Assignment – V (SSIS)

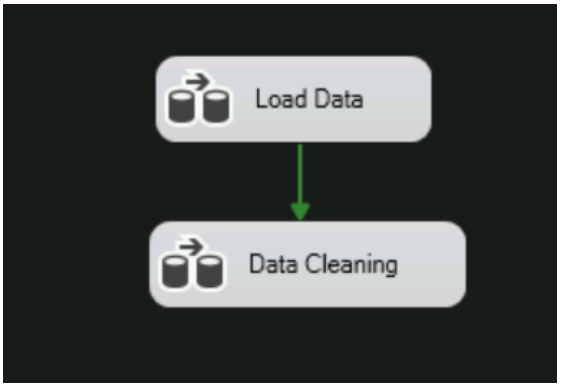
Task 1: Integration with ETL Data Warehouse (DWH)

Scenario: Your company has a data warehouse designed to consolidate data from various sources for analytical purposes. You need to create an SSIS package that extracts data from a transactional database and loads it into the data warehouse.

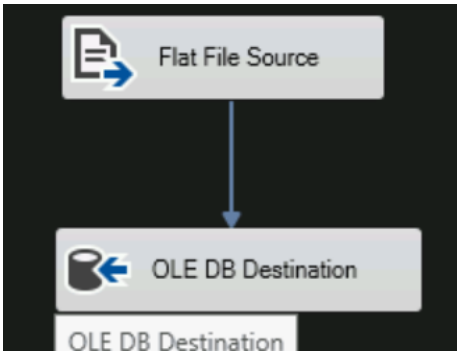
Requirements:

1. Create a Connection Manager to connect to the transactional database and the data warehouse.
2. Extract Data from a transactional table (e.g., SalesData) using an OLE DB Source.
3. Transform Data:
   Apply necessary transformations such as data type conversions, data cleansing, and calculations.
4. Load Data into the data warehouse (e.g., FactSales table).

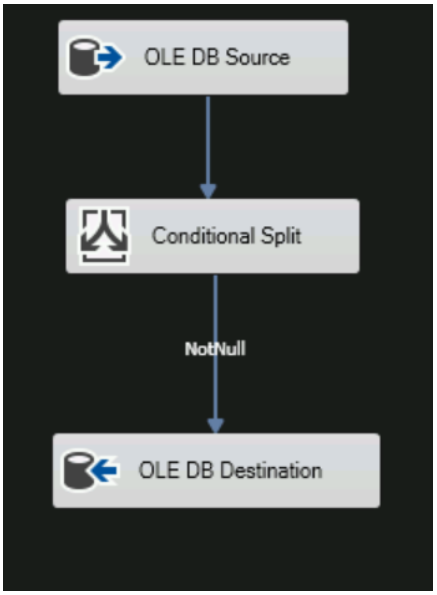
Step 1 - We have 2 Dataflow tasks here one for loading the data and the other for the cleaning up the data



Initially we are loading the data into the data warehouse From a CSV File.



Then we are actually removing the null values and Putting it into an OLE DB Destination



The output will be like this

	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	PRODUCTLINE	MSRP	PRODUCTCODE	CUSTOMERNAME
1	100	1	3965.66	2/20/2004 0:00	Shipped	1	2	2004	Motorcycles	95	S10_1678	Australian Collectors
2	100	7	2333.12	4/5/2004 0:00	Shipped	2	4	2004	Motorcycles	95	S10_1678	Vitachrome Inc.
3	72.55	13	1451	12/17/2004 0:00	Shipped	4	12	2004	Motorcycles	95	S10_1678	Souveniers And Thii
4	100	1	4860.24	10/20/2003 0:00	Shipped	4	10	2003	Classic Cars	214	S10_1949	Classic Legends Inc
5	100	9	4905.39	7/19/2004 0:00	Shipped	3	7	2004	Classic Cars	214	S10_1949	Souveniers And Thii
6	100	1	3944.7	11/29/2004 0:00	Shipped	4	11	2004	Classic Cars	214	S10_1949	Australian Collectors
7	100	4	2416.56	3/9/2005 0:00	Shipped	1	3	2005	Classic Cars	214	S10_1949	Anna's Decorations,
8	96.34	3	2793.86	4/29/2003 0:00	Shipped	2	4	2003	Motorcycles	118	S10_2016	Australian Collectors
9	100	4	5422.39	2/20/2004 0:00	Shipped	1	2	2004	Motorcycles	118	S10_2016	Australian Collectors

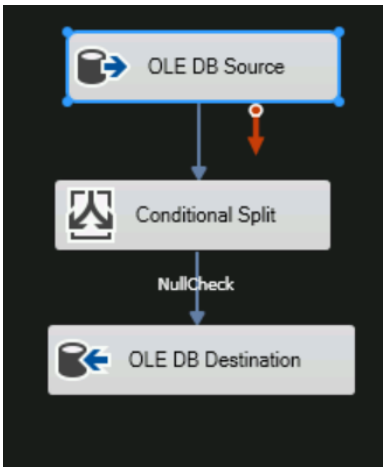
Task 2: Data Warehouse Migrations

Scenario: Your organization is migrating its data warehouse from one server to another. You need to create an SSIS package that facilitates this migration.

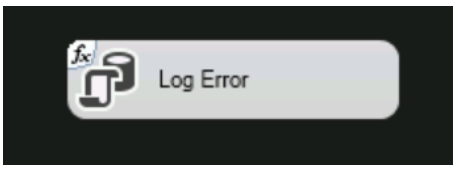
Requirements:

1. Create Connection Managers for both the source and destination data warehouses.
2. Transfer Data from the source data warehouse to the destination using the Data Flow Task.
3. Ensure Data Integrity:
    a) Include checks and balances to ensure data is correctly migrated.
    b) Log the success or failure of the migration process.

We are actually extracting the data from the data warehouse and checking for null values and migrating it into a new table.



Then we are also logging the errors which will be displayed in a table



The expression for error logging would be like this

Expression:

```
"insert into Error_Logsvalues ("+ @[System::MachineName] +"','"+ @[System::PackageName] +"','"+ @[System::SourceName]  +"','"+(DT_WSTR,12) @[System::ErrorCode] +"','"+ @[System::ErrorDescription] +"',getdate())"
```

In the SQL Server we have to do a setup like this ie we are creating a table for logging errors where the errors will be displayed.

```
IF NOT EXISTS(SELECT 1 FROM sysobjects with (nolock) WHERE ID = OBJECT_ID(N'Error_Logs') AND type = (N'U'))
CREATE TABLE Error_Logs(ID INT IDENTITY, MachineName varchar(200),PackageName varchar(200), TaskName varchar(200), ErrorCode int,
ErrorDescription varchar(max), Dated datetime)
go

insert into Error_Logs
values ('MachineName','PackageName','TaskName',0,'Errordesc',getdate())

IF EXISTS(SELECT 1 FROM sysobjects with (nolock) WHERE ID = OBJECT_ID(N'test') AND type = (N'U'))
DROP TABLE test

select * from Error_Logs
```

Output of error table

Results Messages

	ID	MachineName	Package...	TaskName	ErrorCode	ErrorDescription	Dated
1	1	4E702DEACDF5502	Task 2	Data Flow Task	-1071636466	Cannot open the datafile "C:\Users\Administrator\Downloads\sales_data_sample1.csv".	2024-07-25 00:15:2
2	2	4E702DEACDF5502	Task 2	Data Flow Task	-1073450982	Flat File Source failed the pre-execute phase and returned error code 0xC020200E.	2024-07-25 00:15:2
3	3	4E702DEACDF5502	Task 2	Execute SQL Task	0	Unclosed quotation mark after the character string 'WHERE ORDERNUMBER = 1A'.	2024-07-25 00:15:5
4	4	4E702DEACDF5502	Task 2	Execute SQL Task	-1073548784	Executing the query "UPDATE TEST SET PRODUCTLINE = '' WHERE ORDERNU...	2024-07-25 00:15:5

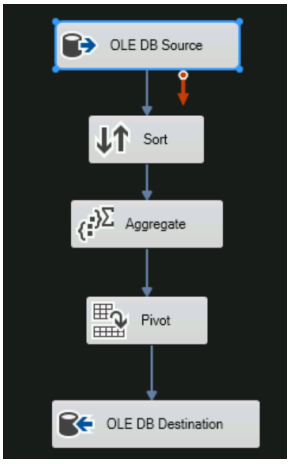
Task 3: Implementing a Pivot Transformation

Scenario: You have data in a normalized format and need to pivot it for reporting purposes.

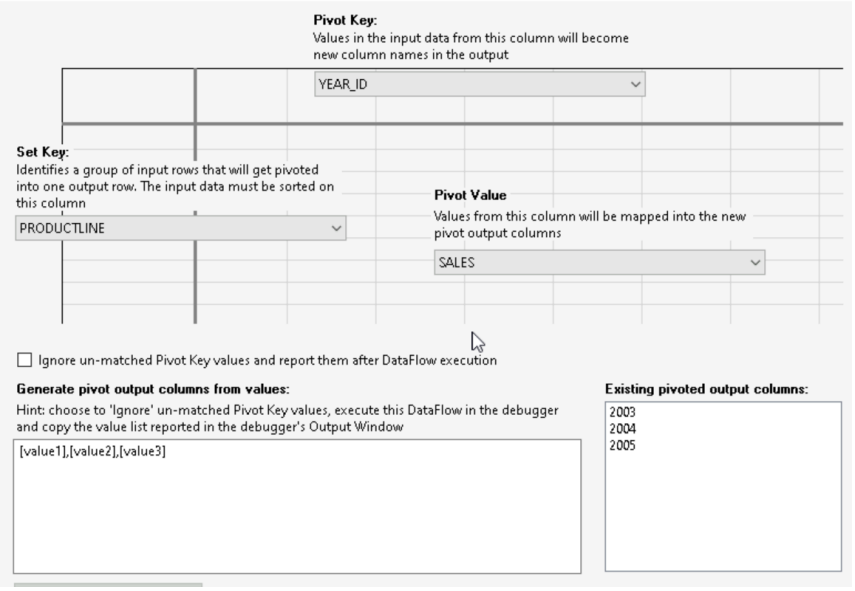
Requirements:

1. Extract Data from the source table using an OLE DB Source.
2. Apply a Pivot Transformation to transform the normalized data into a pivoted format.
3. Load the Pivoted Data into a destination table.

We are actually extracting the data from the data warehouse initially and then we are sorting it by a single key we are also using an aggregate function which is used for aggregating the data else the pivot will not work and finally we are passing it to the OLE DB Destination.



In the pivot key column we are setting up the pivot keys and renaming the new column values



The output of pivoting would be like this

	PRODUCTLINE	2003	2004	2005
1	Classic Cars	NULL	120	NULL
2	Trains	6	NULL	NULL
3	Planes	21	NULL	NULL
4	Vintage Cars	NULL	54	NULL
5	Classic Cars	135	NULL	NULL
6	Trucks and Buses	NULL	51	NULL
7	Ships	NULL	18	NULL
8	Motorcycles	NULL	75	NULL
9	Trains	NULL	3	NULL
10	Vintage Cars	75	NULL	NULL

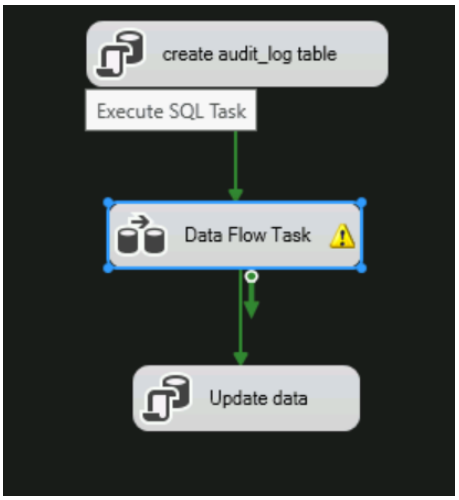
Task 4: Incremental Load

Scenario: To optimize ETL processes, you need to implement an incremental load to update only the changed data in the data warehouse.

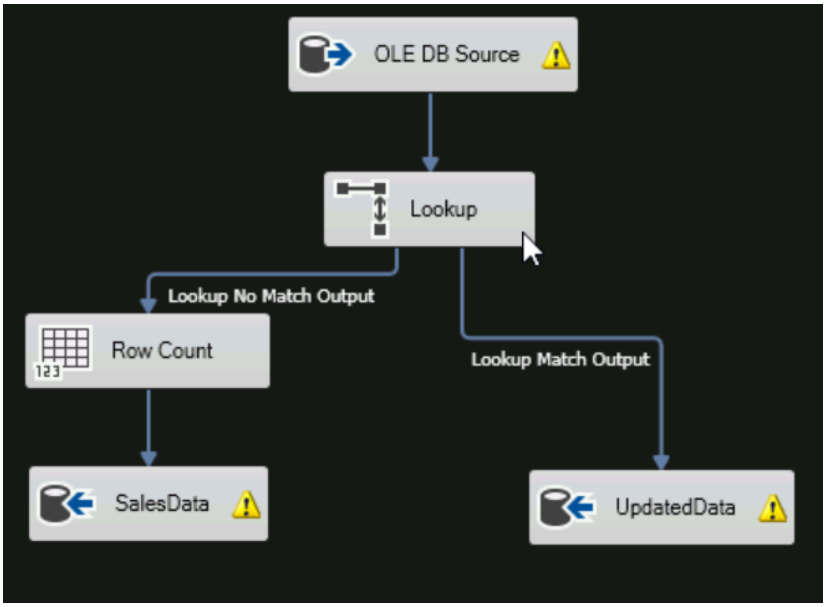
Requirements:

1. Identify Changed Data: Use methods such as timestamps, change data capture using lookup, or checksums.
2. Extract Only the Changed Data from the source.
3. Update the Data Warehouse with the new and changed data only.

What we are doing here is that we are actually creating an audkt_log table using an SQL script and then executing the Data Flow Task after that creating the Update table which is used in Lookup.



Inside the dataflow we are actually extracting the data from the data warehouse and then passing it to a lookup.What the lookup does is that it redirects matching and non matching values based on a criteria in this if we are inserting a data it will be into the SalesData and if we are updating it then it will be put into the UpdatedData table.



The output of the incremental load table would be like this

	Id	PackageName	TableName	RecordsInserted	RecordsUpdated	Dated
1	1	IncrementalLoad.dtsx	sales	750	0	2024-07-25 21:35:02.290
2	2	IncrementalLoad.dtsx	sales	0	750	2024-07-25 21:40:02.957
3	3	IncrementalLoad.dtsx	sales	0	750	2024-07-25 21:52:54.513

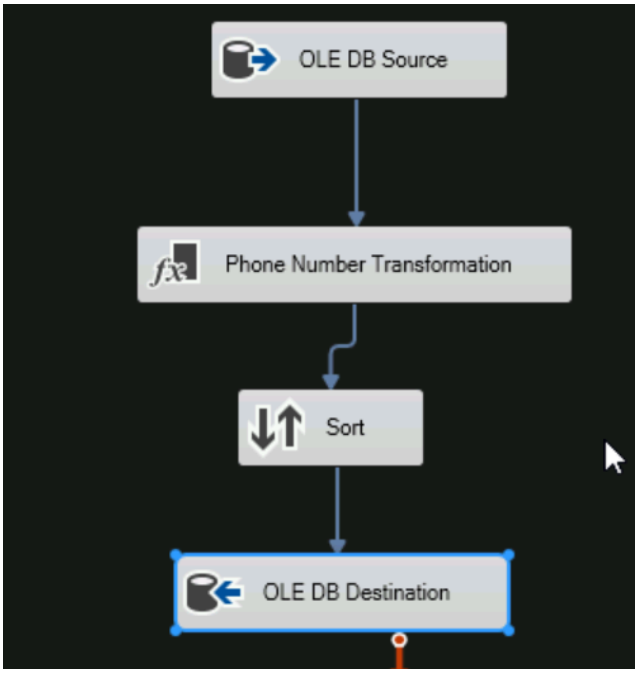
Task 5: Transformations

Scenario: Your company needs to transform raw data into a format suitable for reporting. You need to perform multiple transformations within an SSIS package.

Requirements:

1. Extract Data from a source table using an OLE DB Source.
2. Apply Transformations such as:
    - Data Conversion
    - Derived Column
    - Conditional Split
    - Aggregate
3. Load Transformed Data into a destination table.

Here we are transforming the phone number into a standard format in the extracted data and sorting it based on ORDERNUMBER as KEY and transferring it to a new destination.



The Cleaned PhoneNumber Would look like this

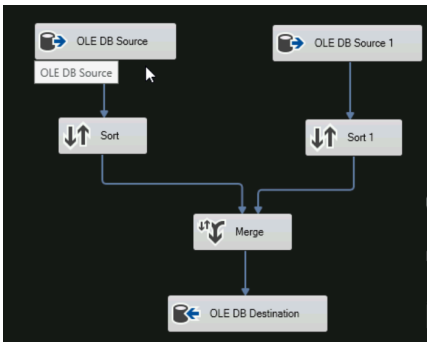
	TLINE	MSRP	PRODUCTCODE	CUSTOMERNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE	POSTALCODE	CC
82		86	S700_1938	Classic Legends Inc.	2125558493	5905 Pompton St.	Suite 750	NYC	NY	10022	U
83		122	S24_2011	Classic Legends Inc.	2125558493	5905 Pompton St.	Suite 750	NYC	NY	10022	U
84	ars	143	S18_4027	Classic Legends Inc.	2125558493	5905 Pompton St.	Suite 750	NYC	NY	10022	U
85		58	S50_1514	Classic Legends Inc.	2125558493	5905 Pompton St.	Suite 750	NYC	NY	10022	U
86	;ars	87	S18_4522	Classic Legends Inc.	2125558493	5905 Pompton St.	Suite 750	NYC	NY	10022	U
87		54	S72_3212	Classic Legends Inc.	2125558493	5905 Pompton St.	Suite 750	NYC	NY	10022	U
88	ars	136	S10_4757	Classic Legends Inc.	2125558493	5905 Pompton St.	Suite 750	NYC	NY	10022	U
89	nd B...	127	S24_2300	Vitachrome Inc.	2125551500	2678 Kingston Rd.	Suite 101	NYC	NY	10022	U
90	ars	143	S18_4027	Vitachrome Inc.	2125551500	2678 Kingston Rd.	Suite 101	NYC	NY	10022	U

Task 6: MERGE & FUZZY LOOKUP
Scenario: You need to merge two datasets and use fuzzy matching to handle potential duplicates.

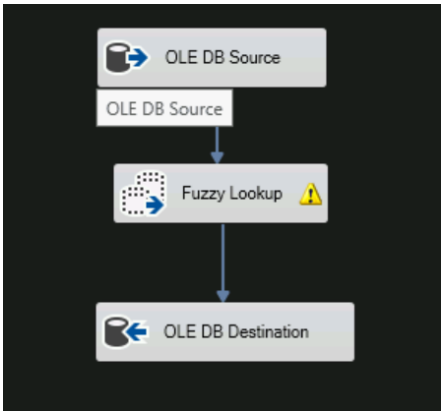
Requirements:

1. Extract Data from two source tables using OLE DB Sources.
2. Apply a Merge Join to combine the datasets based on a common key.
3. Use Fuzzy Lookup to identify and resolve duplicates in the merged data.
4. Load the Cleaned Data into a destination table.

Here we have 2 OLE DB Source One contains the sales_sample_data.csv and the other one contains A custom dataset containing similar columns Then we are sorting both the extracted data based on a common key and merging it so that the output will be a single dataset and then putting it into a destination table.



Here we are performing fuzzy lookup to check for similarity of columns.



The output of fuzzy lookup will be like this

COUNTRY	TERRITORY	CONTACTLASTNAME	CONTACTFIRSTNAME	DEALSIZE	QUANTITYORDERED	ORDERNUMBER	ORDERNUMBER (1)	_Similarity	_Confidence
USA	NA	Frick	Michael	Small	41	10102	10102	1	1
Australia	APAC	Ferguson	Peter	Medium	35	10120	10120	1	1
Australia	APAC	Ferguson	Peter	Medium	39	10120	10120	1	1
Australia	APAC	Ferguson	Peter	Small	24	10120	10120	1	1
Australia	APAC	Ferguson	Peter	Large	46	10120	10120	1	1
Australia	APAC	Ferguson	Peter	Small	29	10120	10120	1	1
Australia	APAC	Ferguson	Peter	Medium	24	10120	10120	1	1
Australia	APAC	Ferguson	Peter	Small	29	10120	10120	1	1
Australia	APAC	Ferguson	Peter	Medium	43	10120	10120	1	1

Task 7: Using Script Task

Scenario: You need to perform a complex data transformation that is not supported by the standard SSIS components. A Script Task can be used to achieve this.

Requirements:

1. Add a Script Task to the Control Flow.
2. Write a Script: that performs the required transformation. e.g. Reading data from a file, processing it, and writing the results to a database table.
3. Execute the Script Task within an SSIS package.

This is a small VB script which writes the data to a table.Initially it sets up a connection to the database and then finds the specified file and finally writing to a table.

```
Public Sub Main()
    Dim connectionString As String = "Data Source=localhost;Initial Catalog=SSIS_ASSMNT;Integrated Security=SSPI;"
    Dim inputFilePath As String = "D:/MOCK_DATA.csv"
    Dim sqlCommandText As String = "INSTERT INTO ExampleTable VALUES (@Column1)"

    Try
        Using connection As New SqlConnection(connectionString)
            connection.Open()

            Using reader As New StreamReader(inputFilePath)
                While Not reader.EndOfStream
                    Dim line As String = reader.ReadLine()
                    Dim values As String() = line.Split(","c)

                    Using command As New SqlCommand(sqlCommandText, connection)
                        command.Parameters.AddWithValue("@Column1", values(0))
                        command.ExecuteNonQuery()
                    End Using
                End While
            End Using

            Dts.TaskResult = ScriptResults.Success
        End Using
    Catch ex As Exception
        Dts.Events.FireError(0, "Script Task", ex.Message, String.Empty, 0)
        Dts.TaskResult = ScriptResults.Failure
    End Try
End Sub
```

The output would look like this

	ORDERNUMBER
4	16534
5	10079
6	10250
7	15692
8	14727
9	16256
10	11491
11	10463
12	10973
13	12464