# SSIS Lookup Transformation

## 1. Introduction

Definition:  
The Lookup Transformation in SSIS searches for matching records in a reference dataset (table, view, or query) based on specified keys. It can:  
- Return matching records (Matched Output)  
- Return non-matching records (No Match Output)  
- Perform additional actions like adding/replacing columns  
  
Key Capabilities:  
- Join incoming rows from a source with reference data  
- Handle both match and no match scenarios  
- Integrate with OLE DB, Cache, or other lookup sources  
  
Common Use Cases:  
- Detecting and inserting only new records  
- Updating existing records based on changes  
- Data validation before insertion

## 2. Example Scenario

We will:  
- Use AdventureWorks database  
- Query product and review details as the Source  
- Compare against a ProductReports table  
- Insert only those records not present in the destination table (based on ProductID and optionally other columns)

## 3. Step-by-Step Implementation

### Step 1 – Prepare the Destination Table

- Use a SELECT INTO or CREATE TABLE statement to create ProductReports in the target database.

- Example:  
SELECT ProductID, Name, Color, Review  
INTO ProductReports  
FROM <joined-tables-query>  
WHERE 1=0; -- creates structure without data

### Step 2 – Create a New SSIS Project

- Open Visual Studio with SQL Server Integration Services Projects extension.

- Create a new project → Name it, e.g., LookupExample.

- Rename the default package for clarity.

### Step 3 – Add a Data Flow Task

- In Control Flow, drag a Data Flow Task.

- Double-click to open the Data Flow tab.

### Step 4 – Configure the Source

- Drag OLE DB Source to Data Flow.

- Configure with:  
 - Connection: AdventureWorks DB  
 - Data Access Mode: SQL Command  
 - SQL Query: Your join query returning product info

- Preview to confirm expected columns.

### Step 5 – Add the Lookup Transformation

- From Common in SSIS Toolbox, drag Lookup into the Data Flow.

- Connect OLE DB Source → Lookup.

- Lookup Settings:  
 - Full Cache (default) is fine for small datasets.  
 - Use OLE DB Connection Manager to connect to the destination DB.

- Specify how to handle no matches:  
 - Change from Fail component to Redirect rows to no match output.

### Step 6 – Configure the Lookup Reference Table

- In Connection tab, select ProductReports table.

- Preview to verify contents.

### Step 7 – Map Columns

- In Columns tab, drag ProductID from Available Input Columns to Available Lookup Columns.

- (Optional) Add more match columns like Color for composite match conditions.

- Decide whether to:  
 - Add lookup columns as new columns in the data flow  
 - Replace existing columns from the source

### Step 8 – Configure the Destination for No Match Records

- Drag OLE DB Destination to Data Flow.

- Connect the No Match Output from Lookup to the destination.

- Configure:  
 - Connection: Same DB as reference table (or different if required)  
 - Table: ProductReports  
 - Mappings: Ensure columns align

### Step 9 – Execute the Package

- Run the package.

- Verify:  
 - First run: Inserts all source records not in destination.  
 - Second run: Inserts only newly missing records.

## 4. Testing Additional Match Conditions

Modify the Lookup to match on multiple columns (e.g., ProductID and Color). Change a value in the destination so it no longer matches both fields → Rerun to confirm that only the changed record is inserted.

## 5. Handling Matched Records

Lookup has two outputs:  
- Match Output → Rows that match the reference table  
- No Match Output → Rows that don’t match  
You can process matched rows separately (e.g., log them, update them, or export to file). Example: Connect Match Output to another destination to store matched rows in a log table.

## 6. Key Points to Remember

- Caching Modes:  
 - Full Cache: Loads all reference data upfront  
 - Partial Cache: Loads only required rows as needed  
 - No Cache: Queries reference data for every input row  
- Performance Tip: Use Full Cache for small/medium datasets and indexed lookup columns.  
- Error Handling: Use the Redirect rows option to avoid package failure when no matches occur.  
- Multiple Match Keys: Improves data integrity but can increase complexity.