

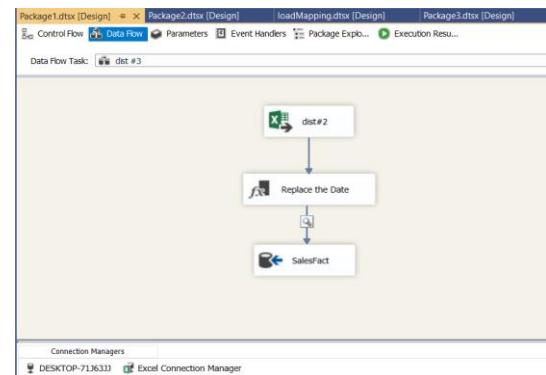
ETL Process: Extracting, Cleaning, and Loading Data from Excel to SQL Server Using SSIS

1. Extracting Data from Excel to SSIS

I used **SSIS (SQL Server Integration Services)** to read

Data from an **Excel** file. This typically involves:

- Adding an **Excel Source** in the **Data Flow Task**.
- Ensuring that column data types are properly mapped.

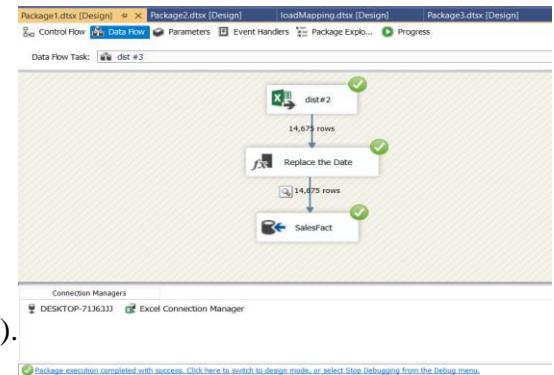


2. Loading Data into SQL Server

Once SSIS extracted the data, I loaded it into

SQL Server using **OLE DB Destination**, which involves:

- Defining the **destination table** in the database.
- Mapping columns between Excel and the SQL table.
- Choosing an **insert mode**:
 - **Append** (Add new records without deleting existing ones).
 - **Overwrite** (Replace old data with new data).



3. Cleaning Data Efficiently in SSIS

I used SSIS tools to clean and transform data before loading, which may include:

- **Derived Column**: Creating new columns or modifying existing values.
- **Data Conversion**: Resolving data type mismatches (e.g., converting text to numbers or dates).
- **Conditional Split**: Filtering data based on conditions (e.g., removing blank rows or invalid values).
- **Lookup**: Matching records with reference tables to ensure data consistency.
- **Aggregate**: Performing summarization before loading the data.

```
SQLQuery1.sql - DE_71J6331AFAQ(51)* - X ****
***** Script for SelectTopNRows command from SSMS *****
--> SELECT [Client Code]
      ,[Item Code]
      ,[Date]
      ,[Quantity]
      ,[Sales]
  FROM [master].[dbo].[salesfact]
```

A screenshot of the SSMS 'Results' tab. It displays a table with five columns: Client Code, Item Code, Date, Quantity, and Sales. The data consists of 16 rows, each representing a sales record with specific values for each column. At the bottom of the results pane, a green bar indicates 'Query executed successfully.'

	Client Code	Item Code	Date	Quantity	Sales
1	426599	2413	2019-07-01	10	1
2	432152	2413	2019-07-01	10	2
3	334071	2413	2019-07-01	1	1
4	405758	2413	2019-07-01	5	1
5	405670	2414	2019-07-01	5	1
6	428149	2414	2019-07-01	5	1
7	389883	2414	2019-07-01	5	1
8	402530	2414	2019-07-01	6	1
9	409164	2414	2019-07-01	15	2
10	422585	2414	2019-07-01	1	1
11	434663	2414	2019-07-01	1	1
12	340297	2414	2019-07-01	3	1
13	331771	2414	2019-07-01	3	1
14	331809	2414	2019-07-01	3	1
15	332711	2414	2019-07-01	4	1
16	435660	2414	2019-07-01	4	1

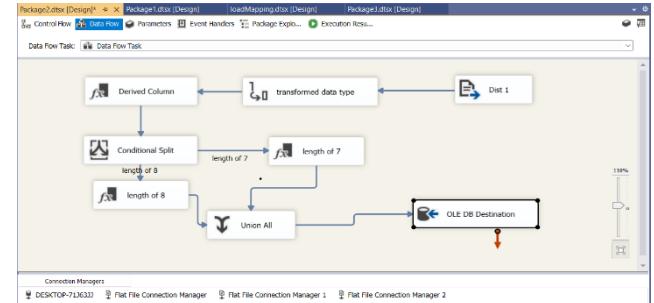
Processing Flat File Data Using SSIS: Extraction, Cleaning, and Loading into SQL Server

After successfully handling Excel data, the next step i completed involved working with a **Flat File**. Here's a breakdown of the process:

1. Extracting Data from a Flat File to SSIS

I used **SSIS** to read data from a **Flat File (CSV, TXT, etc.)** , which requires:

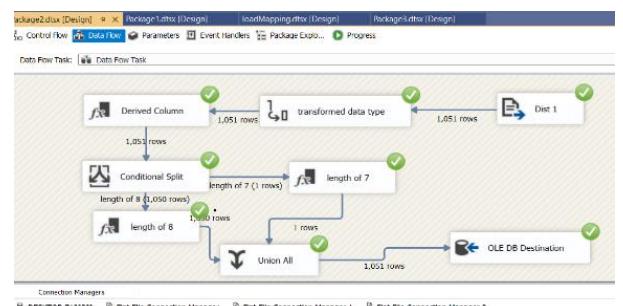
- Adding a **Flat File Source** in the **Data Flow Task**.
- Configuring the **file format** (Delimited, Fixed Width, etc.).
- Setting the correct **column delimiters** .
- Handling encoding issues (e.g., UTF-8, ANSI).



2. Loading Data into SQL Server

After extracting the data, i loaded it into **SQL Server** using **OLE DB Destination**:

- Defined the **destination table** in the SQL database.
- Mapped **columns** between the Flat File and the SQL table.
- Selected an appropriate **loading mode**:
 - **Fast Load** for bulk inserts.
 - **Row-by-Row processing** for complex transformations.



3. Cleaning and Transforming Data in SSIS

Before loading the data into SQL, I used **SSIS transformations**

to clean and enhance the data:

- **Derived Column:** Created new calculated fields or reformatted existing data.
- **Data Conversion:** Fixed data type mismatches (e.g., converting text to numbers or dates).
- **Conditional Split:** Separated records based on conditions (e.g., filtering out invalid data).
- **Lookup Transformation:** Validated data against a reference table in SQL Server.
- **Replace Nulls:** Handled missing values to avoid errors in SQL queries.

The screenshot shows a SQL query window titled 'Query1.sql - DE...71633\NFAQ (51)*'. The query is:`===== Script for SelectTopNRows command from SSMS =====
SELECT [Client Code]
 ,[Item Code]
 ,[Date]
 ,[Quantity]
 ,[Sales]
 FROM [master].[dbo].[salesFact]`

Below the query, the results pane shows a table with columns Client Code, Item Code, Date, Quantity, and Sales. The data is as follows:

	Client Code	Item Code	Date	Quantity	Sales
2088	261213	RN1576	2019-06-10	10	18
2089	262908	RN11661	2019-06-10	2	18
2090	262908	RN1476	2019-06-10	2	9
2091	262908	RN1507	2019-06-10	5	27
2092	262908	RN1655	2019-06-10	1	45
2093	262908	RN1660	2019-06-10	10	18
2094	262908	RN1505	2019-06-10	5	9
2095	243425	RN1507	2019-06-10	5	18
2096	264135	RN1509	2019-06-10	2	27
2097	263336	RN2110	2019-06-10	3	27
2098	263336	RN2702	2019-06-10	5	45
2099	263062	RN1597	2019-06-10	5	27
2100	245033	RN1656	2019-06-10	2	18
2101	245033	RN1716	2019-06-11	2	9
2102	251033	RN1716	2019-06-11	2	18

At the bottom, a message says 'Query executed successfully.'