**SSIS Process.**

Planning an ETL process in a project is a very important step for the project to succeed.

Our thinking as a team immediately helped overcome any issues that may arise in the ETL process right at the beginning of the project.

In this project we used Microsoft tools when the main tool was SSDT.

ETL process map and design: Mirror, Stage and Data Warehouse.

A Data Warehouse splits into two processes: dimensions and fact.

Helps to understand the process more cleanly and More so, preventing future glitches.

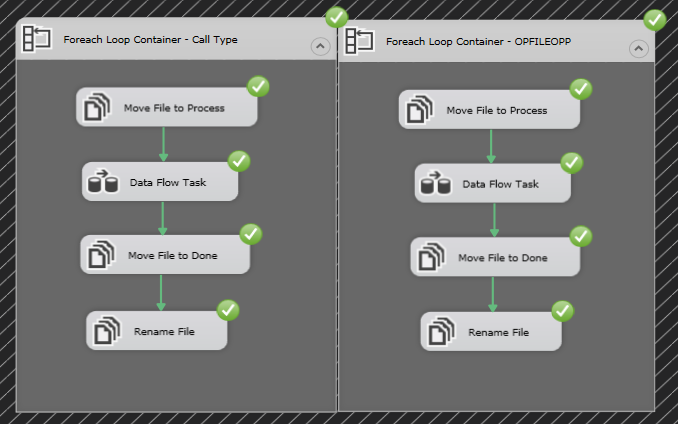
For the ETL process to be correct, the process must be applied according to the conventional methodology.

1. The ETL process should work daily by the "Job" command built into the SSMS software after the ETL process completed.

2. A complete "data cleaning" process for this project has been completed.

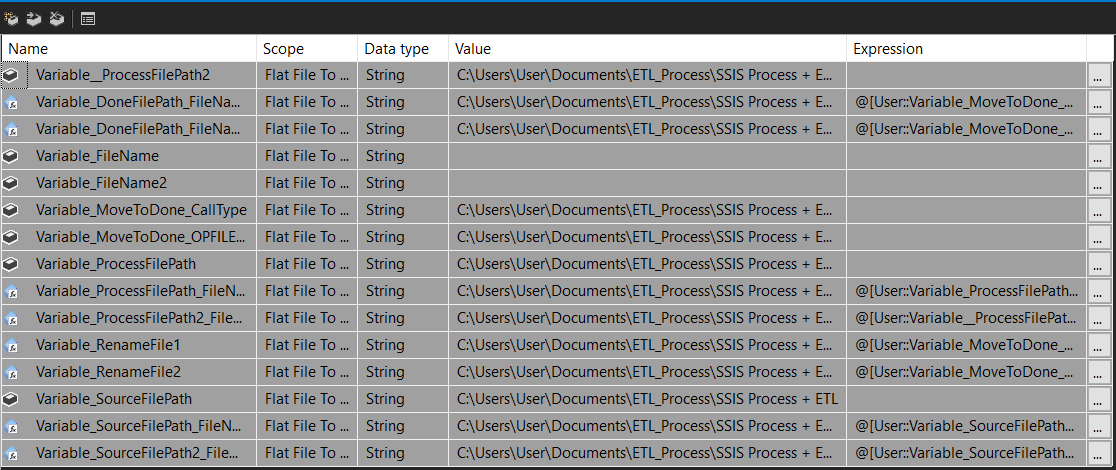
The process began to load CSV files dynamically.

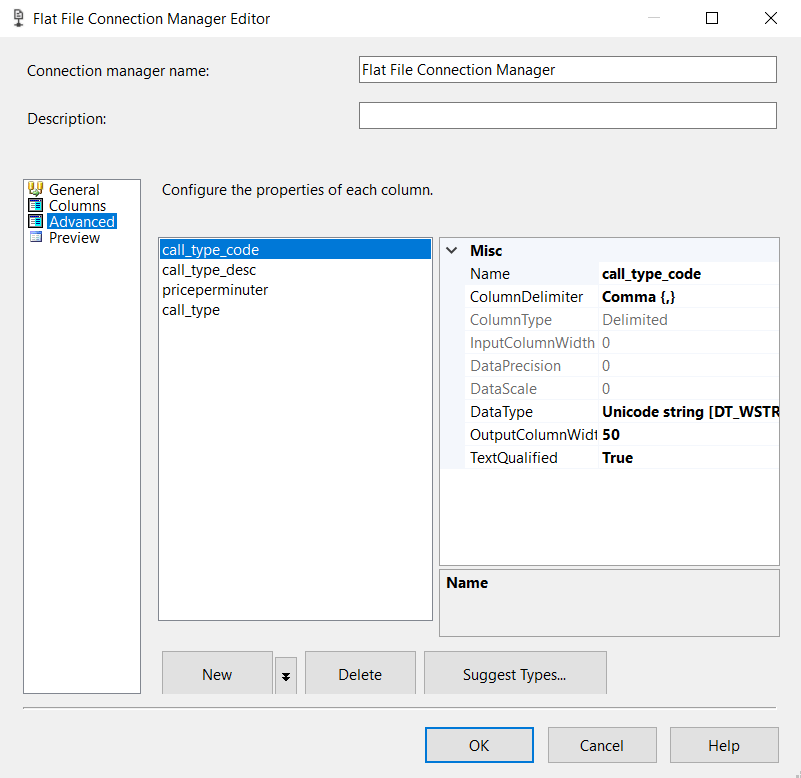
First, as a team we had to build two tables in SSMS to load the data: "OPFILEOPP" and "Call Type"



**Process ETL – Step 1 - Dynamically upload CSV files:**

**Process ETL – The Variables of process Dynamically upload:**



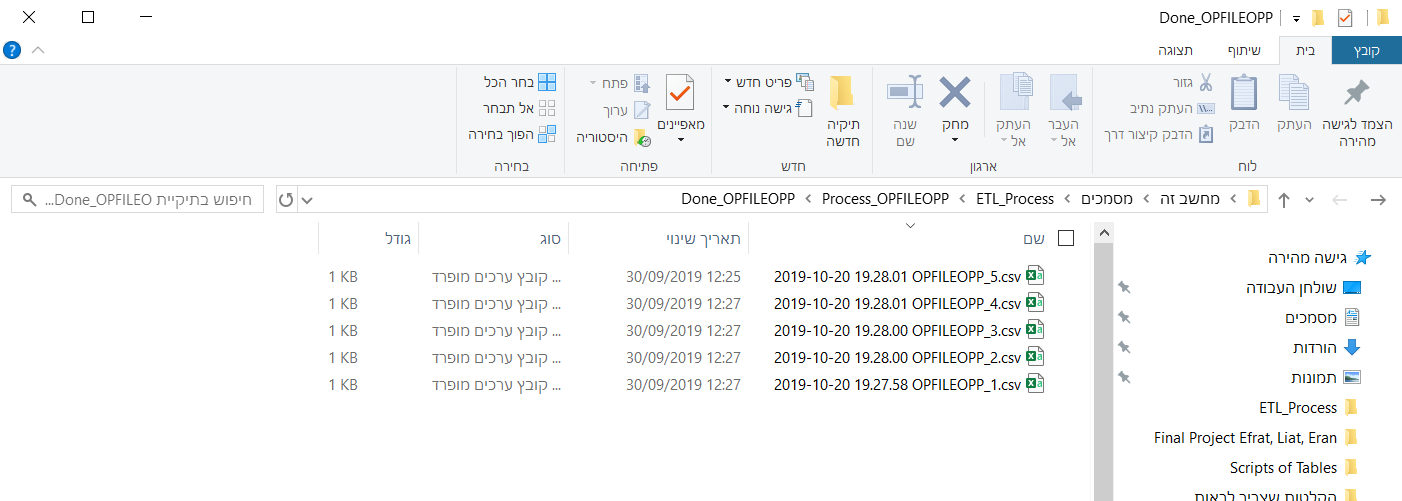
In editing of data in Connection Manager

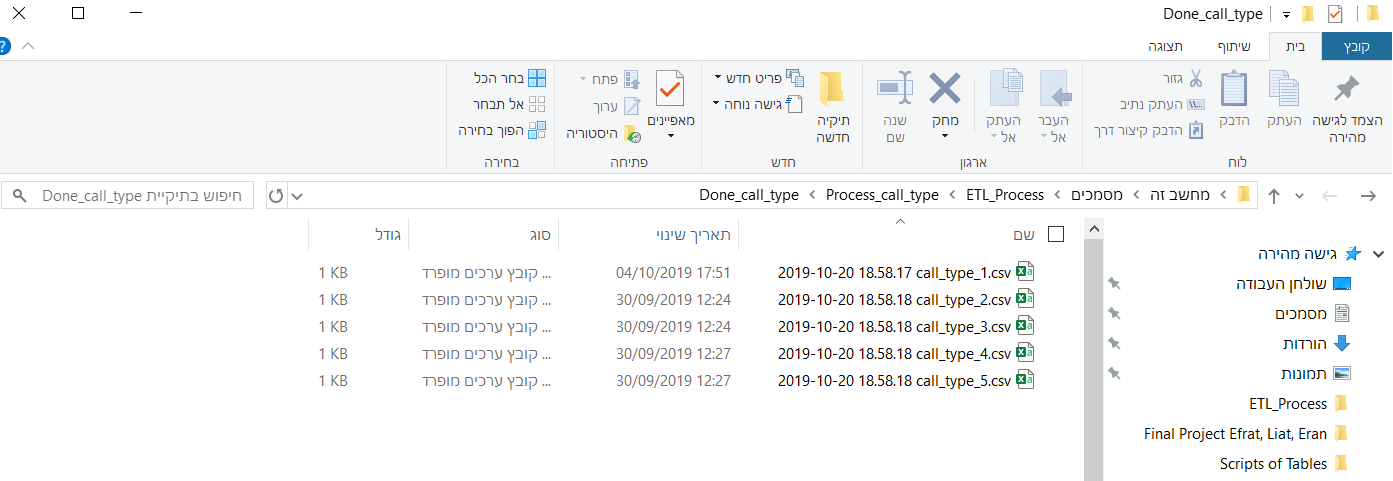
We are change Data Types of the data.

For the loading process to work.

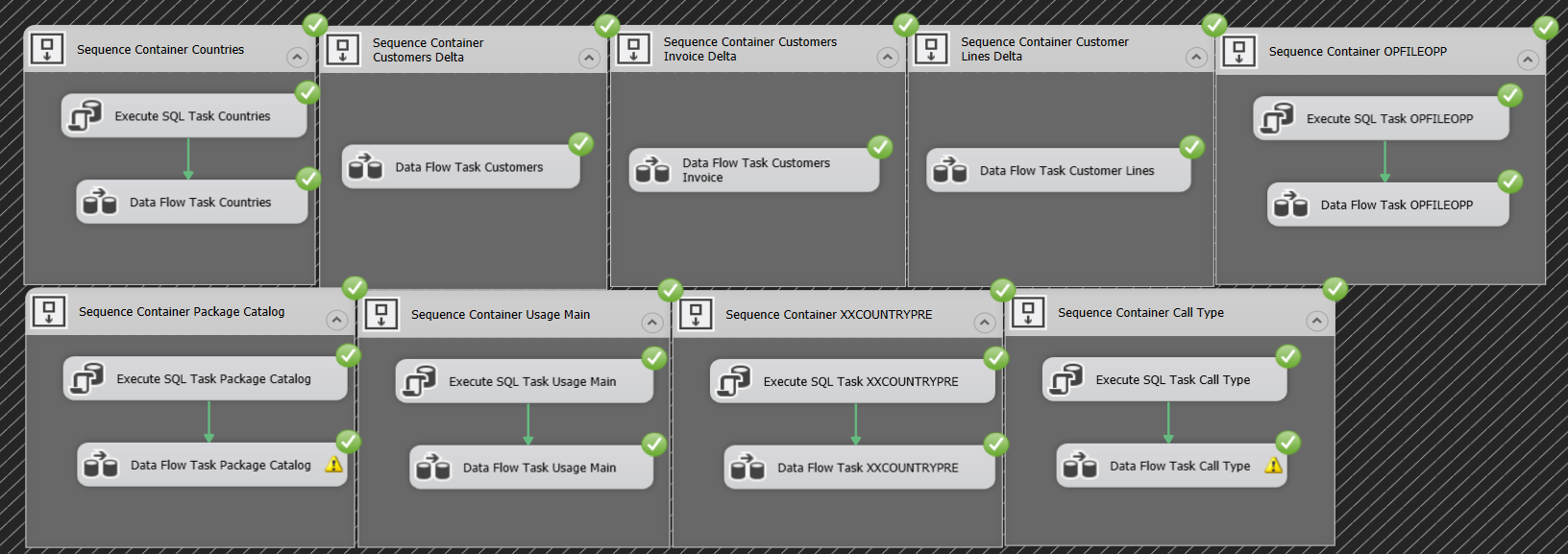
The Result of Process:

**Upload CSV files:**



OPFILEOPP

Call\_Type

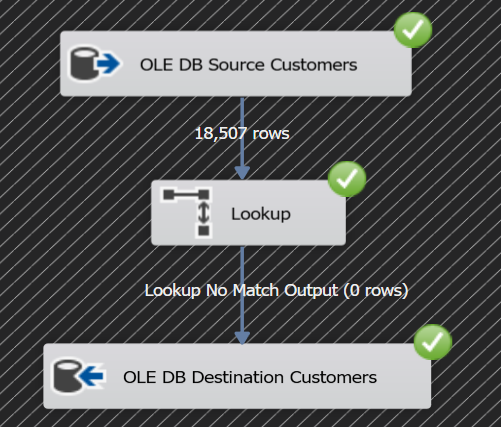
**Process ETL – Step 2 - Operation to Mirror:**

At this point, the goal is to simply copy the data itself from the data as it is.

Designing the data loading process from the Operation step to Mirror step.

We chose to display the two options available at each ETL stage (depending on the amount of company data and data changes).

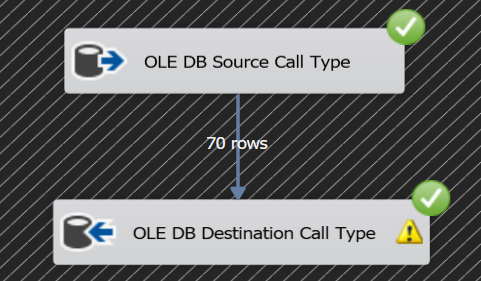
For this project we have selected three tables where data will be transferred in Delta of Data. (Customers, Customers Invoice and Customer Lines).

Process ETL – Transfer only Delta of data - For example we show the Customers Table Only Delta:

This process is faster, and the amount of data transmitted at a time is partial.

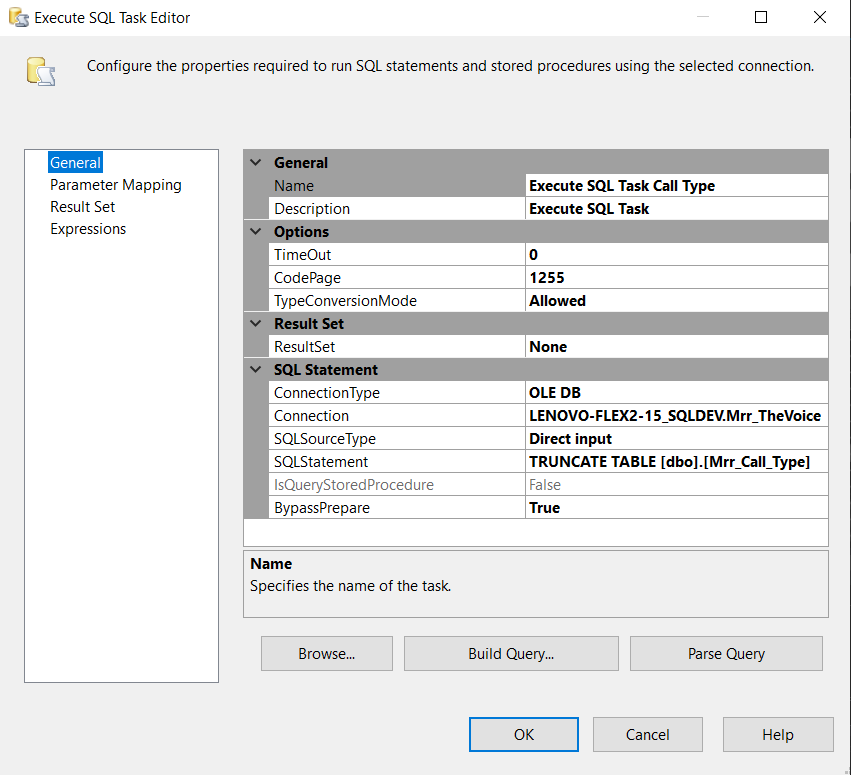
It has advantages, but also some disadvantages.

The second option is to perform truncate every time the process is started:



This action affects the performances of the servers, but this is the practice in the BI world.

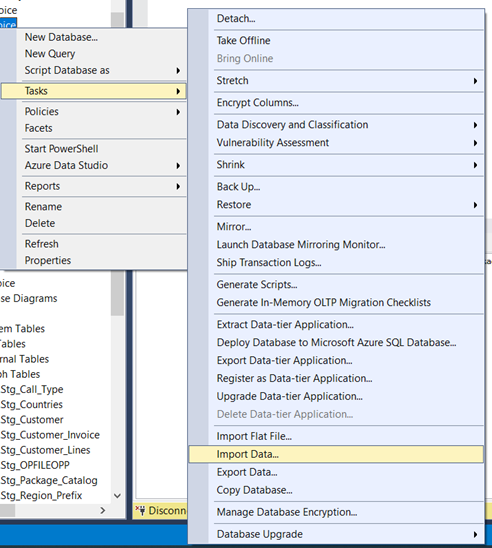
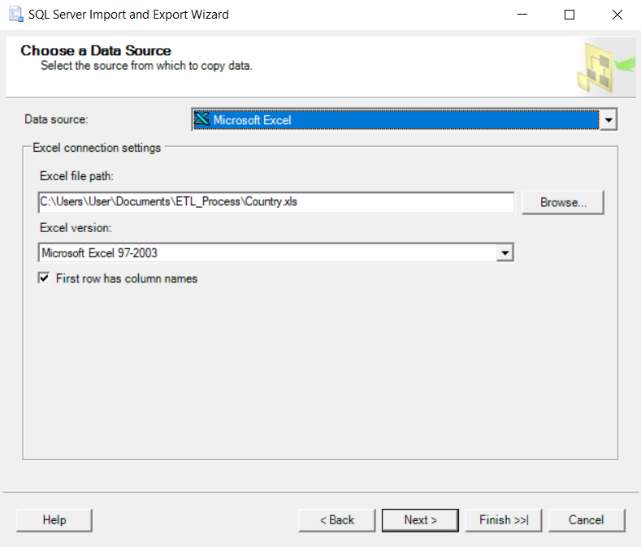
The action truncate done in "SSIS Toolbox "Execute SQL Task Editor".

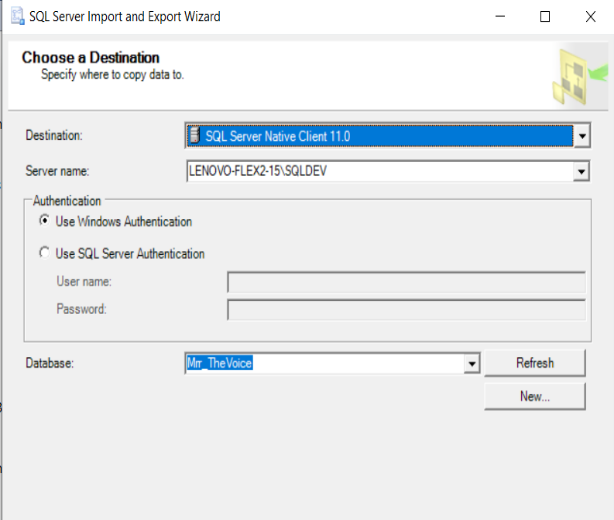
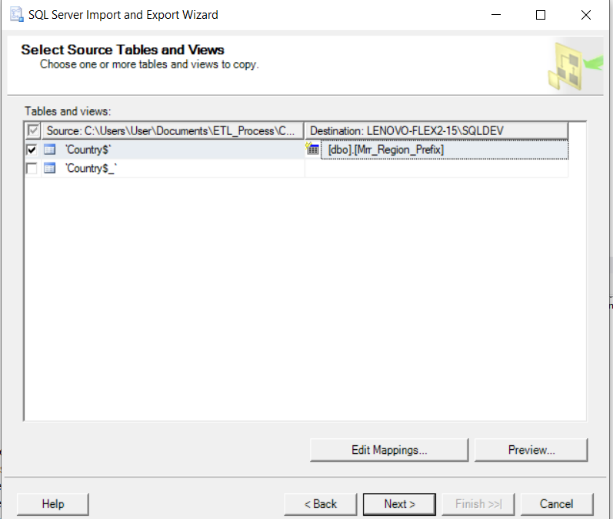


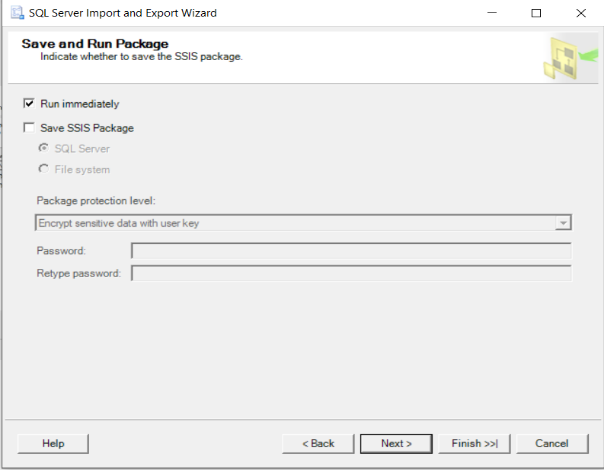
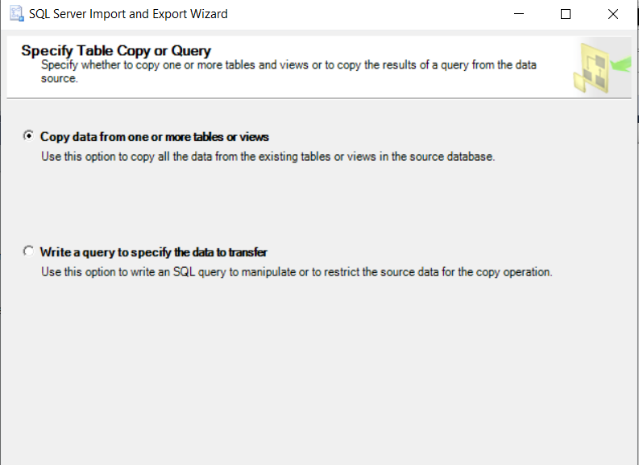
In this action we notice a connection that is correct and In the SQL statement, we write the command we want to execute.

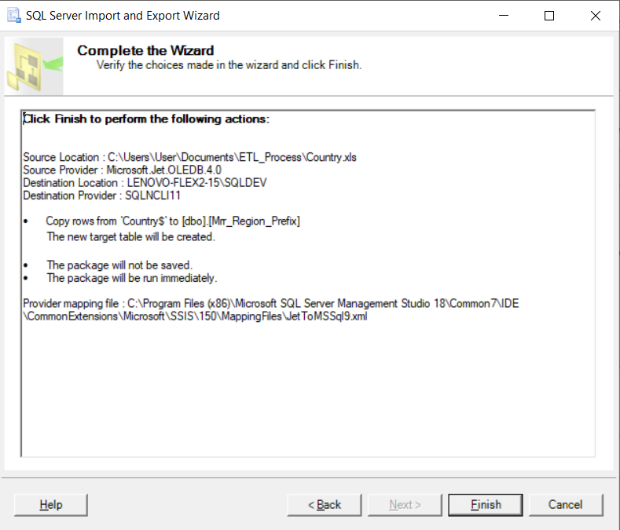
As we already noticed as a team in the initial stage of the ETL process, we chose to import another table into the mirror stage.

**Explain how uploading Excel to SSMS: Table Region Prefix:**

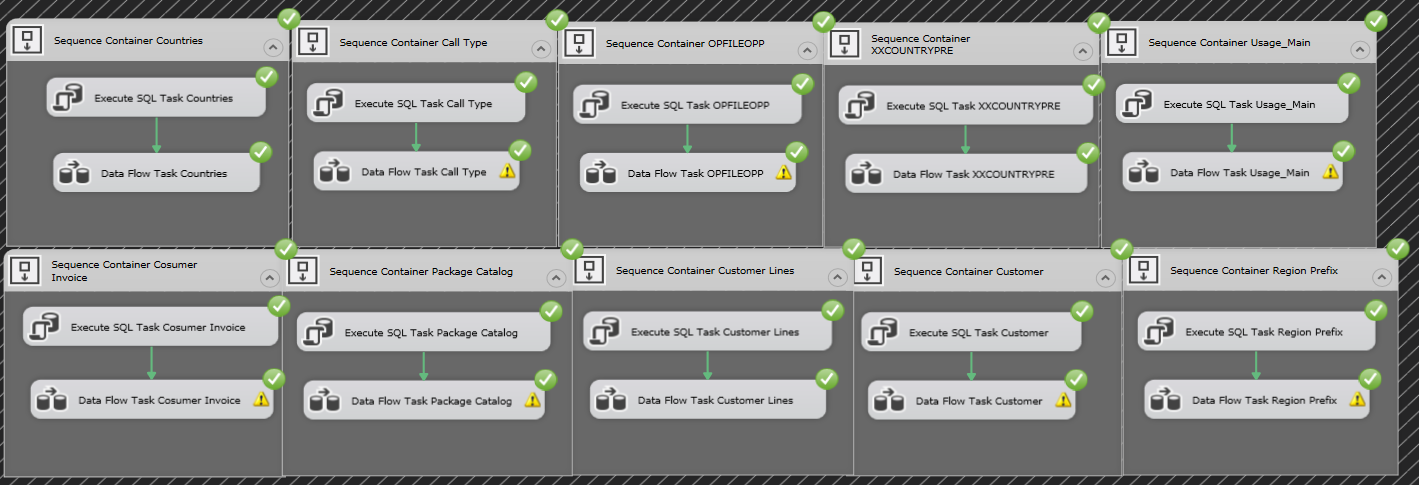
1. 2.

3. 4.

**5. 6.

7.

The reason we made another table import is to avoid errors in the dimension step.

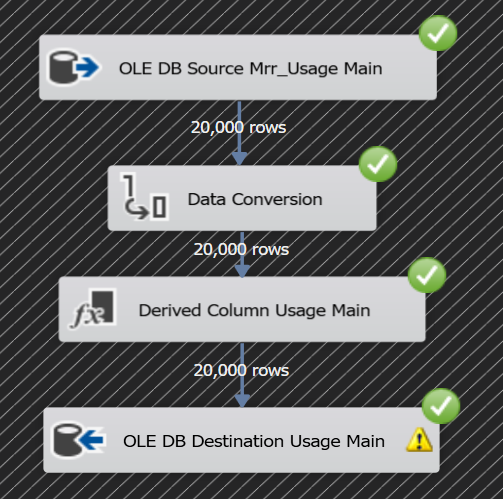
**Process ETL – Step 3 - Mirror to Stage:**

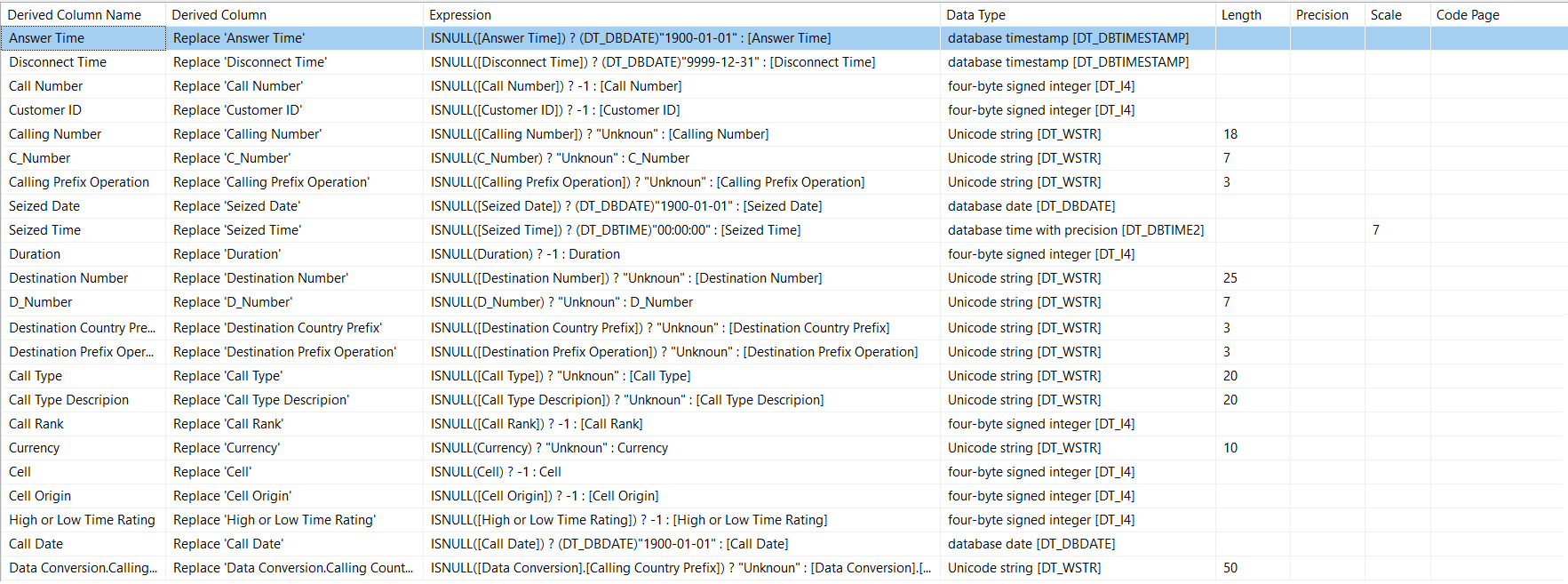
At this stage of the ETL process, we selected a team to do the Truncate process for all tables.

The most important part of this step is handling the "Null" values and "Errors handling".

In this stage we are have sifts the data and added calculated columns in "SSIS Expression" language and "SQL Commend".

Process ETL – Table errors handling – For example we show the Usage Main Table:



Derived Column - Usage Main Table:

In the "Usage Main" Table we can see how we handled the "Null" values.

In course we learned how to handle errors, there are three options for error handling.

1) Character Field Data – for example here – 'Calling Number'

The way of treatment – write "SSIS Expression" with Boolean conditions.

ISNULL([Calling Number]) ? "Unknoun" : [Calling Number]

2) Numerical Data - for example here – 'Duration'

The way of treatment – write "SSIS Expression" with Boolean conditions.

ISNULL(Duration) ? -1 : Duration

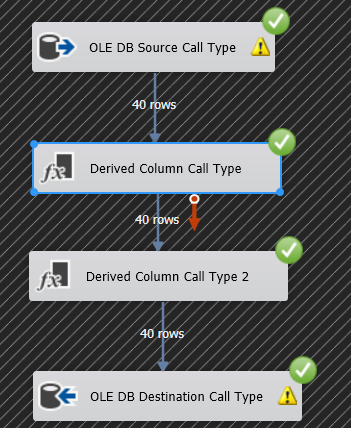
3) Data Dates - for example here – 'Seized Date'

The way of treatment – write "SSIS Expression" with Boolean conditions.

ISNULL([Seized Date]) ? (DT\_DBDATE)"1900-01-01" : [Seized Dat

There is another number of write "SSIS Expression" in project:

For example, in "Call Type" Table.

First, in the case of this table, we wrote a "SQL command" that executes the "Distinct" operation this is to avoid duplications. The command writes within "OLE DB Source Call Type".

Here you can see two commands to Derived column.

The reason:

"SSIS Expression" command within first Derived column.

In this Derived column we can see that we have added a calculated column.

The second Derived column Used to prevent "Null" values ​​from passing the calculated column.

Before this process, 70 rows passed and only 40 rows passed after the process.[[1]](#footnote-1)

These are scripts for example that we wrote in SQL language to make the data transition from stage Mirror to Stage.

**Scripts of Process Mirror to Stage:**

--Mrr\_OPFILEOPP

select Distinct\*

from [dbo].[Mrr\_OPFILEOPP]

--Mrr\_Call\_Type

select distinct \*

from [dbo].[Mrr\_Call\_Type]

--Mrr\_Customer

select \*

from

(select \*, right(CUST\_NUMBER,7) 'Number', case when CUST\_NUMBER like '+\_\_\_%' then SUBSTRING(CUST\_NUMBER,2,3) end 'Prefix country',

case when CUST\_NUMBER like '+\_\_\_%' then SUBSTRING(CUST\_NUMBER,5,2) end 'Prefix Operation'

from Mrr\_Customer

where CUST\_NUMBER like '+972%'

Union all

select \*, right(CUST\_NUMBER,7) 'Number', case when CUST\_NUMBER like '+\_%' then SUBSTRING(CUST\_NUMBER,2,1) end 'Prefix country',

case when CUST\_NUMBER like '+\_%' then SUBSTRING(CUST\_NUMBER,3,3) end 'Prefix Operation'

from Mrr\_Customer C

where CUST\_NUMBER like '+1%') tab

order by customer\_id

--Mrr\_Customer\_Invoice

select \*, right(PHONE\_NO,7) 'Number', case when PHONE\_NO like '+\_\_\_%' then SUBSTRING(PHONE\_NO,2,3) end 'Prefix country', case when PHONE\_NO like '+\_\_\_%' then SUBSTRING(PHONE\_NO,5,2) end 'Prefix Operation'

from Mrr\_Customer\_Invoice

order by INVOICE\_NUM

--Mrr\_Customer\_Lines

select \*, right(PHONE\_NO,7) 'Number', case when PHONE\_NO like '+\_\_\_%' then SUBSTRING(PHONE\_NO,2,3) end 'Prefix country', case when PHONE\_NO like '+\_\_\_%' then SUBSTRING(PHONE\_NO,5,2) end 'Prefix Operation'

from Mrr\_Customer\_Lines

where PHONE\_NO like '+972%'

Union all

select \*, right(PHONE\_NO,7) 'Number', case when PHONE\_NO like '+\_%' then SUBSTRING(PHONE\_NO,2,1) end 'Prefix country',

case when PHONE\_NO like '+\_%' then SUBSTRING(PHONE\_NO,3,3) end 'Prefix Operation'

from Mrr\_Customer\_Lines

where PHONE\_NO like '+1%'

--Mrr\_Usage\_Main

select \*

from

(select CALL\_NO as [Call Number], CUST\_ID as [Customer ID], CALLING\_NO as [Calling Number], right(CALLING\_NO,7) as [C\_Number], case when CALLING\_NO like '+\_\_\_%' then SUBSTRING(CALLING\_NO,2,3) end [Calling Country Prefix], case when CALLING\_NO like '+\_\_\_%' then SUBSTRING(CALLING\_NO,5,2) end [Calling Prefix Operation], ANSWER\_TIME as [Answer Time], convert(date,SEIZED\_TIME) as [Seized Date], convert(time,(convert(smalldatetime,SEIZED\_TIME))) as [Seized Time], DISCONNECT\_TIME as [Disconnect Time], DURATION as [Duration], DES\_NO as [Destination Number], right(DES\_NO,7) as [D\_Number], case when DES\_NO like '+\_\_\_%' then SUBSTRING(DES\_NO,2,3) end [Destination Country Prefix], case when DES\_NO like '+\_\_\_%' then SUBSTRING(DES\_NO,5,2) end [Destination Prefix Operation], CALL\_TYPE as [Call Type], PROD\_TYPE as [Call Type Descripion], RATED\_AMNT as [Call Rank], RATED\_CURR\_CODE as [Currency], CELL as [Cell], CELL\_ORIGIN as [Cell Origin], HIGH\_LOW\_RATE as [High or Low Time Rating], convert(date,CALL\_DATETIME) as [Call Date]

from Mrr\_Usage\_Main

where CALLING\_NO like '+972%' and DES\_NO like '+972%'

union all

select CALL\_NO as [Call Number], CUST\_ID as [Customer ID], CALLING\_NO as [Calling Number], right(CALLING\_NO,7) as [C\_Number], case when CALLING\_NO like '+\_%' then SUBSTRING(CALLING\_NO,2,1) end [Calling Country Prefix], case when CALLING\_NO like '+\_%' then SUBSTRING(CALLING\_NO,3,3) end [Calling Prefix Operation], ANSWER\_TIME as [Answer Time], convert(date,SEIZED\_TIME) as [Seized Date], convert(time,(convert(smalldatetime,SEIZED\_TIME))) as [Seized Time], DISCONNECT\_TIME as [Disconnect Time], DURATION as [Duration], DES\_NO as [Destination Number], right(DES\_NO,7) as [D\_Number], case when DES\_NO like '+\_\_\_%' then SUBSTRING(DES\_NO,2,3) end [Destination Country Prefix], case when DES\_NO like '+\_\_\_%' then SUBSTRING(DES\_NO,5,2) end [Destination Prefix Operation], CALL\_TYPE as [Call Type], PROD\_TYPE as [Call Type Descripion], RATED\_AMNT as [Call Rank], RATED\_CURR\_CODE as [Currency], CELL as [Cell], CELL\_ORIGIN as [Cell Origin], HIGH\_LOW\_RATE as [High or Low Time Rating], convert(date,CALL\_DATETIME) as [Call Date]

from Mrr\_Usage\_Main

where CALLING\_NO like '+1%' and DES\_NO like '+972%'

union all

select CALL\_NO as [Call Number], CUST\_ID as [Customer ID], CALLING\_NO as [Calling Number], right(CALLING\_NO,7) as [C\_Number], case when CALLING\_NO like '+\_\_\_%' then SUBSTRING(CALLING\_NO,2,3) end [Calling Country Prefix], case when CALLING\_NO like '+\_\_\_%' then SUBSTRING(CALLING\_NO,5,2) end [Calling Prefix Operation], ANSWER\_TIME as [Answer Time], convert(date,SEIZED\_TIME) as [Seized Date], convert(time,(convert(smalldatetime,SEIZED\_TIME))) as [Seized Time], DISCONNECT\_TIME as [Disconnect Time], DURATION as [Duration], DES\_NO as [Destination Number], right(DES\_NO,7) as [D\_Number], case when DES\_NO like '+\_%' then SUBSTRING(DES\_NO,2,1) end [Destination Country Prefix], case when DES\_NO like '+\_%' then SUBSTRING(DES\_NO,3,3) end [Destination Prefix Operation], CALL\_TYPE as [Call Type], PROD\_TYPE as [Call Type Descripion], RATED\_AMNT as [Call Rank], RATED\_CURR\_CODE as [Currency], CELL as [Cell], CELL\_ORIGIN as [Cell Origin], HIGH\_LOW\_RATE as [High or Low Time Rating], convert(date,CALL\_DATETIME) as [Call Date]

from Mrr\_Usage\_Main

where CALLING\_NO like '+972%' and DES\_NO like '+1%'

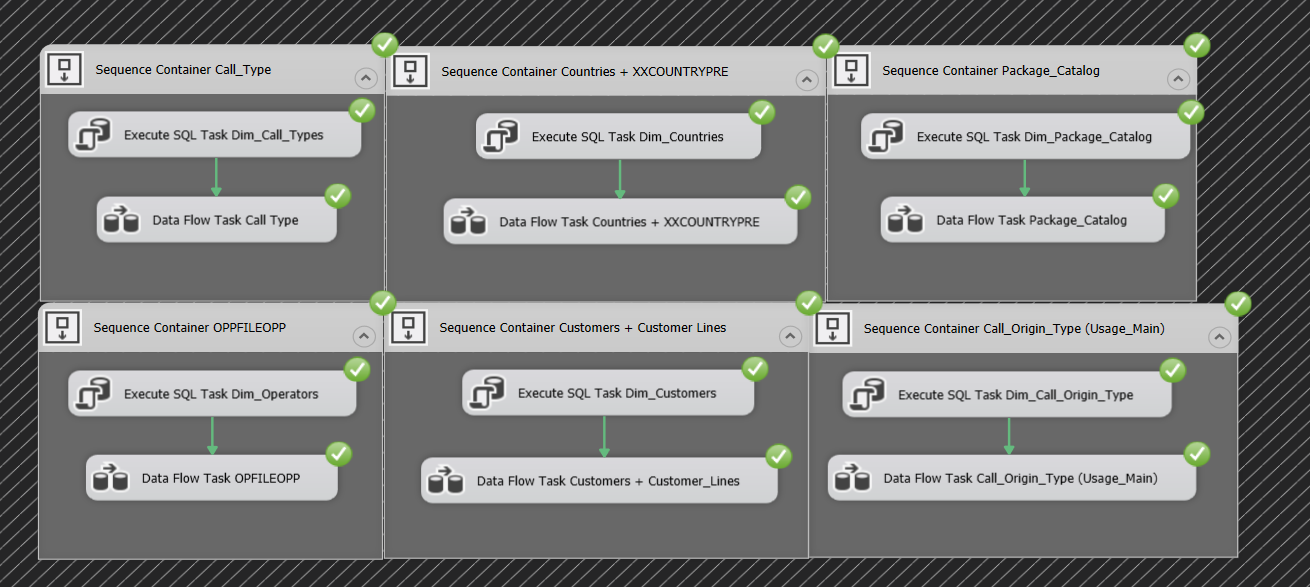
union all

select CALL\_NO as [Call Number], CUST\_ID as [Customer ID], CALLING\_NO as [Calling Number], right(CALLING\_NO,7) as [C\_Number], case when CALLING\_NO like '+\_%' then SUBSTRING(CALLING\_NO,2,1) end [Calling Country Prefix], case when CALLING\_NO like '+\_%' then SUBSTRING(CALLING\_NO,3,3) end [Calling Prefix Operation], ANSWER\_TIME as [Answer Time], convert(date,SEIZED\_TIME) as [Seized Date], convert(time,(convert(smalldatetime,SEIZED\_TIME))) as [Seized Time], DISCONNECT\_TIME as [Disconnect Time], DURATION as [Duration], DES\_NO as [Destination Number], right(DES\_NO,7) as [D\_Number], case when DES\_NO like '+\_%' then SUBSTRING(DES\_NO,2,1) end [Destination Country Prefix], case when DES\_NO like '+\_%' then SUBSTRING(DES\_NO,3,3) end [Destination Prefix Operation], CALL\_TYPE as [Call Type], PROD\_TYPE as [Call Type Descripion], RATED\_AMNT as [Call Rank], RATED\_CURR\_CODE as [Currency], CELL as [Cell], CELL\_ORIGIN as [Cell Origin], HIGH\_LOW\_RATE as [High or Low Time Rating], convert(date,CALL\_DATETIME) as [Call Date]

from Mrr\_Usage\_Main

where CALLING\_NO like '+1%' and DES\_NO like '+1%') tab

order by [Call Number]

Process ETL – Step 4 - Stage to DW - Dimension:

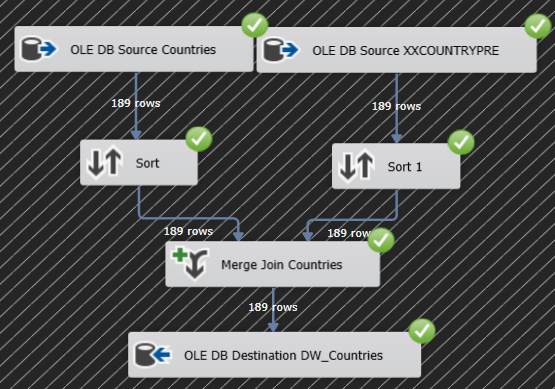
In the Dimension stage we as a team decided how to connect the tables.

For example, table Dim\_Countries:

We've added from "SSIS Toolbox" Option under Common Categories - Sort.

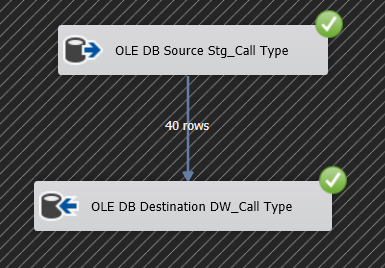
In this tables we sort by country name.

The result is 189 rows passed with data from both tables.



Another example:

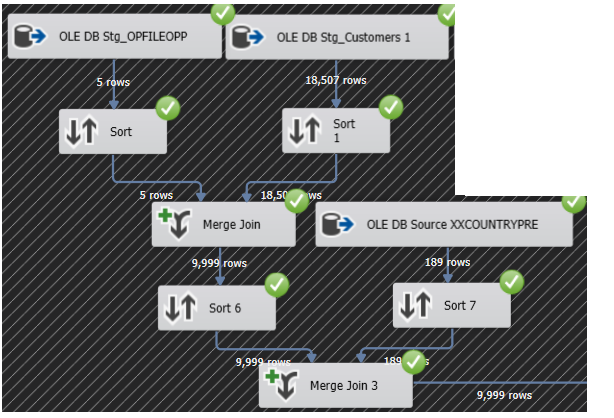
Transferring data from a "Stage" step to the final table – DW\_Call\_Type.



The critical part of the Dimension phase was to characterize the customer table data.

Here we chose the team to Split the customers in two.

"Israeli" customers and customers of the "Rest of the world".



Here we see how we "handled" Israeli customers.

We took three tables and compiled their data.

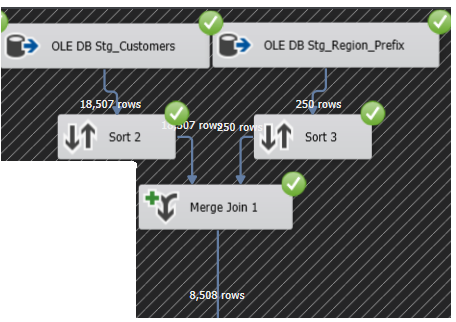
First, we connected the "customers" table to the table "OPFILEOPP", Sort the data by "Prefix Operation" column and after then we performed the "Merge Join" operation.

And so, we pulled out the "Operating Company" column that we need to continue.

next, after connecting the two tables we had to connect another table.

The "XXCOUNTRTPRE" table. This time, we sorted the data with the column that links "Country Prefix" and we did the same thing "Merge Join" and now we have taken the column "Country".

We can see that 9,999 rows are passed. these are "Israeli" customers.



Now, we will refer to the "Rest of the World" customers, we linked the "Customers" table to the "Region\_Prefix" table We created it in the "Mirror" stage Because each area has its own prefix. For example: In this project the "United States", the state's prefix is ​​+1 and New York's prefix is ​​212. Therefore, in sort 2 and 3 We chose the "Country Prefix" and "City Prefix" columns and we compared it to the "Prefix Operation" column.

After that we performed "Merge Join" where we took the country name.

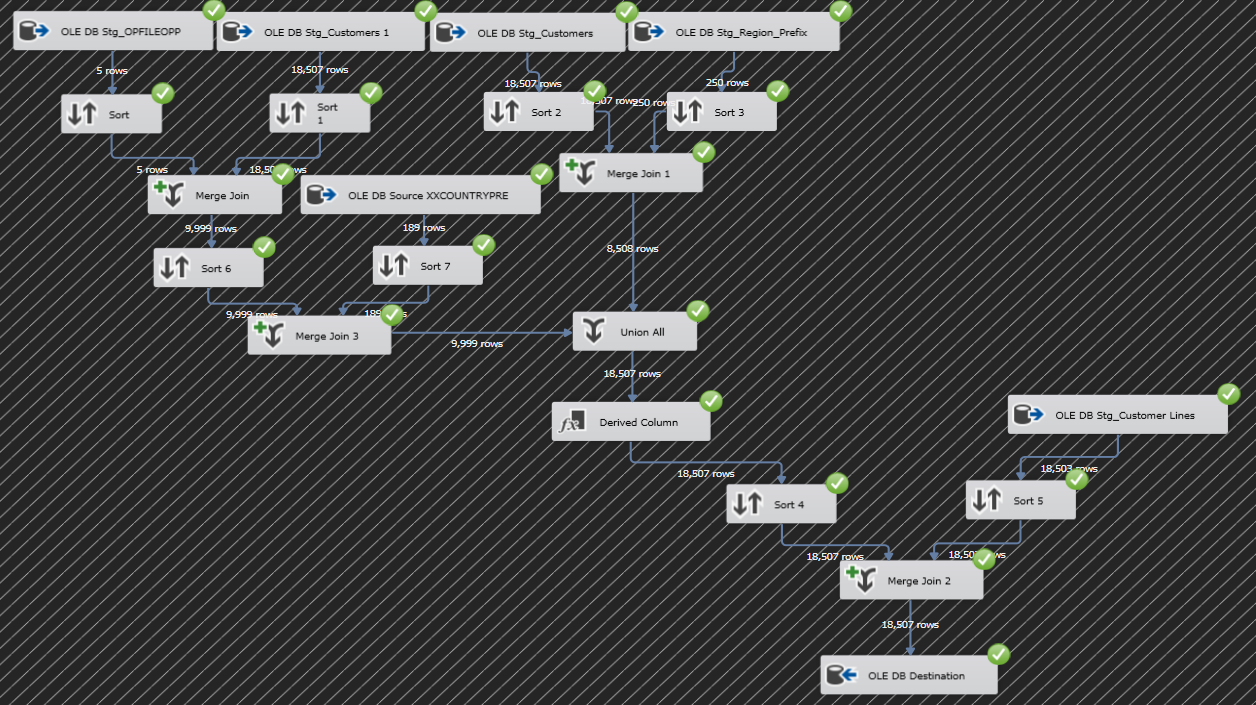
We can see that 8,508 rows are passed. these are " Rest of the World " customers.

Then as a team we made a connection of all customers with option "Union All" that Found in the "SSIS Toolbox", and we make 18,507 of Customers, this figure is featured in the following image.

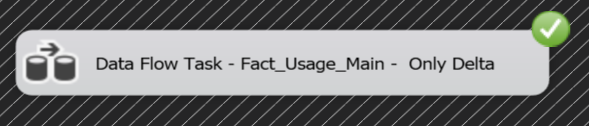
We can also see another connection between the tables that we created the Customers table to "Costumer\_lines" table and this is to produce at the end of the process "Dim\_Costumers".

The sort has been done between "Phone Number" from "Costumer\_lines" and "Customer Number" Of all the connections made on the left side.

**This the result:**

Process ETL - Integration of Countries and Prefix Operators tables with Customers table.

Here as a team we present the creation of a table "Fact":



The fact data transfer only Delta data.

We as a team have decided to split the fact process in two. Origin and Destination.

We've taken the "Usage\_Main" table and the "Customer\_Lines" table and sort the data with "Calling Number" and "Phone Number" and we take all columns.

In Merge Join we merged the data with the same columns.

Next, We begin to connect the data with the dimension We have already created in the previous step.

In "Lookup Dim\_Customers" we choosed "Redirect rows to no match output" Because we want data to be entered without a match.

Within the Lookup under the "Columns" definition we connect "Calling Number" and "Customers Number" and we take from the table "Dim\_Costumers" two columns "Customers ID" and "Country".

Next, We connect to a table of "Dim\_Countries" we choosed again "Redirect rows to no match output".

Within the Lookup under the "Columns" definition we connect "Calling Country Prefix" and "Country" We pulled this column from a table "Dim\_Customers" and we take from the table "Dim\_Countries" the column "Key Country".

Next, in order for data to be matched to the fact table, we need to use the process at "Data Conversion" from "SSIS Toolbox".

We changed the data type to columns "Calling Prefix Operation", "Calling Country Prefix" to A four-byte, signed integer [DT\_I4] data type.

"Seized Date" to date structure that consists of year, month, and day [DT\_DBDATE].

and "Seized Time" to time structure that consists of hour, minute, second, and fractional seconds. The fractional seconds have a maximum scale of 7 digits [DT\_DBTIME2].

Next, we chose to get the derived column to avoid "Null" values (error handling), Because prefix 212 does not belong to any operator.

ISNULL([Key Mobile Prefix]) ? -1 : [Key Mobile Prefix].

Next, We are sorting (4 and 5) by "Merge Join.Customer ID" 🡪 Merge Join 1.Customer ID and Call Number 🡪 Call Number.

After the sorting we do "Merge Join 2" And connects the same data from the sorting stage.

We take all the columns on both sides, Origin and Destination.

Next, in "Lookup Dim\_Call\_Types" we are connected between column "Call Type" 🡪 "Call Type Code" and we take the columns from Lookup "Key Call Type" and "Price Per Minute".

The "Price Per Minute" column we take the next step of column calculation.

Next, in "Lookup Dim\_Package\_Catalog" we connected between column "Package Type" and take "Package Number" column.

Next, in "Lookup Dim\_Call\_Origin\_Type" we connected between column

"Key Source Cell" = "Cell Origin" and take "Key Source Cell" column.

Next, in "Lookup Dim\_Date" After we created this dimension in SSMS with a script "Calander\_Table" 🡪 "Dim\_Date" we connected between column "Key Date" = "Seized Date" and take "Key Date" column.

Next, in "Lookup Dim\_Time" After we created this dimension in SSMS with a script

"dbo.Time" 🡪 "Dim\_Time" we connected between column "Key Time" = "Seized Time" and we take "Key Time" column.

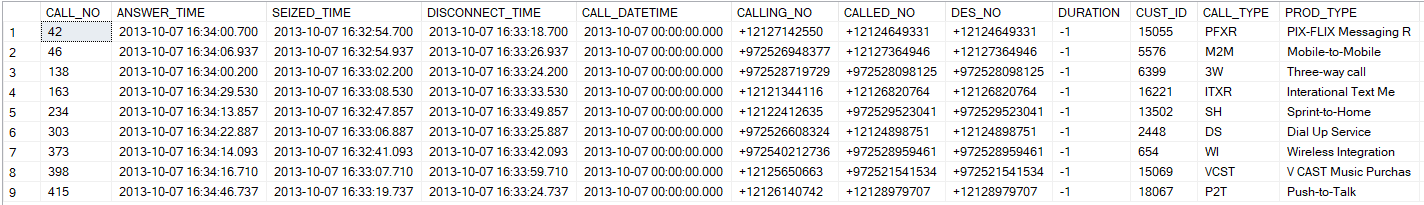
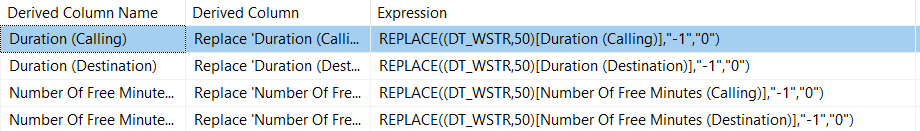
* This note applies on "Dim\_Time" and "Dim\_Date"

The reason we created a table from a table script is to remove columns we don't want.

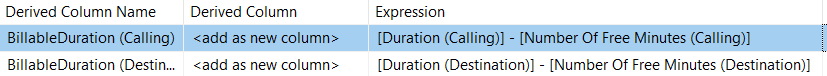
The next step is to handle errors to existing in data.

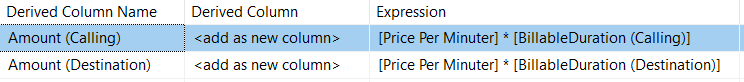
There is data in the "Usege Main" table that they are wrong and that is how we handled it.

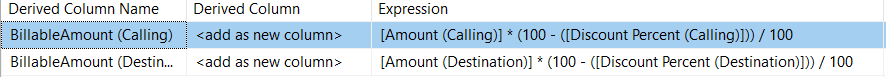
For example:

The "Duration" column had negative data "-1" so convert them to 0.

The next step is to prepare calculated columns in Derived Column:

1) Derived Column BillableDuration

2) Derived Column Amount

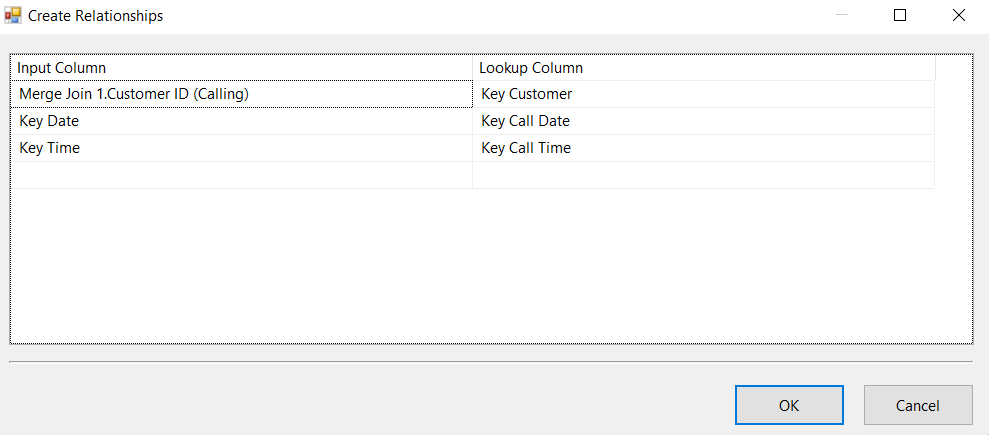
3) Derived Column BillableAmount

The next step is connect to "Lookup Fact\_Usage\_Main", And we decided as a team to put together three columns to make sure they didn't come back.

The Columns is:

"Merge Join 1.Customer ID" 🡪 "Key Customer"

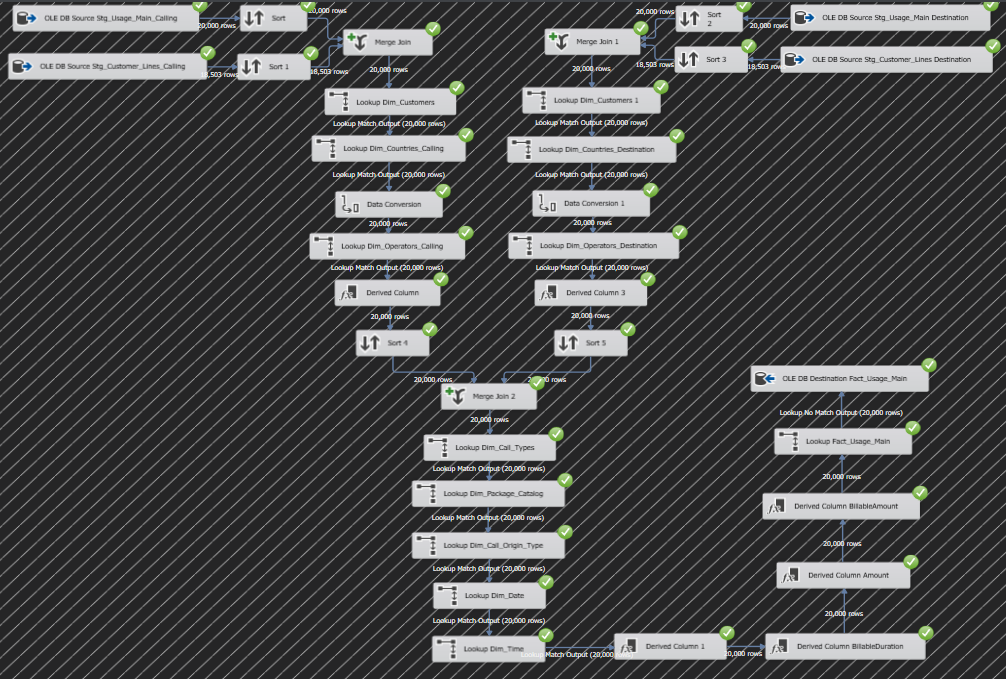
"Key Date" 🡪 "Key Call Date"

"Key Time" 🡪 "Key Call Time"

In "Lookup Fact\_Usage\_Main" we choose that the returned data will not match and thus we get the delta.

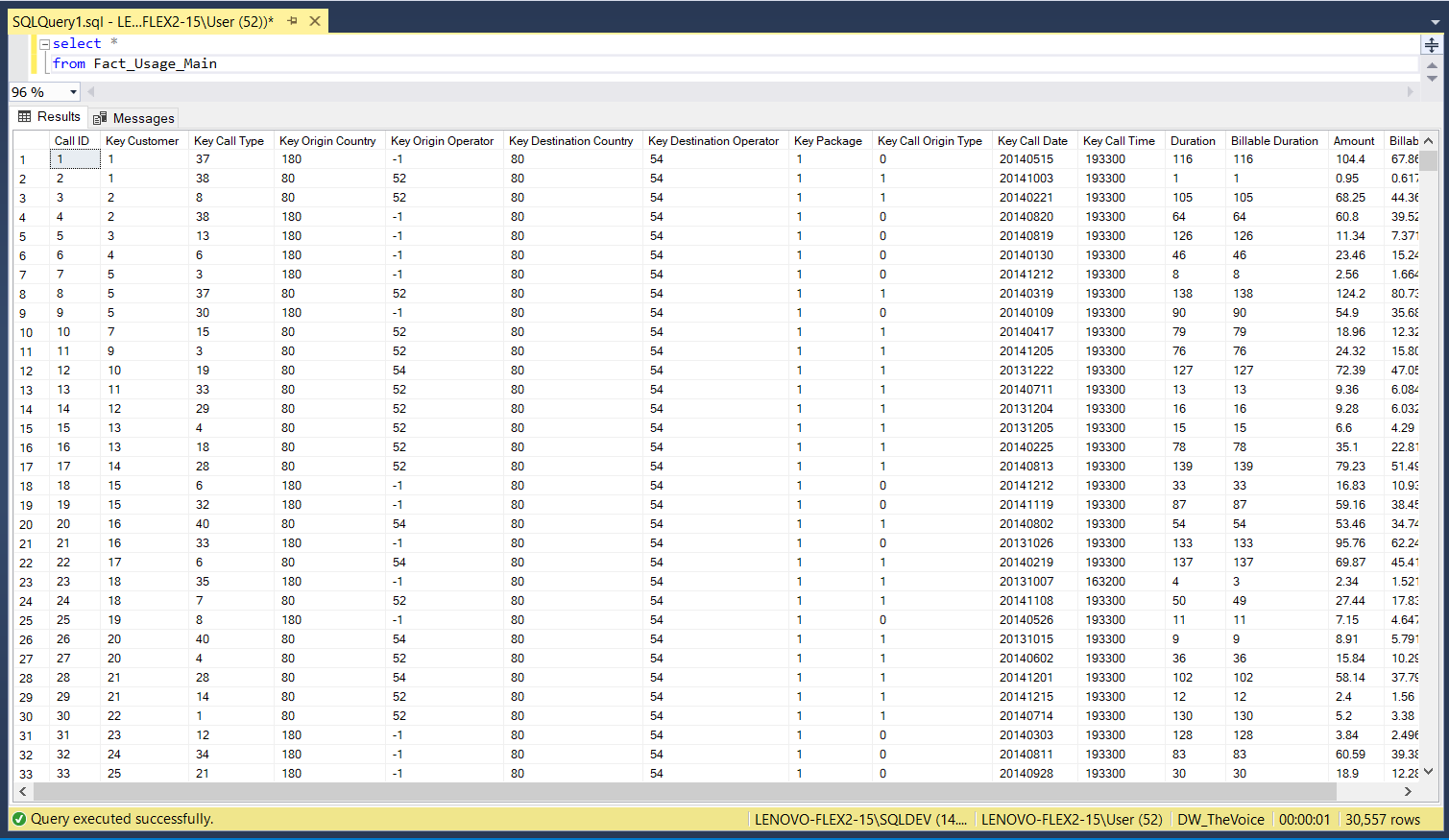
If we restart the package because no data is added then the result will be (0 rows).

**This the result:**

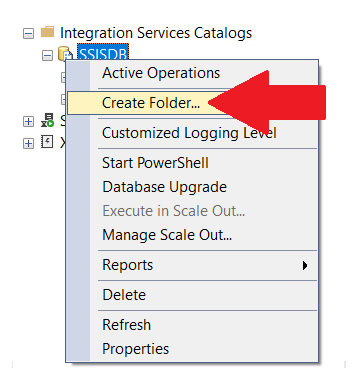
Step 5 - Structure and Process ETL in Fact DW:

After the entire process is complete, the SSMS software should be opened and the role of the "Fact" table we obtained is the desired result.

In the "Fact" table we are supposed to get numeric values.

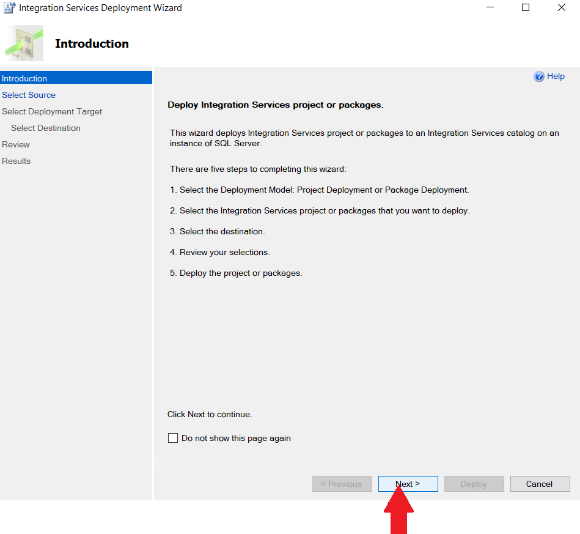
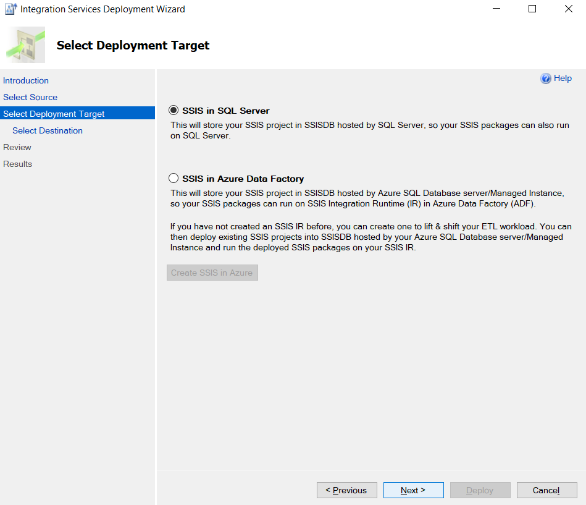
The numerical data is basically the data that links the "Fact" table to the "Dimension" tables.

Before deploying the ETL process, an "Integration Services Catalog" must be produced in SSMS.

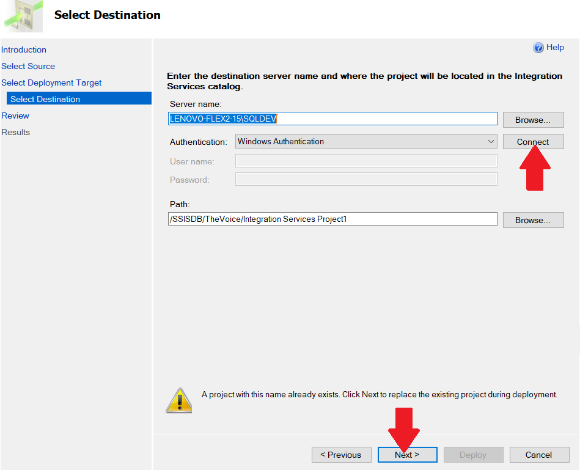
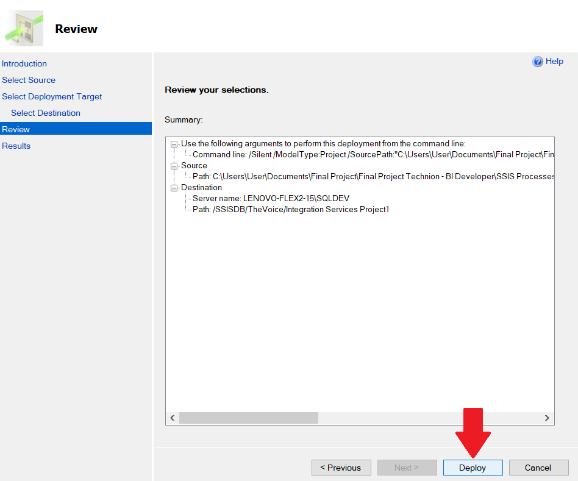


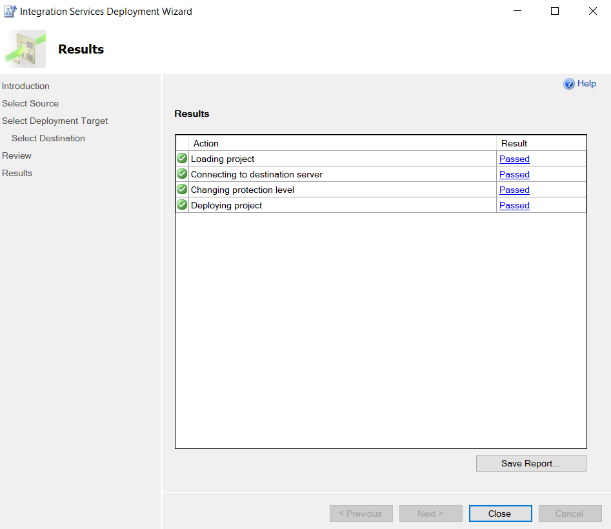
After creating the "Integration Services Catalog" under the project name "TheVoice" we perform the "Deploy".

Access the left side of the project name, right-click and select "Deploy".

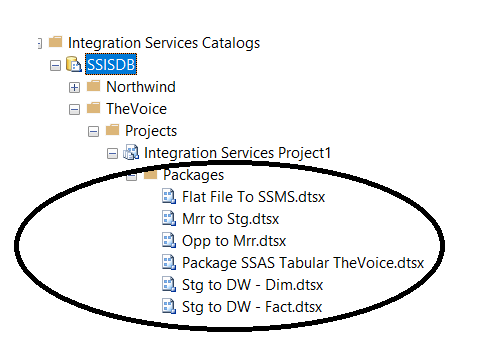
step 1: step 2:

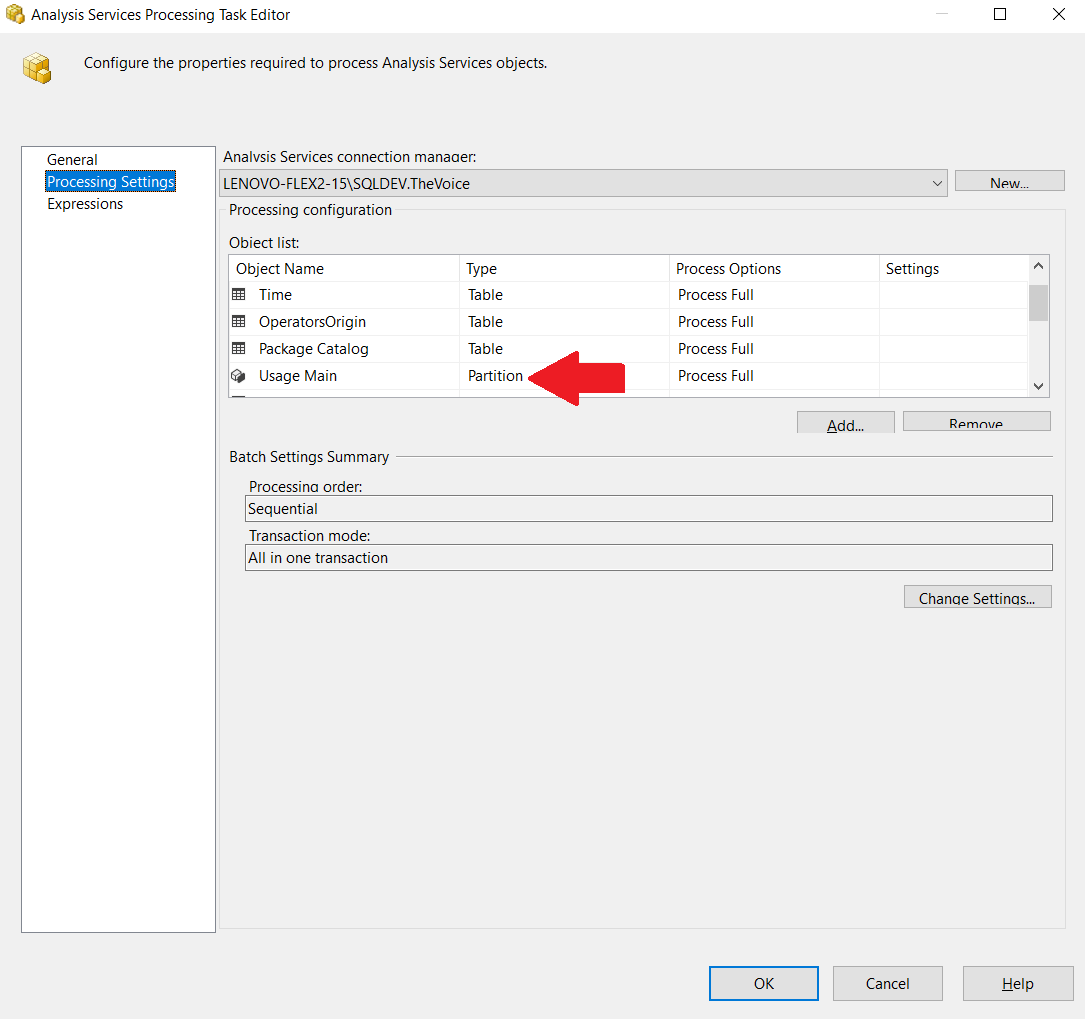


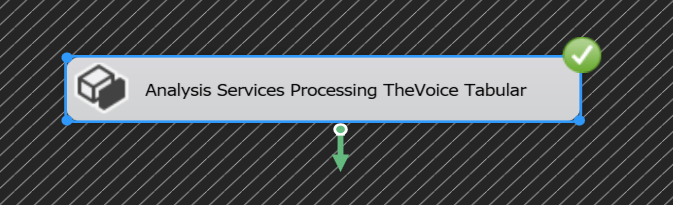
step 3: step 4:

step 5:

After the deployment, enter the project under "Integration Services Catalog" and we can see all the packages we created in SSIS.



We will refer to the "Package SSAS Tabular TheVoice" package later in the review.



1. Page 3, Print screen picture of call type table. [↑](#footnote-ref-1)