

Name: Girum Obse

Date: 02/18/2025

Course: BIDD 320 A /Data Migration Techniques (ETL Processing)

Assignment: A05_G.Obse

ETL Workflow for Phone Number Validation with No-SQL Scripts and SSIS

1. Introduction

2. Extracting Source Data

2.1 Source Data Location

- File path and structure of TestData.csv.

2.2 Explanation of Source Data

- Description of fields in the CSV file.

3. Validating Phone Numbers

- Required phone number format (111-222-3333).
- Separation of valid and invalid records.

3.1 Screenshot of ValidData.csv

3.2 Screenshot of BadData.csv

4. Loading Data into SQL Server

4.1 Create Staging Table

- SQL script to create the CustomersStaging table.

4.2 Explanation of Staging Table

- Description of table structure and columns.

4.3 Screenshot of SQL Table Creation

5. SSIS Package Setup

6. Summary

1. Introduction

This document provides a step-by-step guide to an ETL (Extract, Transform, Load) process designed to validate phone numbers from a CSV file. The process ensures that phone numbers adhere to a standardized format before being stored in a structured database. By implementing this validation, organizations can maintain data integrity, improve operational efficiency, and ensure consistency in their customer records.

The ETL process involves three main phases:

1. Extract: The data is retrieved from a source file located at C:\Data\TestData.csv.
2. Transform: The phone numbers are validated against a defined pattern, separating valid and invalid records.
3. Load: The valid records are stored in a SQL Server staging table for further processing, while invalid records are logged separately for review and correction.

This document is intended for data engineers, ETL developers, and support teams responsible for maintaining data quality and troubleshooting the process. Detailed explanations of each step, along with relevant screenshots, are provided to facilitate understanding and implementation.

2. Extracting Source Data

2.1 Source Data Location

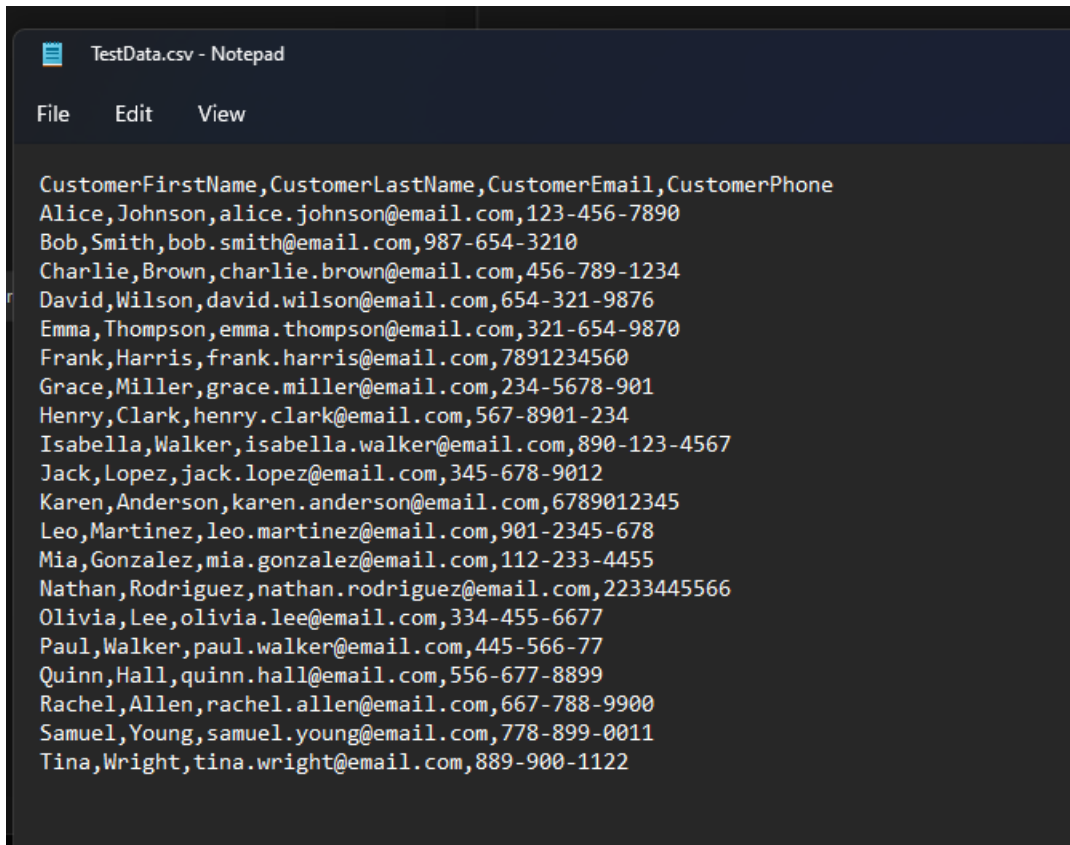
The input data file is stored at: C:\Data\TestData.csv

2.2 Explanation of Source Data

The source data file is a CSV (Comma-Separated Values) file that consists of multiple customer records. Each row represents a customer and includes the following fields: Customer First Name, Customer Last Name, Customer Email, and Customer Phone Number.

2.3 Screenshot of TestData.csv

Picture 1, Screenshot of TestData.csv



```
CustomerFirstName, CustomerLastName, CustomerEmail, CustomerPhone
Alice, Johnson, alice.johnson@email.com, 123-456-7890
Bob, Smith, bob.smith@email.com, 987-654-3210
Charlie, Brown, charlie.brown@email.com, 456-789-1234
David, Wilson, david.wilson@email.com, 654-321-9876
Emma, Thompson, emma.thompson@email.com, 321-654-9870
Frank, Harris, frank.harris@email.com, 7891234560
Grace, Miller, grace.miller@email.com, 234-5678-901
Henry, Clark, henry.clark@email.com, 567-8901-234
Isabella, Walker, isabella.walker@email.com, 890-123-4567
Jack, Lopez, jack.lopez@email.com, 345-678-9012
Karen, Anderson, karen.anderson@email.com, 6789012345
Leo, Martinez, leo.martinez@email.com, 901-2345-678
Mia, Gonzalez, mia.gonzalez@email.com, 112-233-4455
Nathan, Rodriguez, nathan.rodriguez@email.com, 2233445566
Olivia, Lee, olivia.lee@email.com, 334-455-6677
Paul, Walker, paul.walker@email.com, 445-566-77
Quinn, Hall, quinn.hall@email.com, 556-677-8899
Rachel, Allen, rachel.allen@email.com, 667-788-9900
Samuel, Young, samuel.young@email.com, 778-899-0011
Tina, Wright, tina.wright@email.com, 889-900-1122
```

3. Validating Phone Numbers

Phone numbers are expected to follow the format: 111-222-3333.

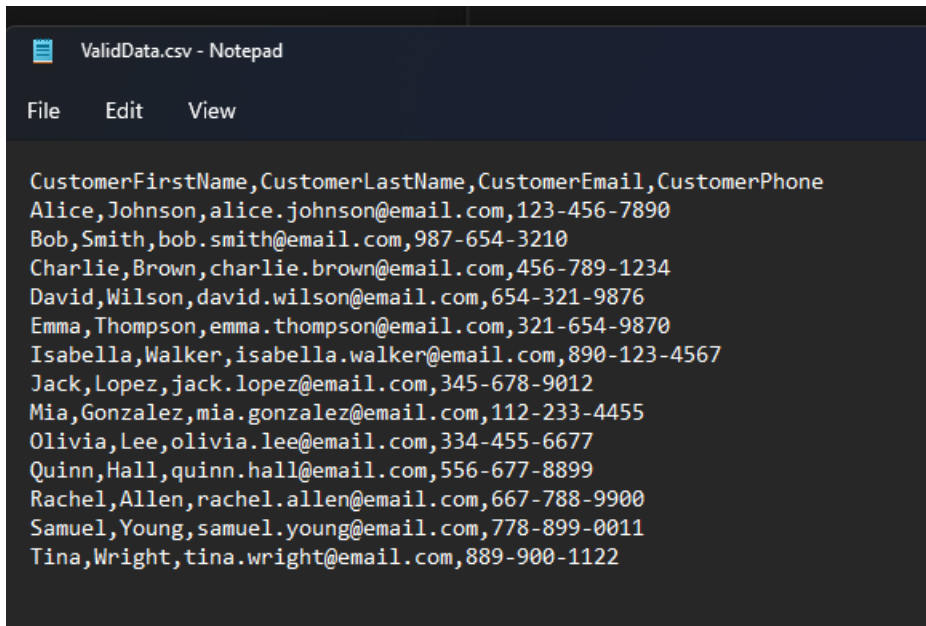
Any entry that does not match this pattern is considered invalid.

The transformation phase separates valid and invalid phone numbers, storing them in two different CSV files:

- Valid Data: Stored in C:\Data\ValidData.csv
- Invalid Data: Stored in C:\Data\BadData.csv

3.1 Screenshot of ValidData.csv

Picture 2, Screenshot of valid data

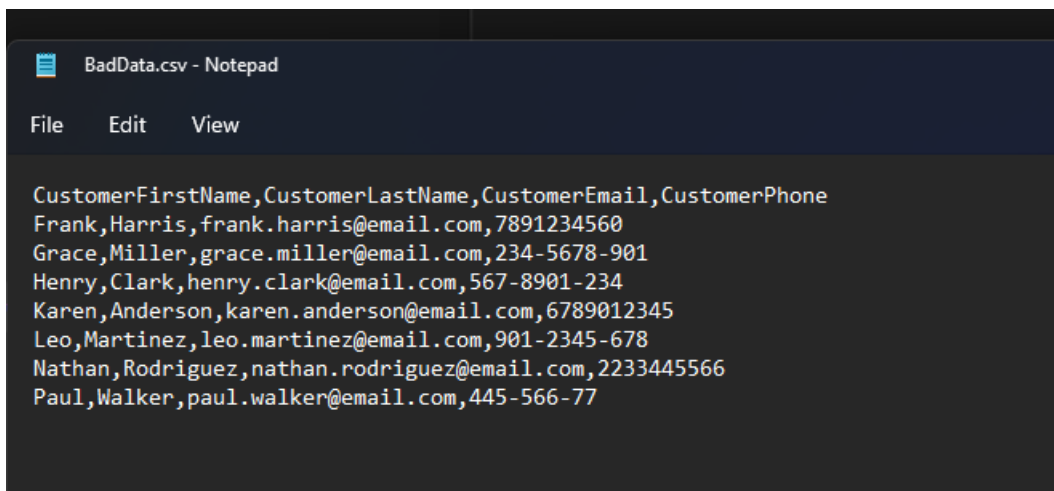


```
ValidData.csv - Notepad
File Edit View

CustomerFirstName,CustomerLastName,CustomerEmail,CustomerPhone
Alice,Johnson,alice.johnson@email.com,123-456-7890
Bob,Smith,bob.smith@email.com,987-654-3210
Charlie,Brown,charlie.brown@email.com,456-789-1234
David,Wilson,david.wilson@email.com,654-321-9876
Emma,Thompson,emma.thompson@email.com,321-654-9870
Isabella,Walker,isabella.walker@email.com,890-123-4567
Jack,Lopez,jack.lopez@email.com,345-678-9012
Mia,Gonzalez,mia.gonzalez@email.com,112-233-4455
Olivia,Lee,olivia.lee@email.com,334-455-6677
Quinn,Hall,quinn.hall@email.com,556-677-8899
Rachel,Allen,rachel.allen@email.com,667-788-9900
Samuel,Young,samuel.young@email.com,778-899-0011
Tina,Wright,tina.wright@email.com,889-900-1122
```

3.2 Screenshot of BadData.csv

Picture3, Screenshot of BadData.



```
BadData.csv - Notepad
File Edit View

CustomerFirstName,CustomerLastName,CustomerEmail,CustomerPhone
Frank,Harris,frank.harris@email.com,7891234560
Grace,Miller,grace.miller@email.com,234-5678-901
Henry,Clark,henry.clark@email.com,567-8901-234
Karen,Anderson,karen.anderson@email.com,6789012345
Leo,Martinez,leo.martinez@email.com,901-2345-678
Nathan,Rodriguez,nathan.rodriguez@email.com,2233445566
Paul,Walker,paul.walker@email.com,445-566-77
```

4. Loading Data into SQL Server

4.1 Create Staging Table

To store the valid customer records, a staging table is created in SQL Server. The table schema ensures that customer data is structured correctly and can be efficiently queried for further processing.

Use TempDB;

Go

Create Table CustomersStaging

```
(
    CustomersStaging int identity Primary Key,
    CustomerFirstName nvarchar(100),
    CustomerLastName nvarchar(100),
    CustomerEmail nvarchar(100),
    CustomerPhone nvarchar(100)
);
```

4.2 Explanation of Staging Table

The table consists of the following columns:

- CustomersStaging: Auto-incremented primary key.
- CustomerFirstName: Stores the first name of the customer.
- CustomerLastName: Stores the last name of the customer.
- CustomerEmail: Stores the email address.
- CustomerPhone: Stores the validated phone number.

4.3 Screenshot of SQL Table Creation

Picture 4, screenshot of the tables creation in tempdb

The screenshot displays the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the database structure, including the tempdb database. The SQL script in the center creates two tables: CustStaging_ValidPhones and CustStaging_InvalidPhones. The Results pane at the bottom shows the data inserted into these tables.

```
use TempDB;
If (object_id('CustStaging_ValidPhones') is not null)
Drop Table CustStaging_ValidPhones;
Go

Create Table CustStaging_ValidPhones
(
    CustomersStaging int identity Primary Key,
    CustomerFirstName nvarchar(100),
    CustomerLastName nvarchar(100),
    CustomerEmail nvarchar(100),
    CustomerPhone nvarchar(100) );
Go

If (object_id('CustStaging_InvalidPhones') is not null)
Drop Table CustStaging_InvalidPhones;
Go

Create Table CustStaging_InvalidPhones
(
    CustomersStaging int identity Primary Key,
    CustomerFirstName nvarchar(100),
    CustomerLastName nvarchar(100),
    CustomerEmail nvarchar(100),
    CustomerPhone nvarchar(100) );
Go

Select * From CustStaging_ValidPhones;
Select * From CustStaging_InvalidPhones;
```

CustomersStaging	CustomerFirstName	CustomerLastName	CustomerEmail	CustomerPhone
1	Alice	Johnson	alice.johnson@email.com	123-456-7890
2	Bob	Smith	bob.smith@email.com	987-654-3210
3	Charlie	Brown	charlie.brown@email.com	456-789-1234
4	David	Wilson	david.wilson@email.com	654-321-0987

CustomersStaging	CustomerFirstName	CustomerLastName	CustomerEmail	CustomerPhone
1	Frank	Harris	frank.harris@email.com	789-123-4560
2	Grace	Miller	grace.miller@email.com	234-567-901
3	Henry	Clark	henry.clark@email.com	567-890-1234

Query executed successfully.

5. SSIS Package Setup

5.1 Overview of SSIS Package

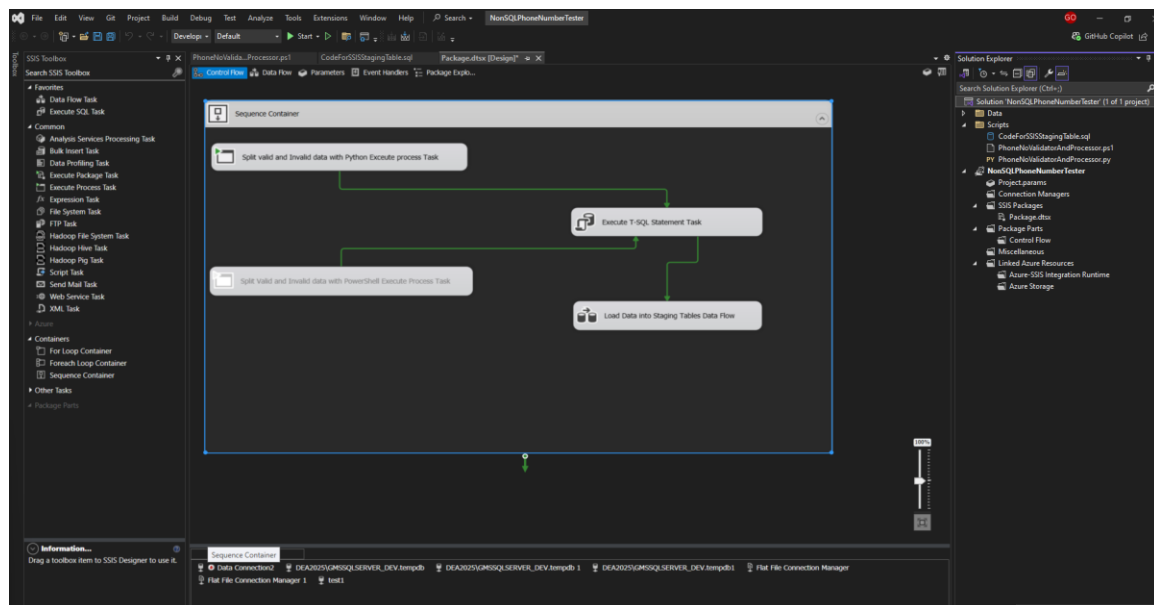
An SSIS (SQL Server Integration Services) package is used to automate the ETL process. The package consists of multiple tasks that execute sequentially to extract, validate, and load the data.

5.2 Execute Process Task

- Runs the validation process to filter valid and invalid phone numbers.
- Ensures only correctly formatted phone numbers proceed to the database.

5.3 Screenshot of SSIS Package Control Flow

Picture 5, screenshot of the SSIS Package Control

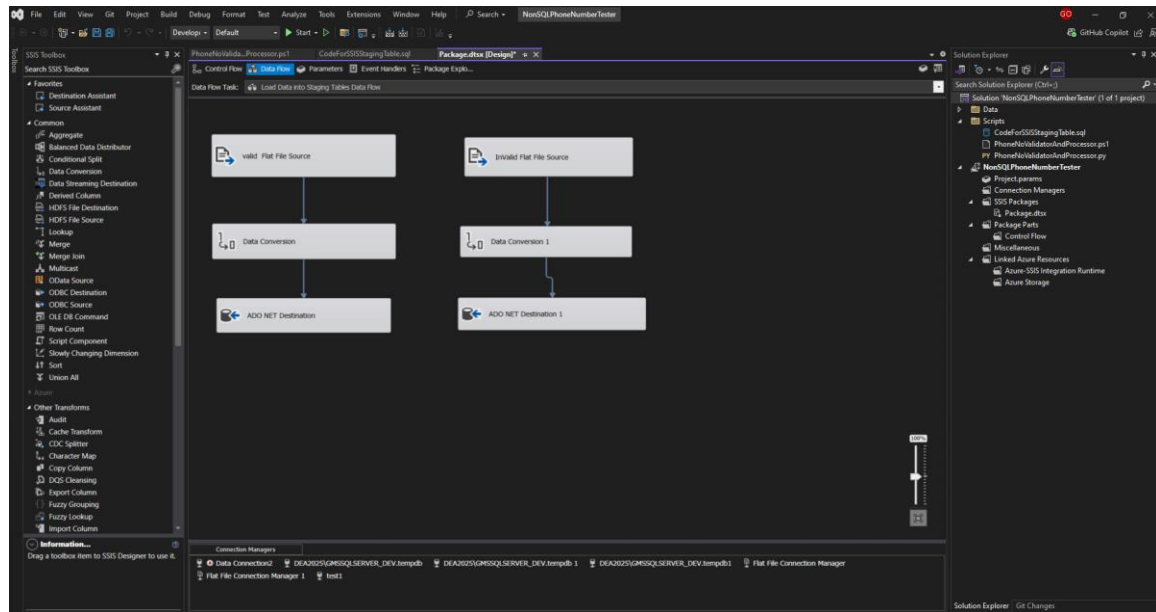


5.4 Data Flow Task

- Extracts data from ValidData.csv.
- Transforms and maps the data to match the staging table schema.
- Loads the processed data into SQL Server.

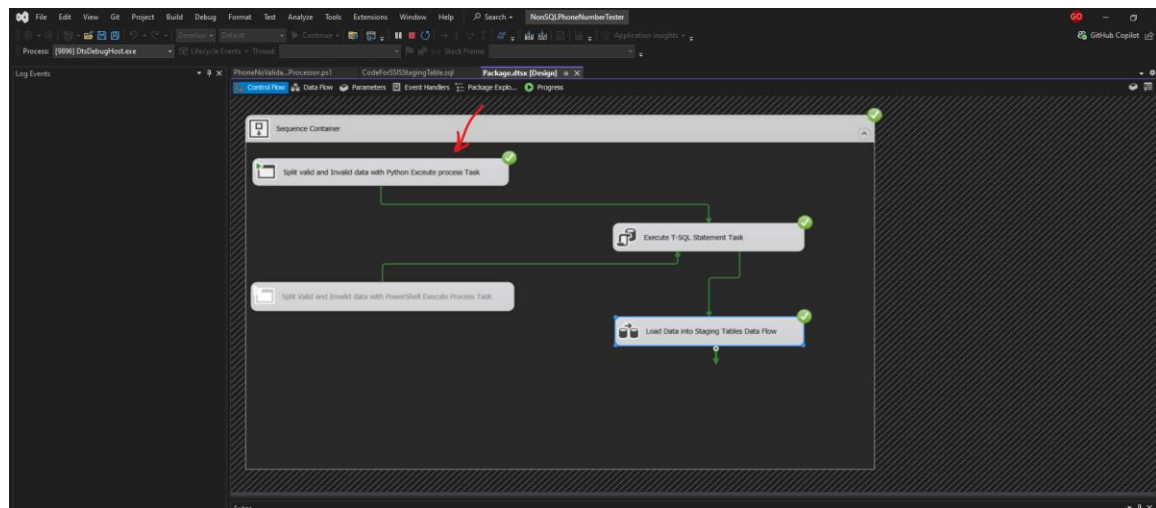
5.5 Screenshot of SSIS Package Data Flow

Picture 6, screenshot of the SSIS Package Data flow

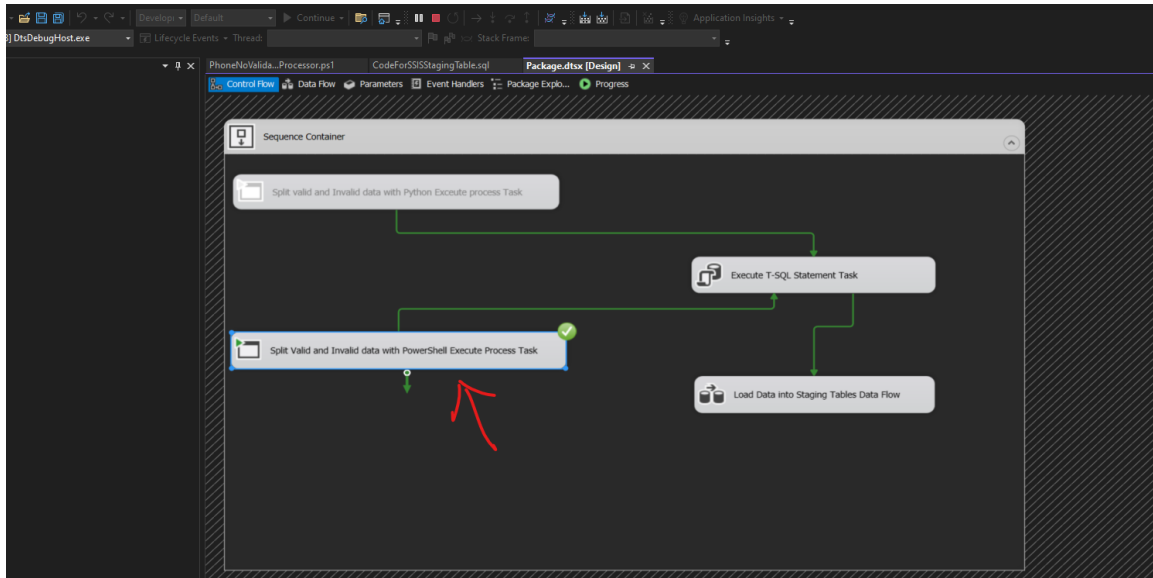


5.6 Entire SSIS Package Overview using Python and Power shell Execute process

Picture7, screenshot of the SSIS using Python Execute process



Picture 8, screenshot of the ssis using Powershell execute process.



Picture 9, screenshot of the tables showing valid and bad date

SSS ToolBox
Per tool windows can only be used by a SSS
event Integration Services package document.

PhoneNoValidator_Processor.ps1 CodeForSSISStagingTable.sql Package.dtsx [Design]

```

1  Set Tempdb
2  IF (Object_Id('CustStaging_ValidPhones') is not null)
3  Drop Table CustStaging_ValidPhones;
4
5
6
7  Create Table CustStaging_ValidPhones
8  (CustomerStaging int Identity Primary key
9  ,CustomerFirstName varchar(100)
10 ,CustomerLastName varchar(100)
11 ,CustomerEmail varchar(100)
12 ,CustomerPhone varchar(100) );
13
14
15 IF (Object_Id('CustStaging_InvalidPhones') is not null)
16 Drop Table CustStaging_InvalidPhones;
17
18
19 Create Table CustStaging_InvalidPhones
20 (CustomerStaging int Identity Primary key
21 ,CustomerFirstName varchar(100)
22 ,CustomerLastName varchar(100)
23 ,CustomerEmail varchar(100)
24 ,CustomerPhone varchar(100) );
25
26
27 Select * From CustStaging_ValidPhones;
28 Select * From CustStaging_InvalidPhones;

```

90 % No Issues Found

Results 3 Message

CustomerStaging	CustomerFirstName	CustomerLastName	CustomerEmail	CustomerPhone
1	John	Johnson	john.johnson@gmail.com	412-456-7890
2	Bob	Smith	bob.smith@gmail.com	887-654-3210
3	Charlie	Brown	charlie.brown@gmail.com	456-789-1234
4	David	Wilson	david.wilson@gmail.com	854-321-9876
5	Eve	Thompson	eve.thompson@gmail.com	321-654-4321
6	Frank	Miller	frank.miller@gmail.com	987-654-3210
7	Grace	Lee	grace.lee@gmail.com	567-890-1234
8	Henry	Clark	henry.clark@gmail.com	234-567-8901
9	Ivan	Anderson	ivan.anderson@gmail.com	789-012-3456
10	Jane	Martinez	jane.martinez@gmail.com	654-321-0987
11	Kevin	Roberts	kevin.roberts@gmail.com	123-456-7890
12	Laura	White	laura.white@gmail.com	456-789-0123

CustomerStaging	CustomerFirstName	CustomerLastName	CustomerEmail	CustomerPhone
1	Frank	Harris	frank.harris@gmail.com	789-123-4567
2	Grace	Miller	grace.miller@gmail.com	234-567-8901
3	Henry	Clark	henry.clark@gmail.com	567-890-1234
4	Ivan	Anderson	ivan.anderson@gmail.com	890-123-4567
5	Jane	Martinez	jane.martinez@gmail.com	123-456-7890
6	Kevin	Roberts	kevin.roberts@gmail.com	456-789-0123
7	Laura	White	laura.white@gmail.com	789-012-3456

Query executed successfully at 1:55:54 PM

DEAD2025\MSQLSERVER_DEV (1) - DEAD2025\gsmn (3) Tempdb | 00:00:01 | 20 rows

Solution Explorer - Get Changes

7. Summary

This document serves as a comprehensive guide to the ETL process for validating and loading phone number data. By implementing this ETL workflow, organizations can maintain high data integrity, streamline customer data management, and reduce errors in communication and reporting.

By following the structured steps outlined in this document, users can effectively manage and troubleshoot the ETL process, ensuring seamless execution and data accuracy. The use of SSIS enhances automation, minimizing manual intervention and improving efficiency.

This document is designed to be a reference guide for new hires, ETL developers, and support technicians who work with data validation and integration tasks. Detailed explanations, along with screenshots, ensure clarity and ease of use, making troubleshooting and implementation straightforward.