LAB : SSIS Staging with Transformations

Complete Example: sales.customers → stg.customers

Duration: 30 minutes
What you'll learn: Extract → Transform → Load with real transformations

---

Step 1: Update Staging Table Structure

First, add audit columns to your staging table

In SSMS:

sql
```
USE BikeStores_DW;
GO

ALTER TABLE stg.customers
ADD
    LoadDate DATETIME,
    SourceSystem VARCHAR(50);
GO

SELECT TOP 0 * FROM stg.customers;
```

Step 2: Create SSIS Project

A. Create Project

1. **Open Visual Studio**
2. File → New → Project
3. Search: "Integration Services"
4. Select: "Integration Services Project"
5. Name: `CustomerStagingDemo`
6. Location: Choose your folder
7. Click: Create

B. Rename Package

1. In Solution Explorer (right side), see `Package.dtsx`

2. Right-click `Package.dtsx` → Rename
3. Type: `Load_Customer_Staging.dtsx`
4. Press Enter

C. Create Connections

Connection 1: Source (BikeStores)

1. Connect to the bikestore database (previous one with data)

Connection 2: Target (Data Warehouse)

1. Destination is the Bikestore_DW (the one with staging/fact/demension table)

Step 3: Build the Complete Data Flow

A. Add Data Flow Task (2 min)

1. Look at left panel: **SSIS Toolbox**
2. Make sure you're on **Control Flow** tab (bottom of design surface)
3. From toolbox, drag **Data Flow Task** to design surface
4. Double-click the task name → Type: `Load Customer with Transformations`
5. Press Enter

B. Enter Data Flow

1. **Double-click** the Data Flow Task box
2. You'll switch to **Data Flow** tab
3. Design surface is now empty
4. SSIS Toolbox shows different components

C. Add Source (3 min)

1. From SSIS Toolbox → **Other Sources** → Drag **ADO NET Source** to design surface
2. **Double-click** ADO NET Source
3. Configure:
    - **Connection manager:** S
    - **Data access mode:** Table or view
    - **Table or view:** Click dropdown → Select `[sales].[customers]`
4. Click: **Preview...** button
    - Should see customer data
    - Click: Close
5. Click: **OK**

D. Add Transformation 1: Derived Column (Clean Data)

What we're doing: Clean names, standardize email, clean phone, add audit columns

1. From SSIS Toolbox → **Common** → Drag **Derived Column** to design surface
2. **Position it below** the ADO NET Source
3. **Drag the blue arrow** from ADO NET Source down to Derived Column
    - Click source → drag from bottom blue arrow to Derived Column
4. **Double-click** Derived Column transform
5. You'll see a table with columns. Add these new derived columns:

Row 1: Clean First Name

Field	Value
Derived Column Name	first_name
Derived Column	replace
Expression	`TRIM(first_name)`

Row 2: Clean Last Name

Field	Value
Derived Column Name	last_name
Derived Column	replace
Expression	`UPPER(TRIM(last_name))`

Row 3: Clean Email

Field	Value
Derived Column Name	email

Derived Column	replace
Expression	`LOWER(TRIM(email))`

Row 5: Add LoadDate

Field	Value
Derived Column Name	LoadDate
Derived Column	add as new column
Expression	`GETDATE()`
Data Type	database timestamp [DT_DBTIMESTAMP]

Row 6: Add SourceSystem

Field	Value
Derived Column Name	SourceSystem
Derived Column	add as new column
Expression	`"BikeStores"`

E. Add Transformation 2: Conditional Split (Validate)

What we're doing: Only load customers with valid email and names

1. From SSIS Toolbox → **Common** → Drag **Conditional Split** to design surface
2. Position it below Derived Column
3. **Drag blue arrow** from Derived Column to Conditional Split
4. **Double-click** Conditional Split
5. Configure the condition:

Add this condition:

Field	Value

Output Name	ValidCustomers
Condition	`!ISNULL(email) && !ISNULL(first_name) && !ISNULL(last_name)`
Order	1

How to build the condition:

- In the condition box, type the expression above
- This means: email, first name, and last name must all have values

6. Leave **Default output name** as: `Conditional Split Default Output`
7. Click: **OK**

✅ **Checkpoint:** Conditional Split will filter valid customers

---

F. Add Destination (4 min)

1. From SSIS Toolbox → **Other Destinations** → Drag **ADO NET Destination**
2. Position it below Conditional Split
3. **Drag blue arrow** from Conditional Split to ADO NET Destination
    - A popup appears: "Input Output Selection"
    - Select: **ValidCustomers**
    - Click: OK
4. **Double-click** ADO NET Destination
5. Send to the Stg_customer table in Bikestore_DW

IMPORTANT: Map the CLEAN columns to staging, not the original ones!

How to map:

- Left side shows available columns from previous transforms
- Right side shows stg.customers columns
- Draw lines OR use dropdown
- **Unmap** any auto-mapped "dirty" columns (first_name, last_name, phone, email)
- **Map** the _clean versions instead

8. Click: **OK**

✅ **Checkpoint:** Destination configured with clean data!

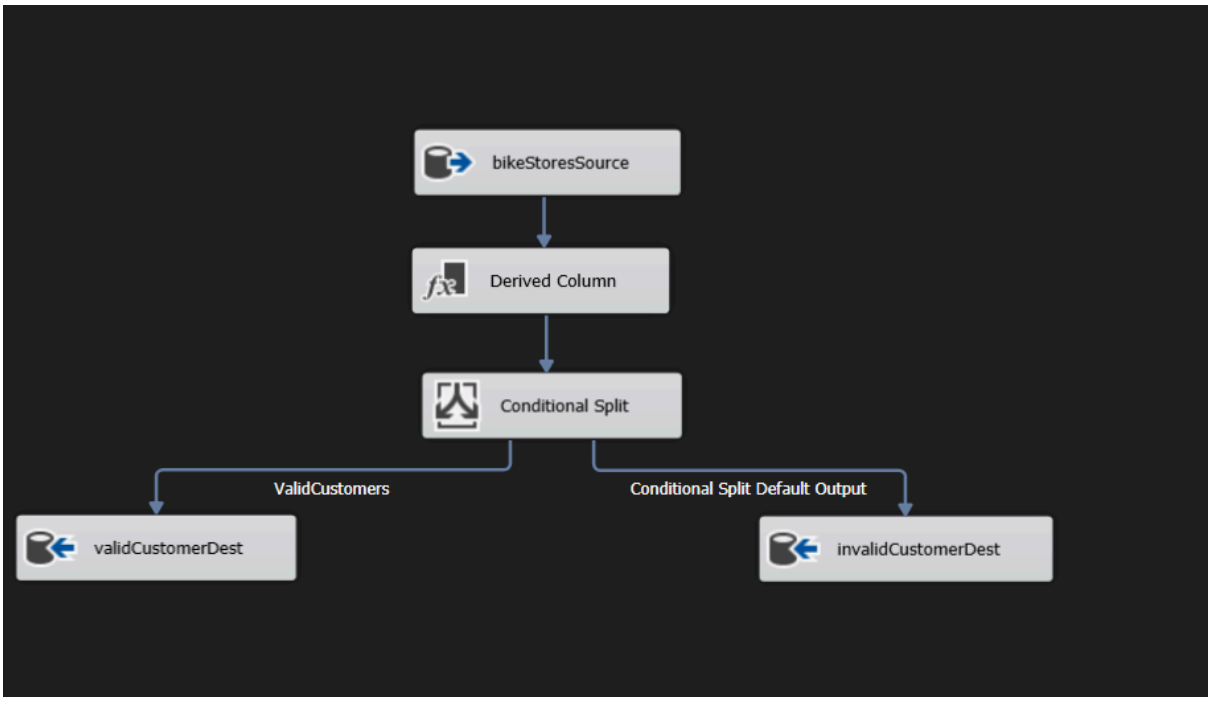
G. Add Error Output (2 min)

Handle invalid customers that don't pass validation

1. **Drag another ADO NET Destination** to design surface
2. Position it to the **right** of Conditional Split
3. **Right-click** on Conditional Split
4. Select: **Edit...**
5. Click: **OK** (we already have our condition)
6. Now **drag blue arrow** from Conditional Split to the NEW destination
7. In popup, select: **Conditional Split Default Output**
8. Click: OK
9. **Double-click** this new destination
10. Create new table to sent the default output in the BikeStores_DW

Create error table in SSMS or you can create in SSIS:

```sql
CREATE TABLE stg.customers_errors (
    customer_id INT,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    phone VARCHAR(25),
    LoadDate DATETIME,
    ErrorReason VARCHAR(200)
);
```



Step 4: Test the Package (5 min)

A. Run the Package

ASSIGNMENT

BUILD and load the data in SSMS from SSIS for all the staging tables.