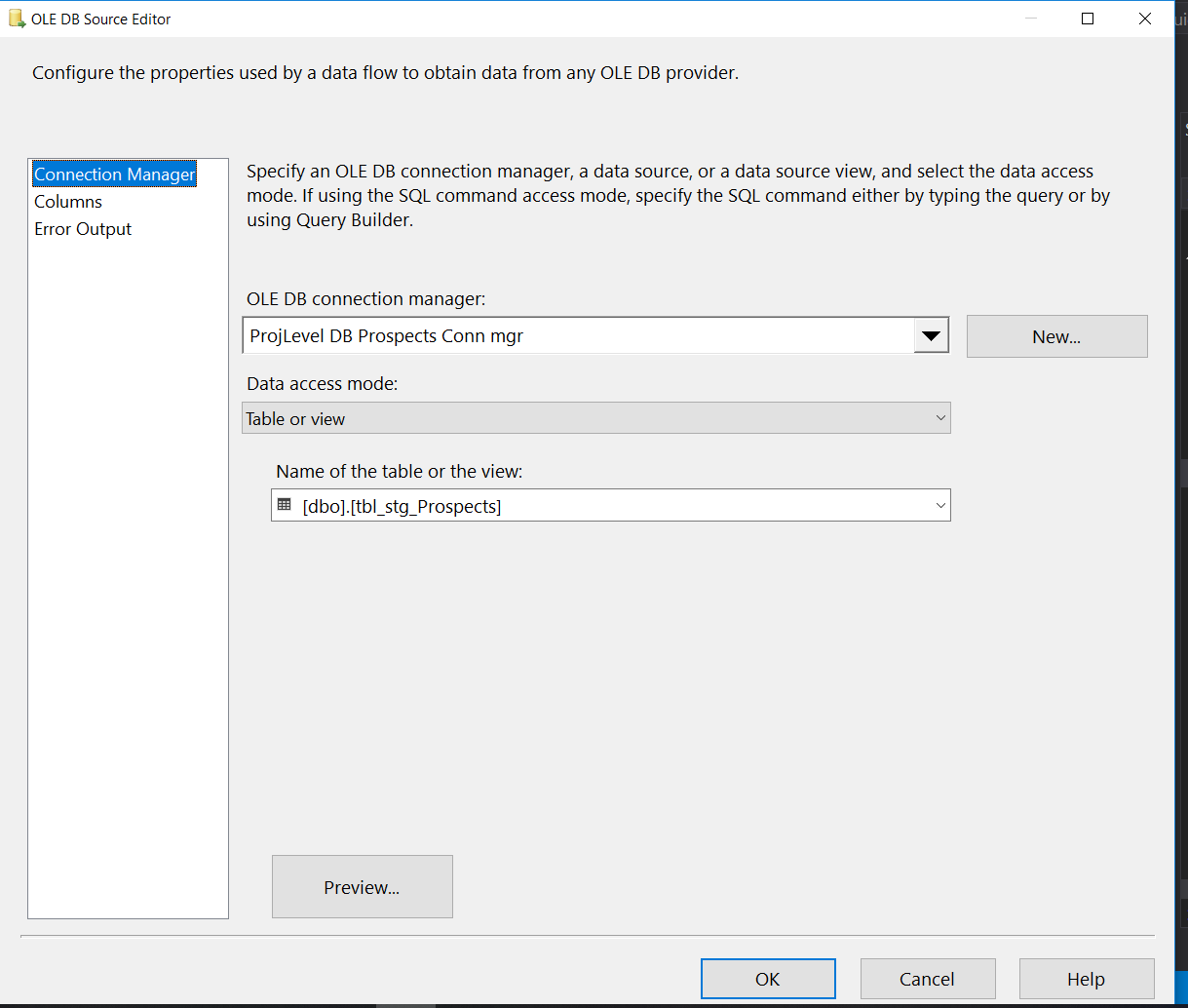
3. Create table tbl\_Prospects - Refer to the project documentation for details



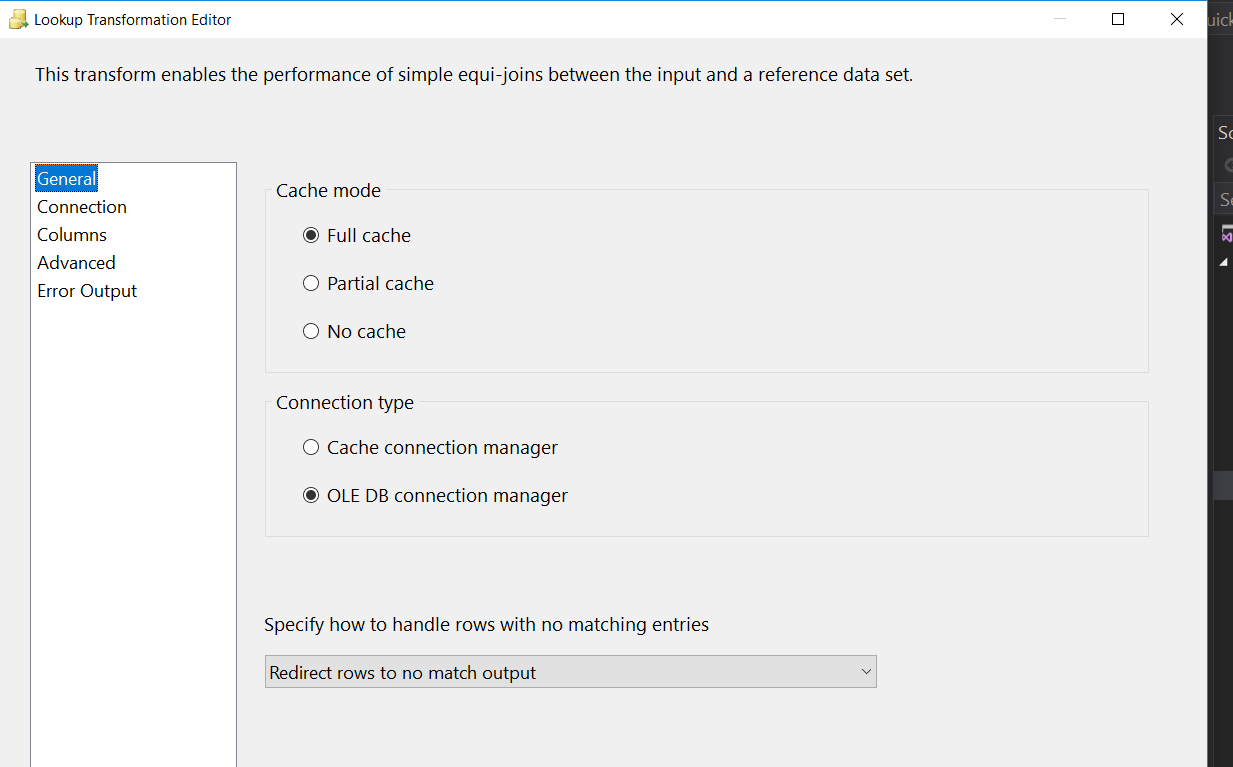
Created prospects SQL table with required columns as shown in above image.

4. Create package Load\_Prospects.dtsx to read from staging and dictionary tables and load the

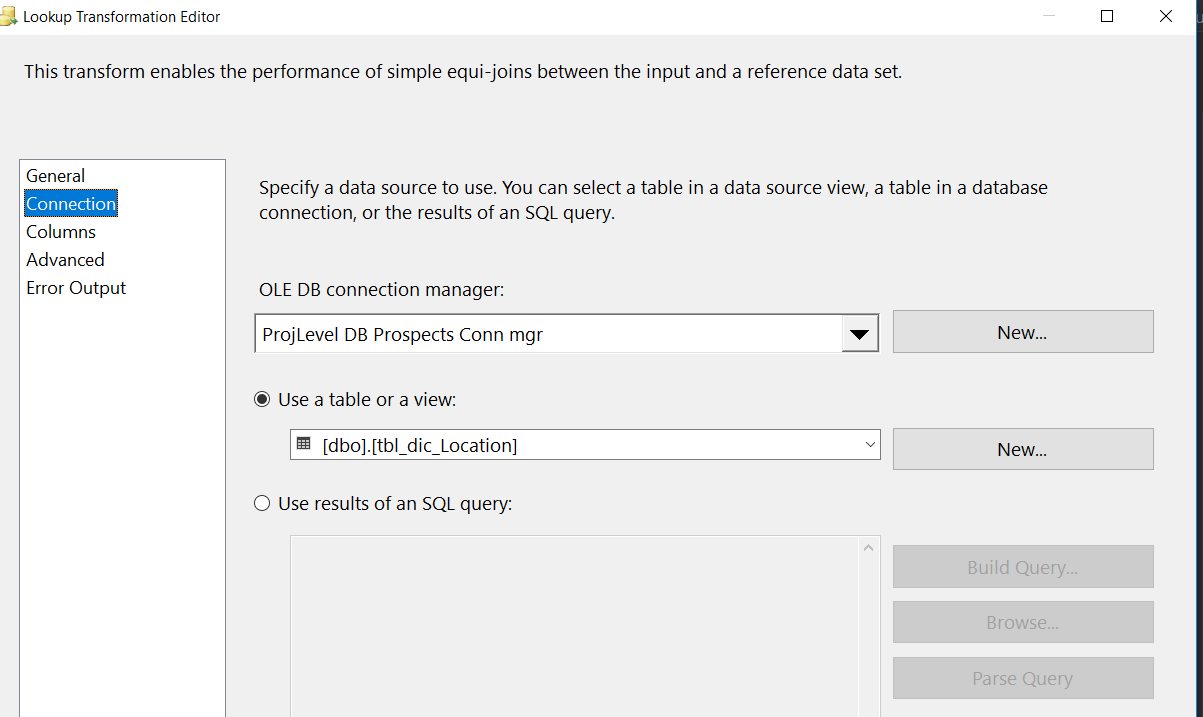
final tbl\_Prospects table.



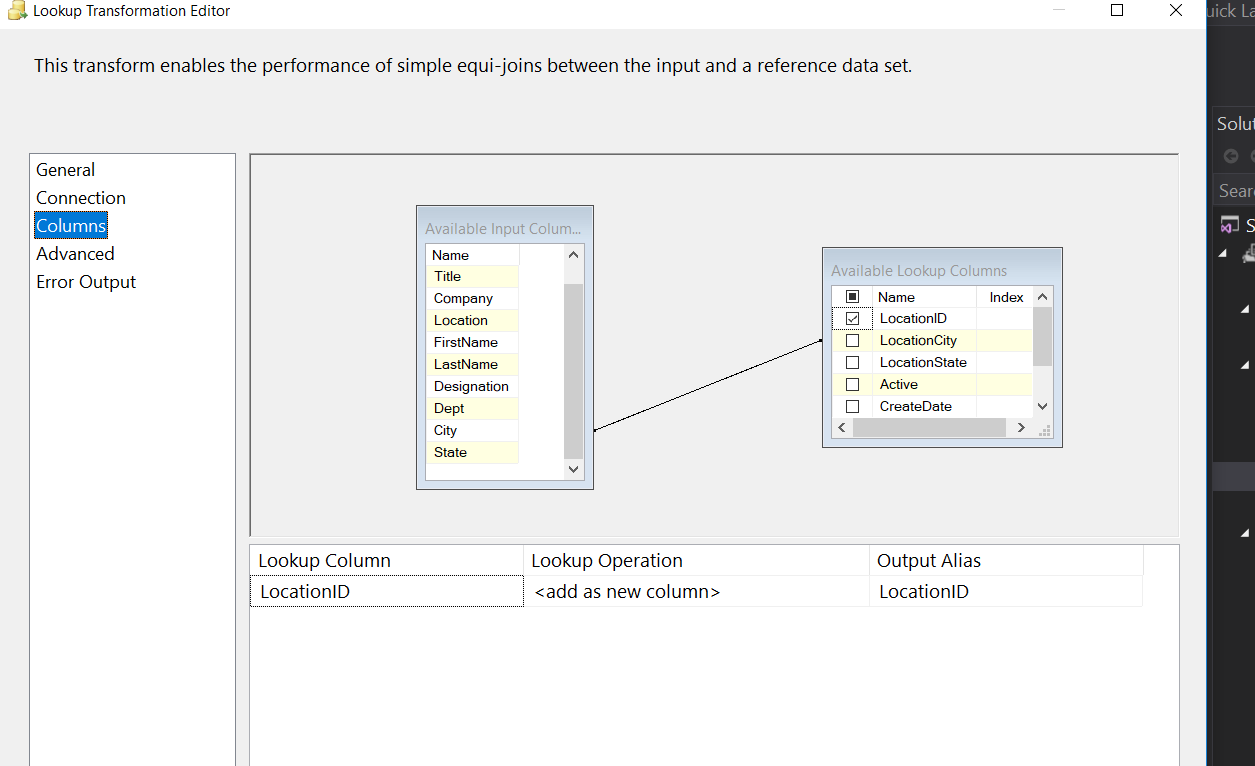
We need source as Prospect table from sql. So we can use ole db source for prospect table with all available columns.



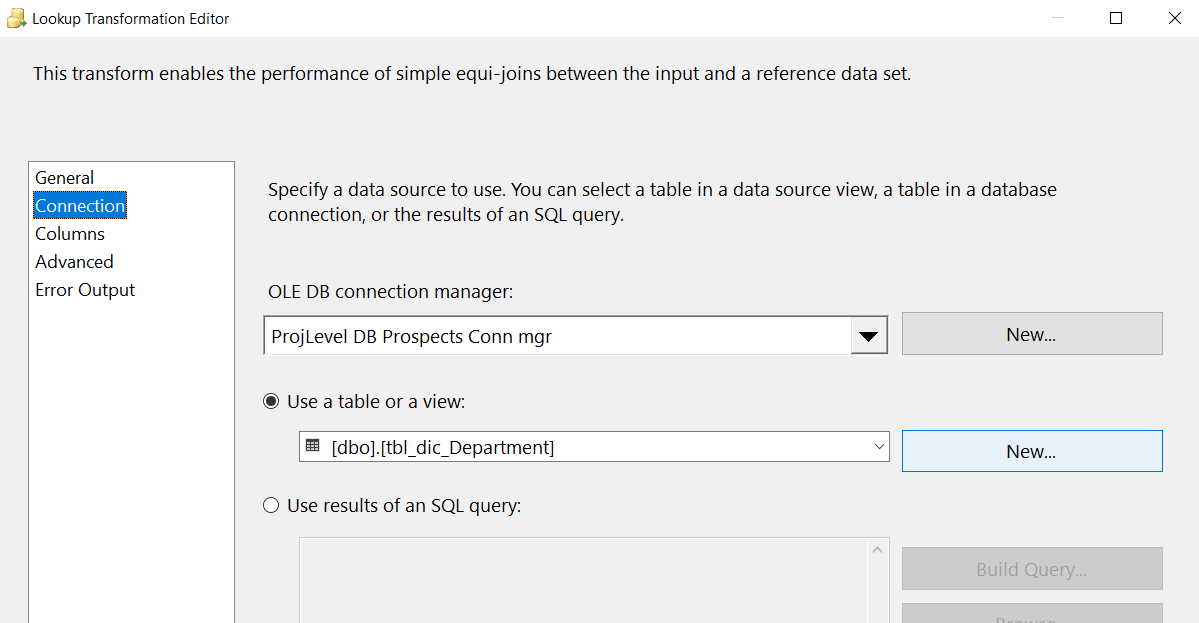
We can use lookup to get different columns from dictionary tables. So each look up for each dictionary tables with above setting in general tab.



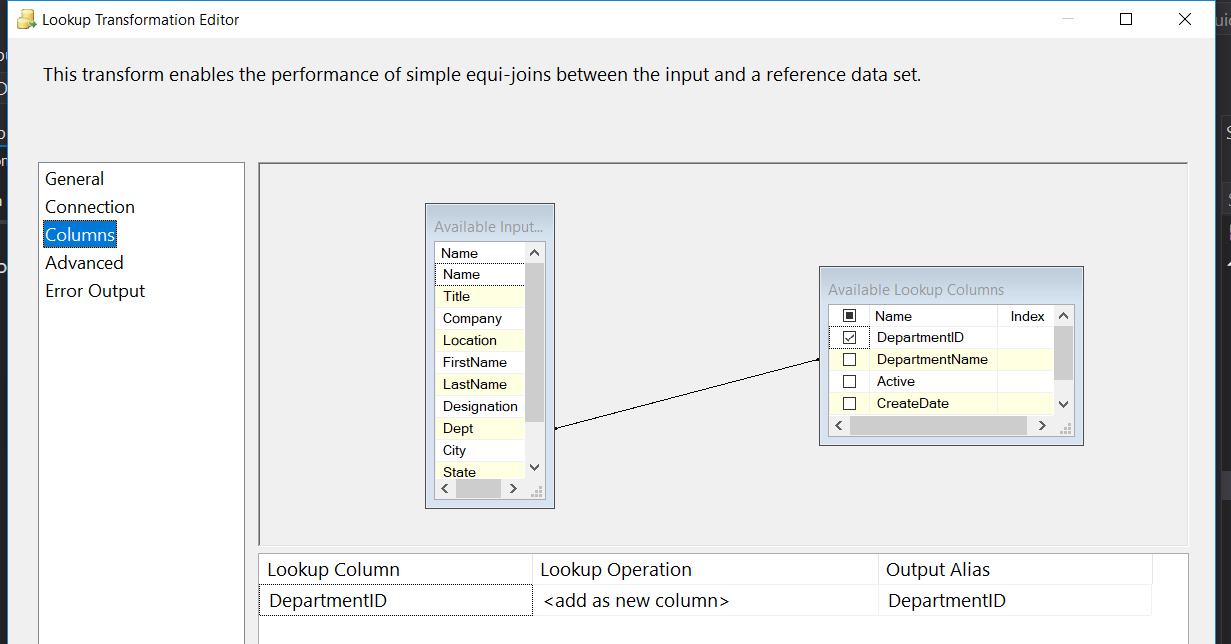
Above image is for lookup transformation where we can join between location table and prospect table with location city as lookup column.



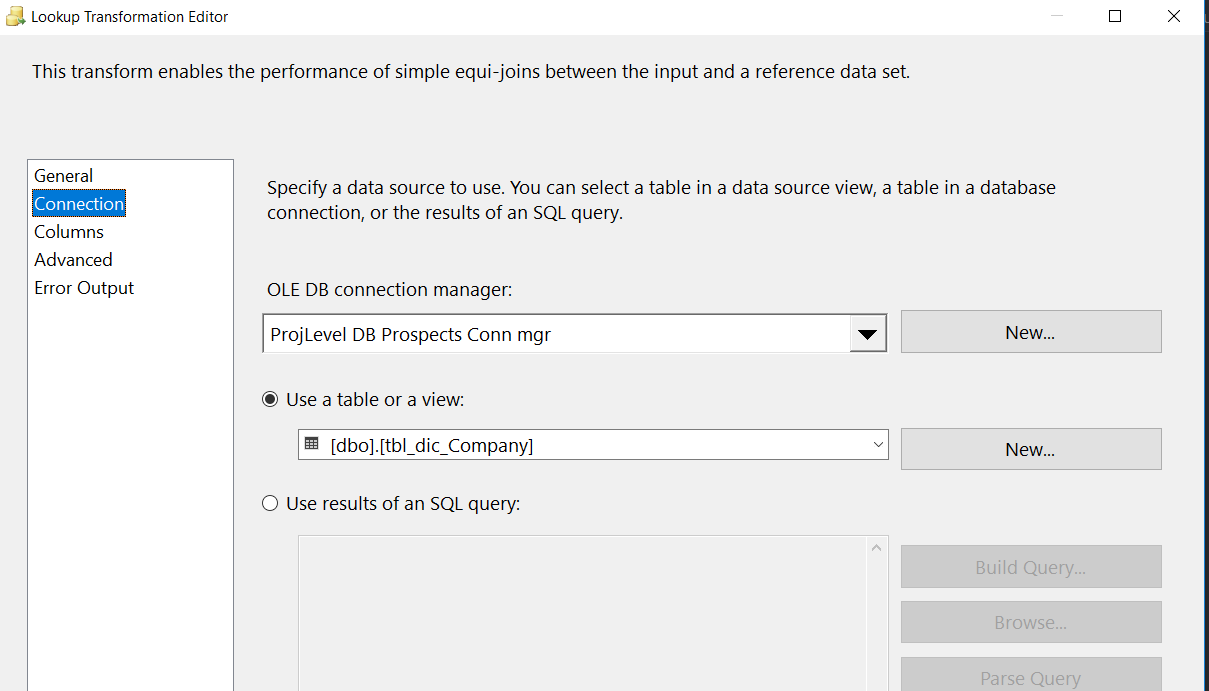
Above image shows that we using dictionary\_Location table’s locationCity column to join with prospect table columns and fetch LocationID column.



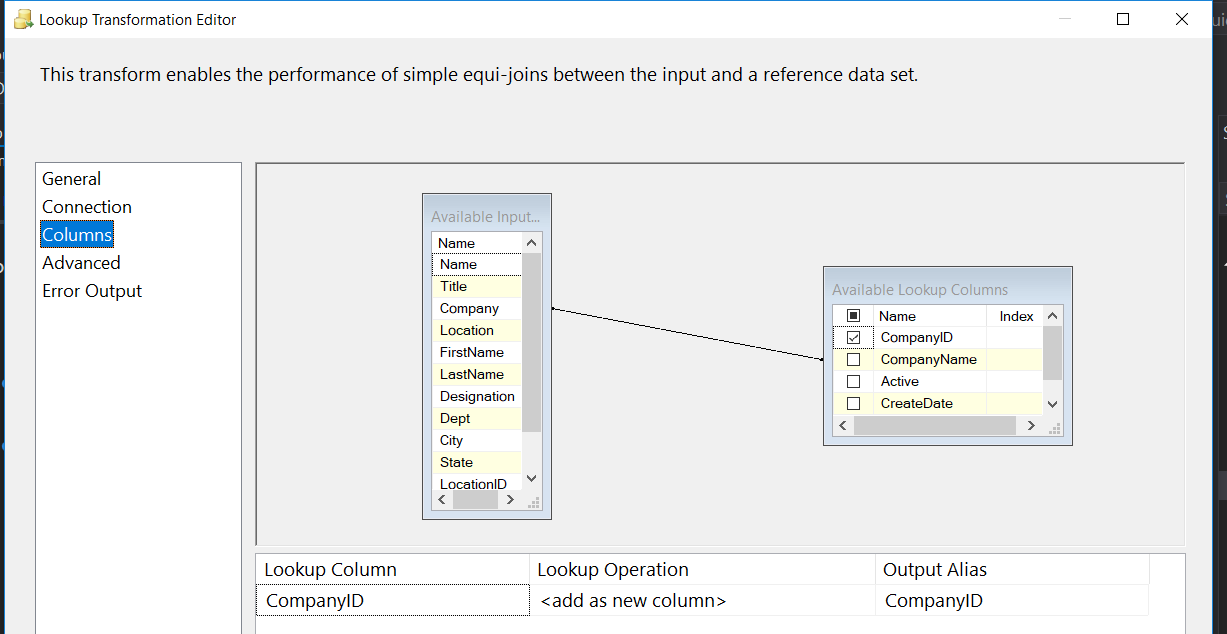
Above image is for lookup transformation where we can join between department table and prospect table with departmentName as lookup column.



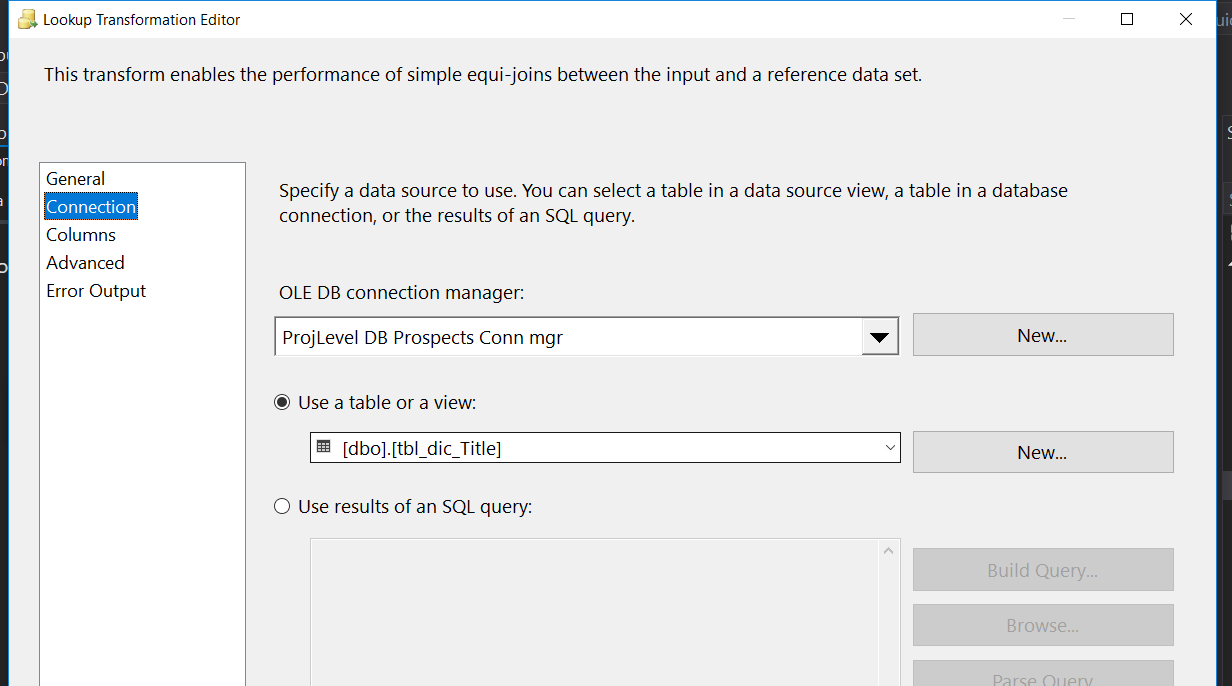
Above image shows that we using dictionary\_Department table’s departmentName column to join with prospect table columns and fetch DepartmentID from it.



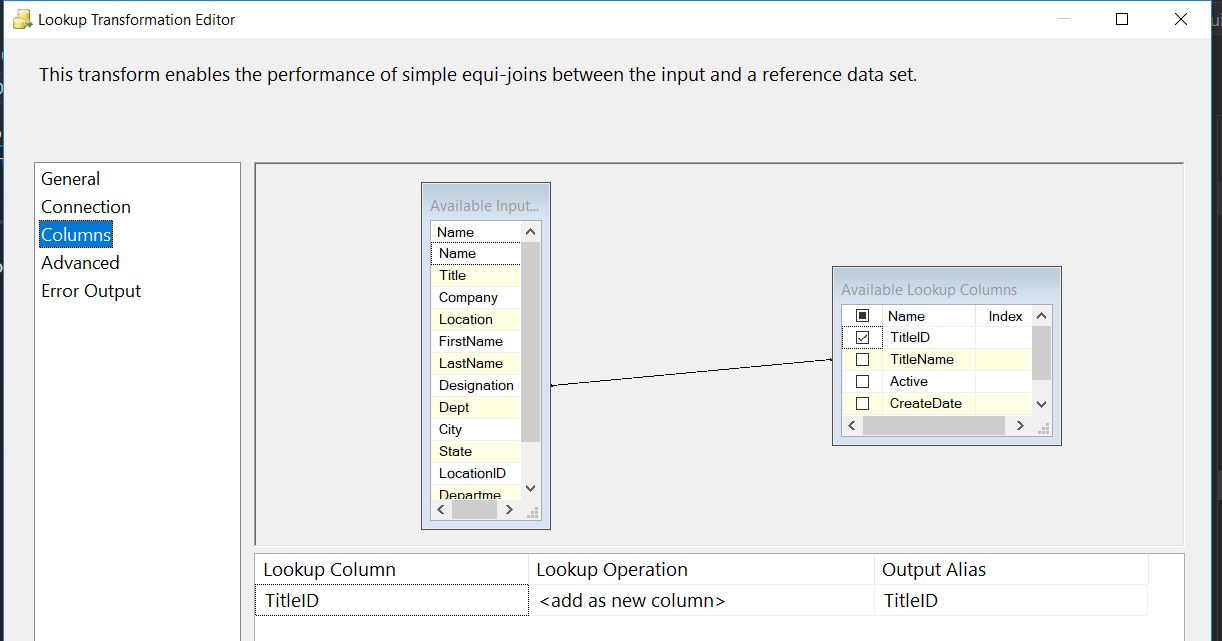
Above image is for lookup transformation where we can join between Company table and prospect table with CompanyName as lookup column.



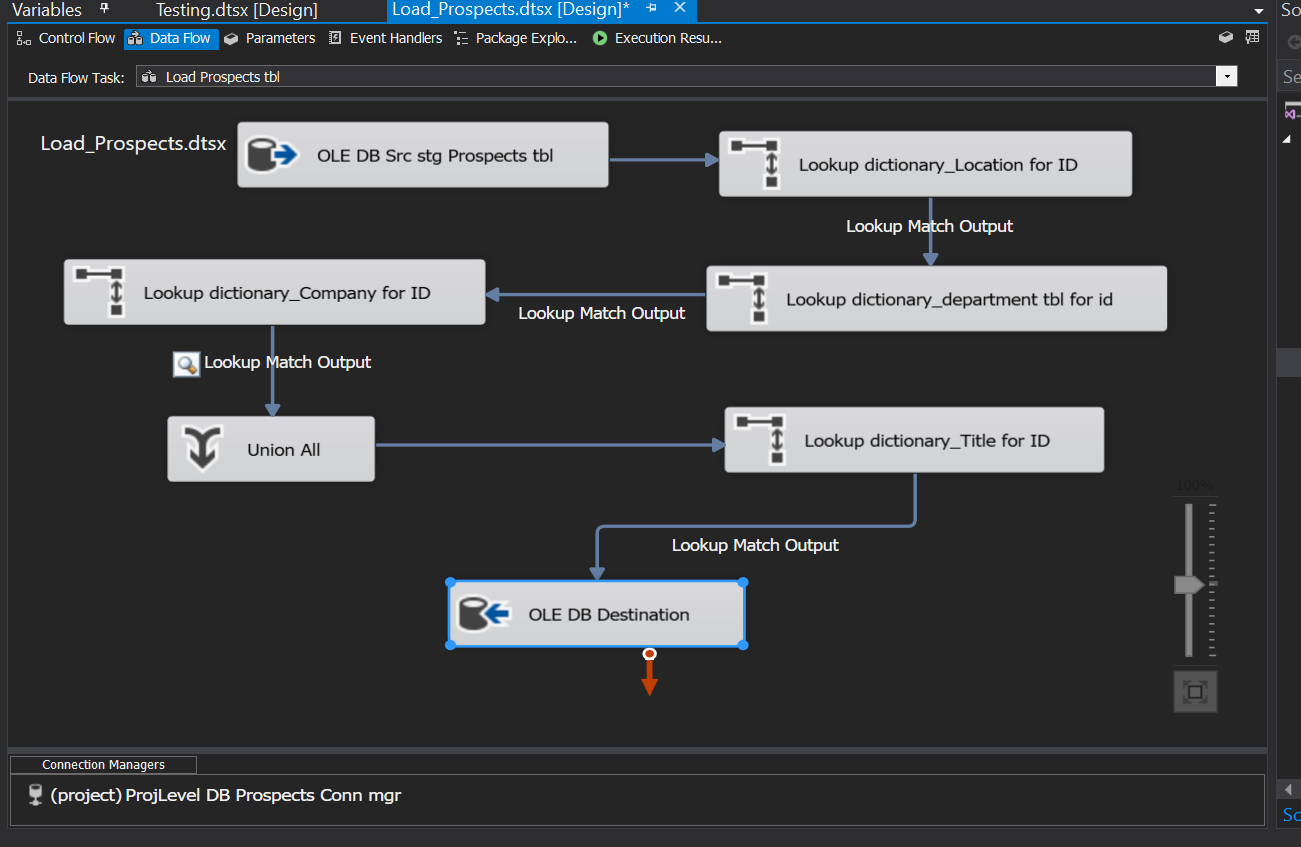
Above image shows that we using dictionary\_ Company table’s CompanyName column to join with prospect table columns and fetch CompanyID column.



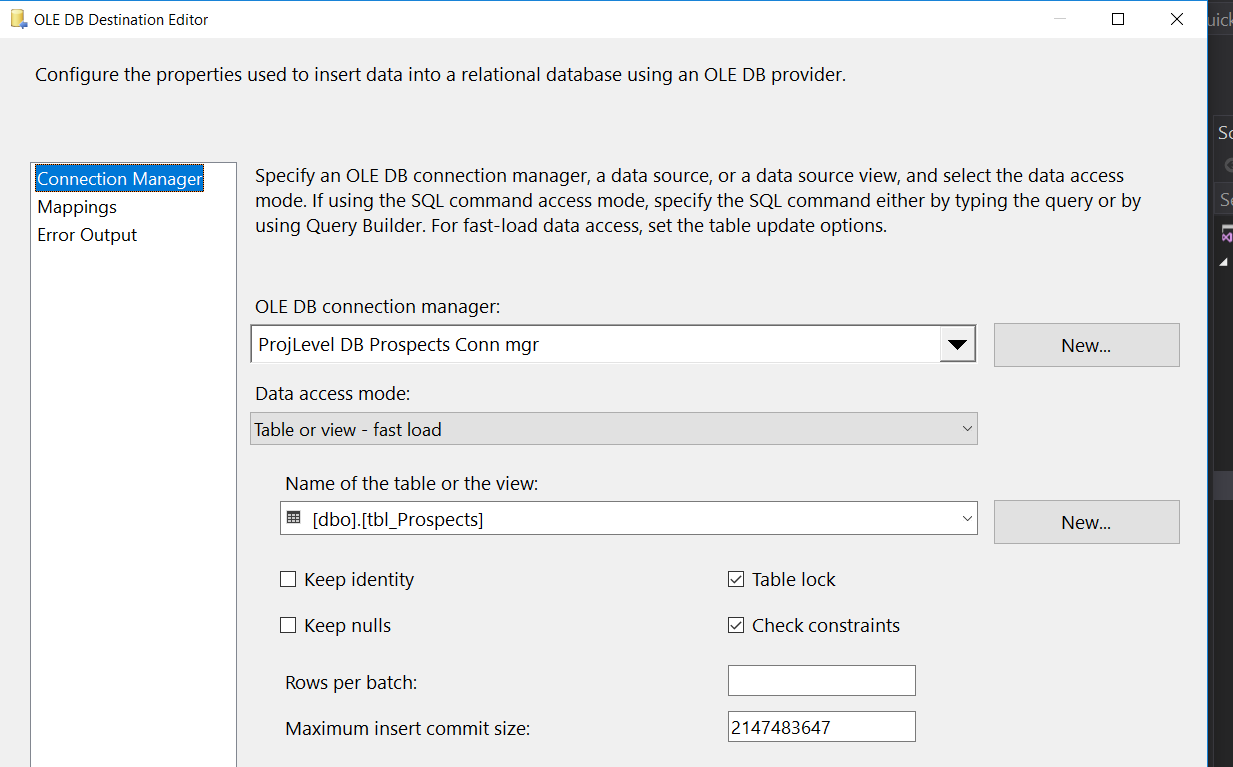
Above image is for lookup transformation where we can join between Title table and prospect table with Designation / TitleName as lookup column.



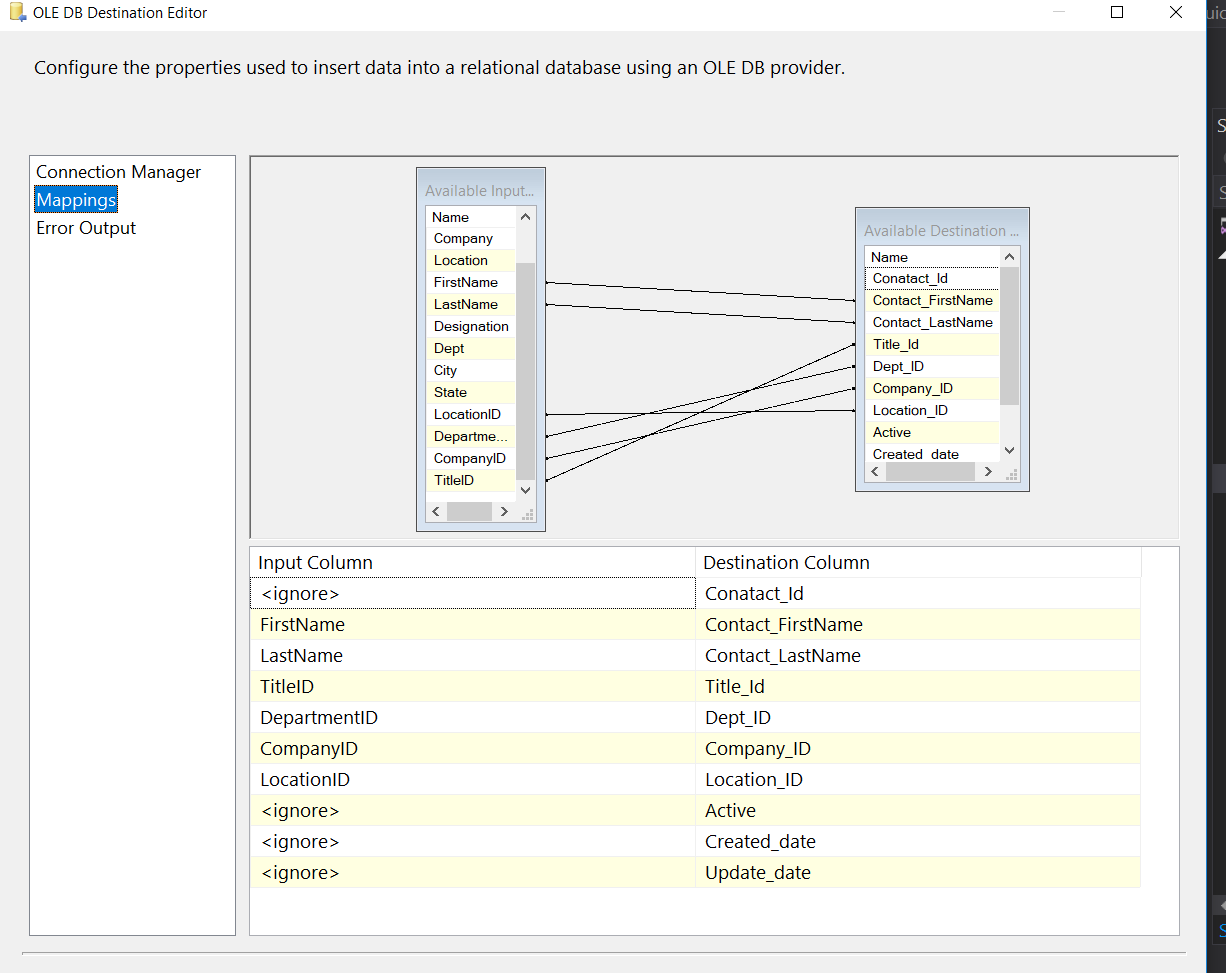
Above image shows that we using dictionary\_ Title table’s TitleName column to join with prospect table’s Designation column and fetch TitleID column.



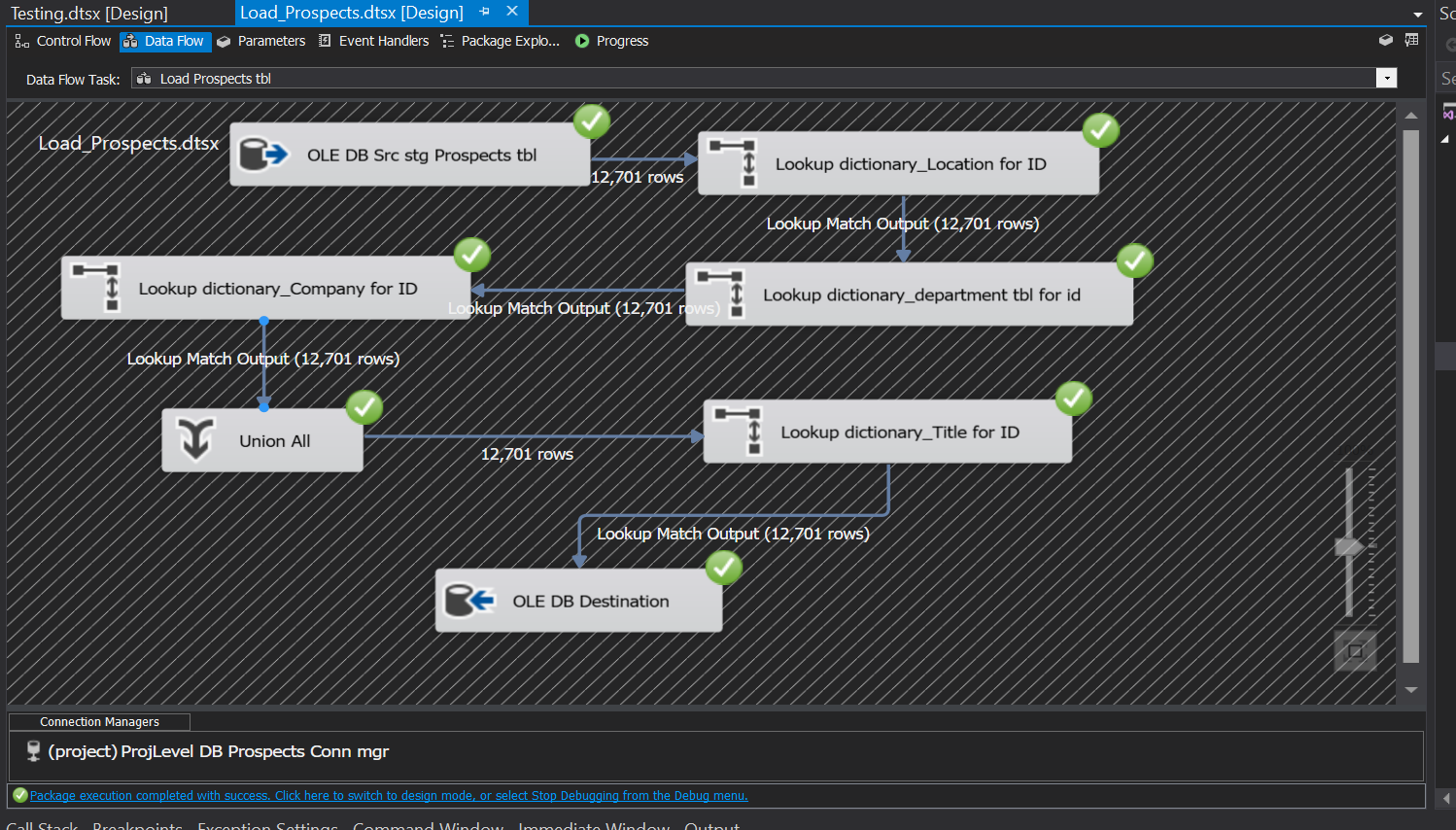
After all Lookup and union All we can load all data into Prospects Table with ole db destination.



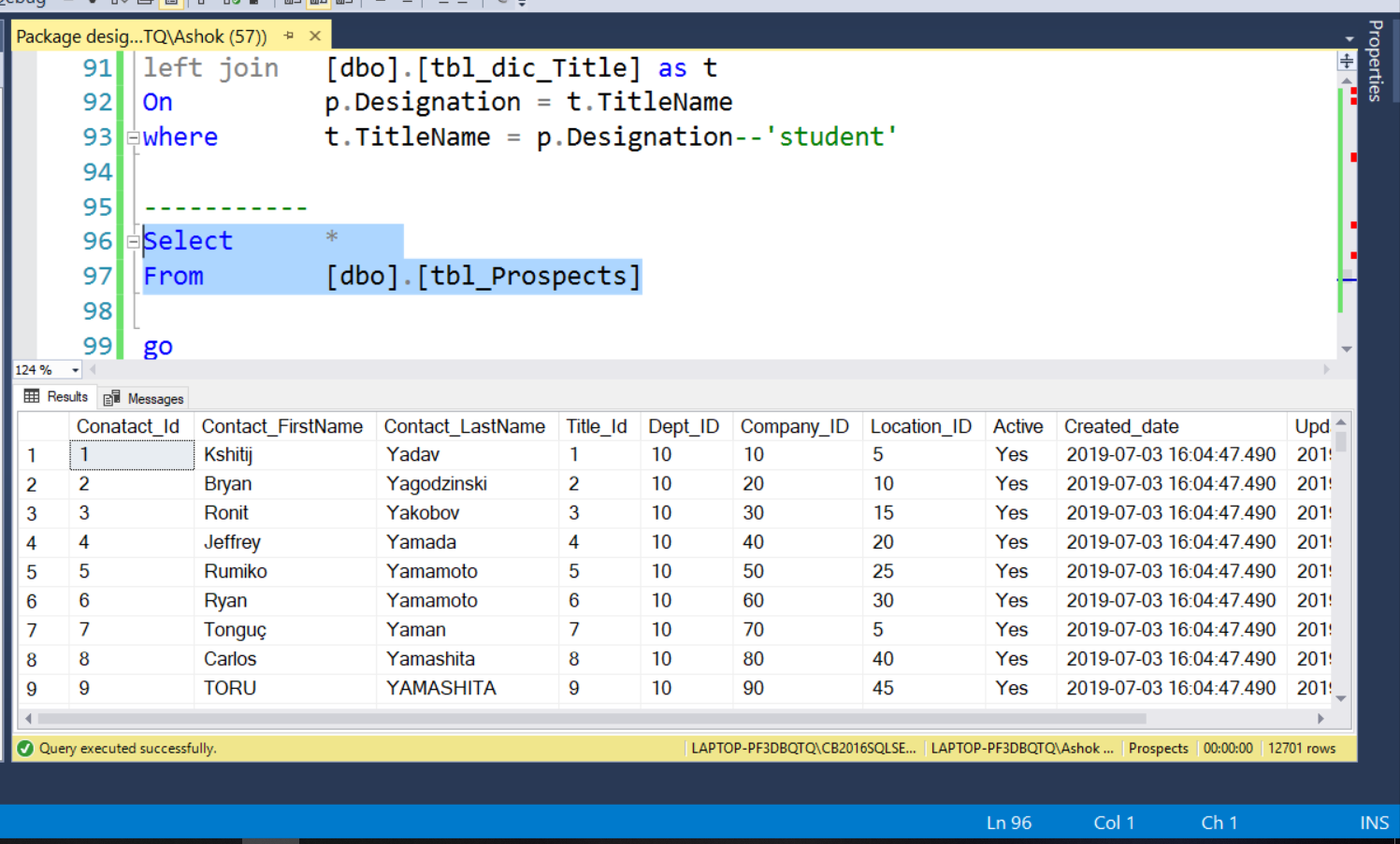
Above OLE db destination editor for Prospects table connection as shown in image.



Mapping for Prospects table with up stream/input columns as above image.



Above is the execution of package with successful data flow process.



Above is the final output of tbl\_Prospects table with loaded data for each column from SSIS package.