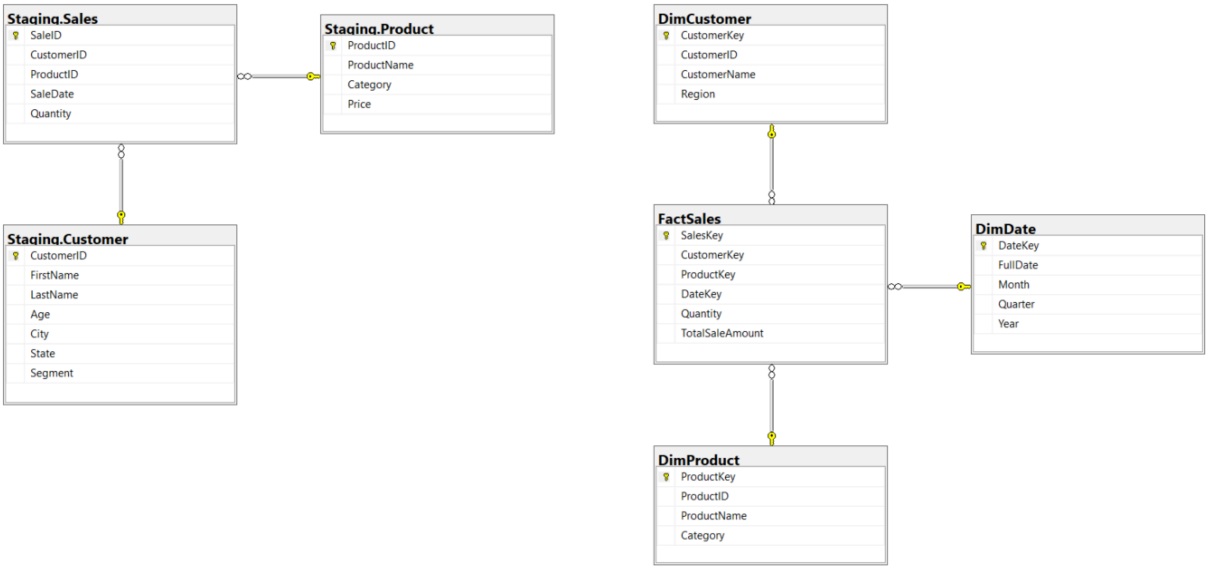
Cap Proj: Staging to Warehouse Deliverable

Started with the steps, created a database, then as requested make tables through a query executable.

1. Create tables
    a. Create Staging tables - Staging is just raw data so the tables we'll be making will only fit with the data's structure.
        i. Customer
        ii. Product
        iii. Sales
    b. Create DIM tables
        i. Customer
        ii. Product
        iii. Date
        iv. FactSales
    c. Create Fact table - The table needs some data from a few spots, Price from total sales amount.
        i. SalesKey - From SalesID into SalesKey?
        ii. CustomerKey - DimCustomer
        iii. ProductKey - DimProduct
        iv. DateKey - DimDate
        v. Quantity - Staging Sales

vi. TotalSalesAmount - Quantity multiply by price Staging.Sales and Staging.Product

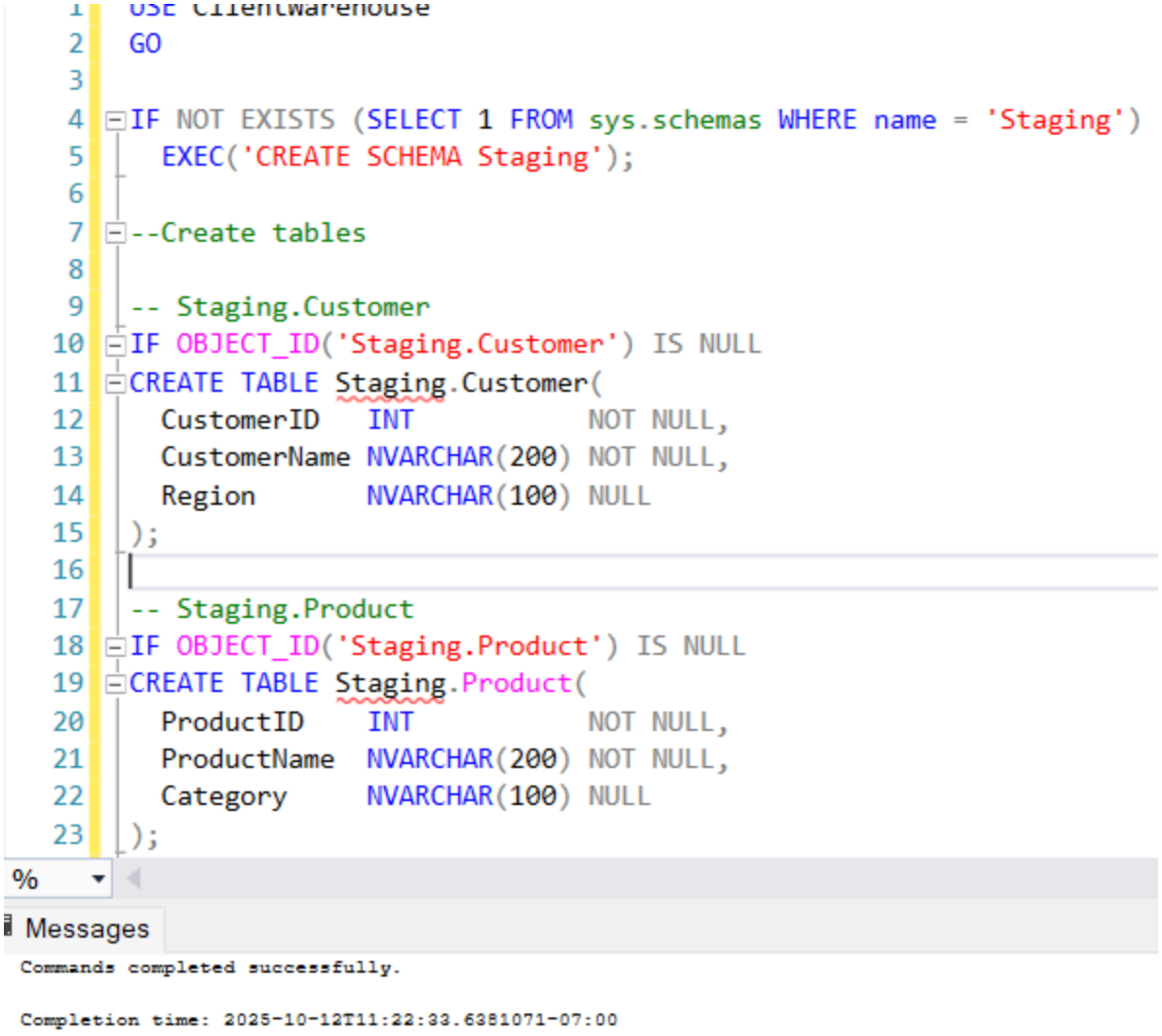


Tables and their connections

2. Revisit data flow

    a. Add a to table task after each clean file creation to push into DIM tables

        i. Requires creation of Staging and import of clean csv from last week



SQL code to make tables

```
USE ClientWarehouse;

GO


/* Drop Tables if already created */

IF OBJECT_ID('dbo.FactSales','U') IS NOT NULL DROP TABLE dbo.FactSales;

IF OBJECT_ID('dbo.DimDate','U')     IS NOT NULL DROP TABLE dbo.DimDate;
```

```
IF OBJECT_ID('dbo.DimProduct','U')  IS NOT NULL DROP TABLE dbo.DimProduct;
IF OBJECT_ID('dbo.DimCustomer','U') IS NOT NULL DROP TABLE
dbo.DimCustomer;
GO

/* Dimension tables */
CREATE TABLE dbo.DimCustomer
(
   CustomerKey     INT IDENTITY(1,1) NOT NULL
      CONSTRAINT PK_DimCustomer PRIMARY KEY CLUSTERED,
   CustomerID      NVARCHAR(50)  NOT NULL
      CONSTRAINT UQ_DimCustomer_CustomerID UNIQUE,
   CustomerName    NVARCHAR(200) NOT NULL,
   Region          NVARCHAR(100) NULL
);
GO

CREATE TABLE dbo.DimProduct
(
   ProductKey      INT IDENTITY(1,1) NOT NULL
      CONSTRAINT PK_DimProduct PRIMARY KEY CLUSTERED,
   ProductID       NVARCHAR(50)  NOT NULL
```

```
    CONSTRAINT UQ_DimProduct_ProductID UNIQUE,
  ProductName    NVARCHAR(200) NOT NULL,
  Category       NVARCHAR(100) NULL
);
GO

CREATE TABLE dbo.DimDate
(
  DateKey    INT        NOT NULL
    CONSTRAINT PK_DimDate PRIMARY KEY CLUSTERED,
  FullDate   DATE       NOT NULL
    CONSTRAINT UQ_DimDate_FullDate UNIQUE,
  [Month]    TINYINT    NOT NULL
    CONSTRAINT CK_DimDate_Month CHECK ([Month] BETWEEN 1 AND 12),
  [Quarter]  TINYINT    NOT NULL
    CONSTRAINT CK_DimDate_Quarter CHECK ([Quarter] BETWEEN 1 AND 4),
  [Year]     SMALLINT   NOT NULL
);
GO

/* Fact table */
CREATE TABLE dbo.FactSales
```

```
(
    SalesKey        BIGINT IDENTITY(1,1) NOT NULL
        CONSTRAINT PK_FactSales PRIMARY KEY CLUSTERED,
    CustomerKey     INT     NOT NULL,
    ProductKey      INT     NOT NULL,
    DateKey         INT     NOT NULL,
    Quantity        INT     NOT NULL
        CONSTRAINT CK_FactSales_Quantity CHECK (Quantity >= 0),
    TotalSaleAmount DECIMAL(19,4) NOT NULL
        CONSTRAINT CK_FactSales_TotalAmt CHECK (TotalSaleAmount >= 0),

    CONSTRAINT FK_FactSales_Customer
        FOREIGN KEY (CustomerKey) REFERENCES dbo.DimCustomer(CustomerKey),
    CONSTRAINT FK_FactSales_Product
        FOREIGN KEY (ProductKey)  REFERENCES dbo.DimProduct(ProductKey),
    CONSTRAINT FK_FactSales_Date
        FOREIGN KEY (DateKey)     REFERENCES dbo.DimDate(DateKey)
);
```

```sql
USE ClientWarehouse;
GO

/* Drop Tables if already created */
IF OBJECT_ID('dbo.FactSales','U') IS NOT NULL DROP TABLE dbo.FactSales;
IF OBJECT_ID('dbo.DimDate','U')     IS NOT NULL DROP TABLE dbo.DimDate;
IF OBJECT_ID('dbo.DimProduct','U')  IS NOT NULL DROP TABLE dbo.DimProduct;
IF OBJECT_ID('dbo.DimCustomer','U') IS NOT NULL DROP TABLE dbo.DimCustomer;
GO

/* Dimension tables */
CREATE TABLE dbo.DimCustomer
(
    CustomerKey     INT IDENTITY(1,1) NOT NULL
        CONSTRAINT PK_DimCustomer PRIMARY KEY CLUSTERED,
    CustomerID      NVARCHAR(50)  NOT NULL
        CONSTRAINT UQ_DimCustomer_CustomerID UNIQUE,
    CustomerName    NVARCHAR(200) NOT NULL,
```

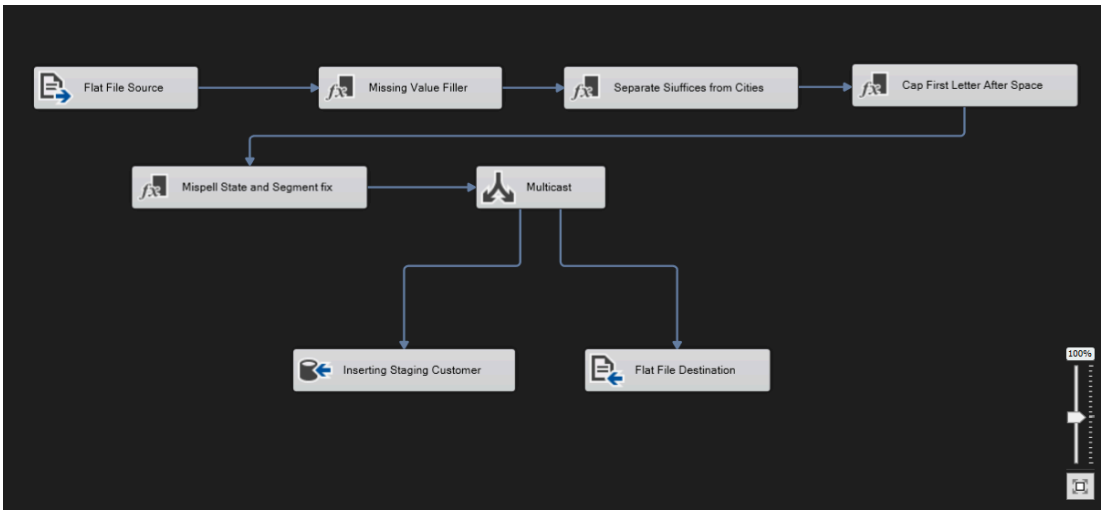
100 %

Messages

```
Commands completed successfully.

Completion time: 2025-10-11T15:58:55.9964969-07:00
```

Extending SSIS package



Extended, added a multi cast to preserve the clean csv creation, then data conversion to match the proper data type in the SSMS.

SQLQuery5.sql - (...WGMOTH\narze (64)) Yawgmoth\LOCAL

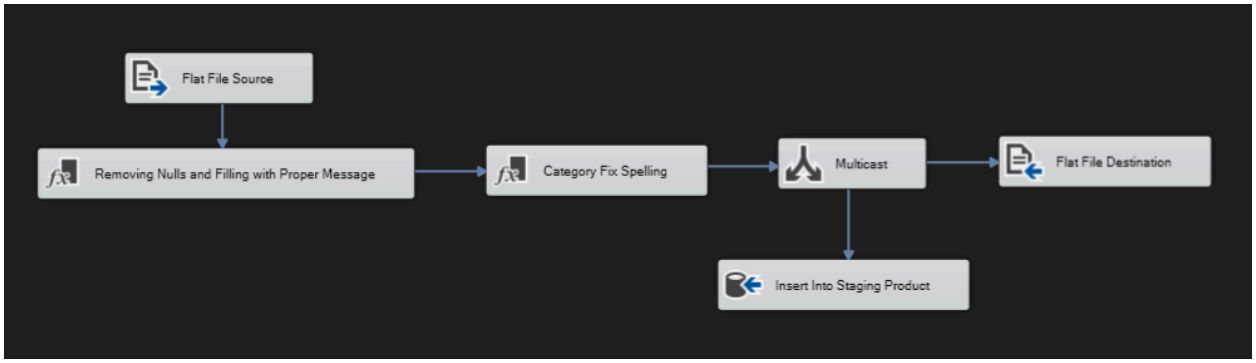
```
SELECT TOP (1000) [CustomerID]
      ,[CustomerName]
      ,[Region]
  FROM [ClientWarehouse].[Staging].[Customer]
```

91 %

Results Messages

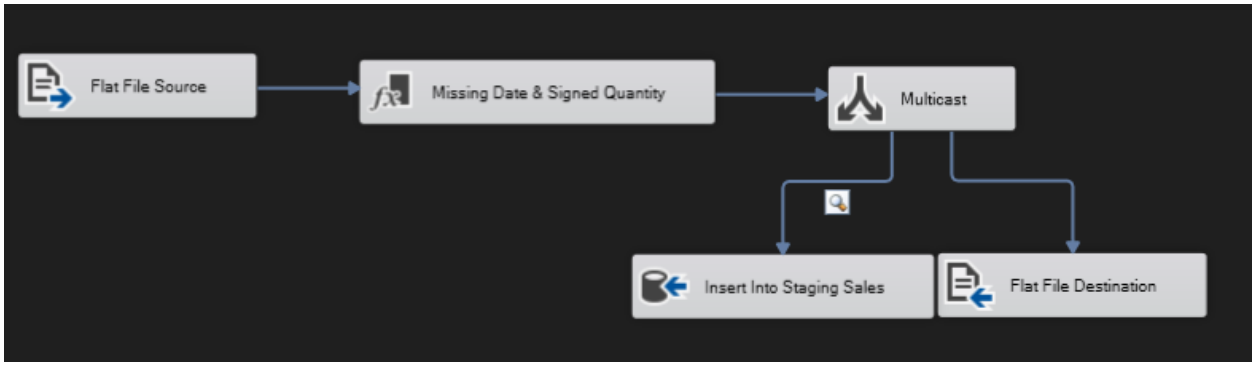
	CustomerID	CustomerName	Region
1	1	Jacob	WA
2	2	Joshua	WA
3	3	Jesse	CA
4	4	Amanda	CA
5	5	Diane	CA
6	6	Dawn	CA
7	7	Mark	CA
8	8	Krystal	WA
9	9	Amanda	CA
10	10	Christina	OR
11	11	Jesse	Missing State
12	12	Lisa	WA
13	13	Janice	OR
14	14	Phillip	OR
15	15	Mark	CA

Successful clean and insertion from SSIS into staging.customer of SSMS.

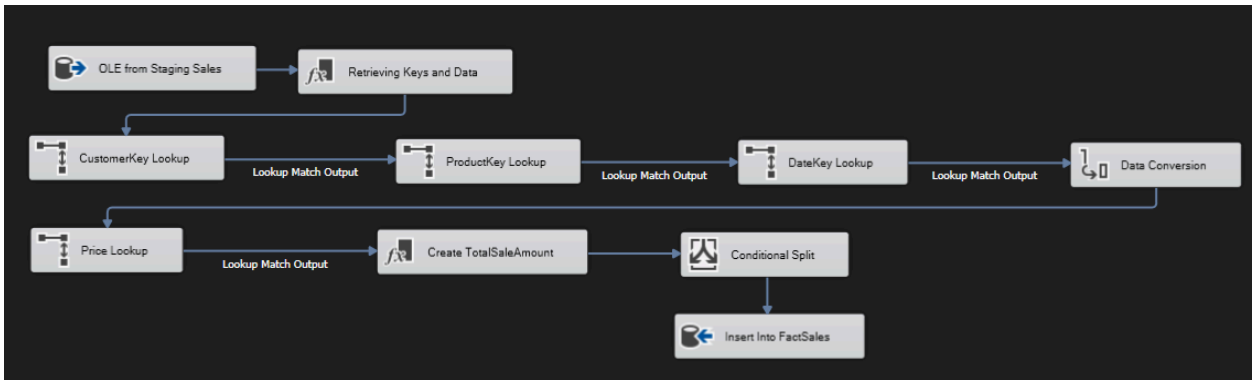


Similar issue with data conversion, SSMS like DT_WSTR, but I had it set to string.

Added Multicast and OLE DB DESTINATION



Staging sales had no issues



FactTable with all the Lookups

Staging into Dim

Staging.Customer doesn't have Customer name, it has First and Last name so a derived column is added to make CustomerName.

Results Messages

	CustomerKey	CustomerID	CustomerName	Region
1	1	1	Jacob Sellers	WA
2	2	2	Joshua Johnson	WA
3	3	3	Jesse Berger	CA
4	4	4	Amanda Mann	CA
5	5	5	Diane Rasmussen	CA
6	6	6	Dawn Wells	CA
7	7	7	Mark Hanson	CA
8	8	8	Krystal Taylor	WA
9	9	9	Amanda Gordon	CA
10	10	10	Christina Thornton	OR
11	11	11	Jesse Ramos	Missing State
12	12	12	Lisa Stone	WA
13	13	13	Janice Willis	OR

Success in Merging First and Last name for CustomerName Column

90 %

Results Messages

	ProductKey	ProductID	ProductName	Category
1	1	1	Call 865	Missing Category
2	2	2	Truth 306	Electronics
3	3	3	Management 741	Furniture
4	4	4	Team 470	Furniture
5	5	5	Perhaps 594	Electronics
6	6	6	Few 937	Home Appliances
7	7	7	Head 154	Furniture
8	8	8	Term 882	Home Appliances
9	9	9	Teach 594	Electronics
10	10	10	Argue 623	Home Appliances
11	11	11	Huge 306	Home Appliances
12	12	12	Common 356	Electronics

Successful insert into DimProduct

```sql
USE ClientWarehouse;
GO

;WITH d AS
(
  SELECT DISTINCT CAST(SaleDate AS date) AS FullDate
  FROM [dbo].[Staging.Sales]
  WHERE SaleDate IS NOT NULL
)
INSERT INTO [dbo].[DimDate] (DateKey, FullDate, [Year], [Month], [Quarter])
SELECT
    (YEAR(FullDate) * 10000) + (MONTH(FullDate) * 100) + DAY(FullDate) AS DateKey
    FullDate,
    CONVERT(smallint, YEAR(FullDate))                          AS [Year],
    CONVERT(tinyint,  MONTH(FullDate))                          AS [Month],
    CONVERT(tinyint, ((MONTH(FullDate) - 1) / 3) + 1)          AS [Quarter]
FROM d AS s
WHERE NOT EXISTS
(
  SELECT 1
  FROM [dbo].[DimDate] x
  WHERE x.FullDate = s.FullDate
);
```

0 %

Messages

```
(118 rows affected)

Completion time: 2025-10-13T21:16:00.6435673-07:00
```

Completing the DimDate so we can put together the FactSales table in SSIS.

	DateKey	FullDate	Month	Quarter	Year
1	20000101	2000-01-01	1	1	2000
2	20240907	2024-09-07	9	3	2024
3	20240915	2024-09-15	9	3	2024
4	20240916	2024-09-16	9	3	2024
5	20240917	2024-09-17	9	3	2024
6	20240921	2024-09-21	9	3	2024
7	20240927	2024-09-27	9	3	2024
8	20240928	2024-09-28	9	3	2024
9	20240930	2024-09-30	9	3	2024
10	20241005	2024-10-05	10	4	2024

Successful making DimDate

	SalesKey	CustomerKey	ProductKey	DateKey	Quantity	TotalSaleAmount
1	1	150	75	20250331	16	14608.0000
2	2	66	46	20241205	1	852.0000
3	3	77	132	20250325	1	151.0000
4	4	130	107	20241023	20	0.0000
5	5	129	30	20241120	25	11175.0000
6	6	145	11	20000101	47	45308.0000
7	7	143	66	20250504	1	846.0000
8	8	153	36	20241210	46	4692.0000
9	9	37	3	20250221	1	673.0000
10	10	87	26	20250228	12	11484.0000
11	11	44	142	20250611	18	10944.0000

Successful Insertion of FactSales

Consultant Memo

DATE: October 13, 2025

TO: Project Sponsor, ClientWarehouse DW

FROM: Khai Ha, ETL Consultant

RE: Implementation of SSIS Pipeline for Staging → Dimensions → FactSales

Thank you for the opportunity to work with your team on the first release of your analytics warehouse. Per your request, we evaluated the raw flat-file loads, designed/implemented SSIS dataflows to cleanse and stage the data, and completed the star-schema loads for DimCustomer, DimProduct, DimDate, and FactSales. The deliverable is a re-runnable package that prevents duplicate dimension members, enforces business rules (e.g., Quantity > 0), and loads a validated FactSales with correct foreign keys and computed TotalSaleAmount.

Executive Summary (Key Conclusions)

1. **Data model is live and consistent**. All dimensions load without duplicates; FactSales rows contain valid CustomerKey, ProductKey, and DateKey values and a calculated TotalSaleAmount = Price × Quantity.

2. **Pipeline is resilient and re-runnable.** Lookups on business keys (CustomerID, ProductID) allow re-execution without duplicating dim rows. Error outputs capture “no match” cases instead of failing the job.
3. **Quality rules are enforced.** Dates are normalized; Quantity <= 0 is corrected to 1 to satisfy the table constraint; malformed/missing keys and missing prices are quarantined for review.
4. **Next priority:** formalize reject handling and (optionally) enable updates to dimension attributes when they change.

What We Built (Overview of the SSIS Package)

Control Flow

- Load Staging tables from flat files (types aligned to SQL Server).
- Load DimCustomer and DimProduct (insert-only via Lookup to prevent duplicates).
- Populate DimDate from distinct SaleDate values (produces DateKey yyyymmdd, FullDate, Year, Month, Quarter).
- Load FactSales after dimensions are ready.

Core Dataflows

- Staging → Dims

- **Customer:** build CustomerName = FirstName + ' ' + LastName; map State → Region; Lookup on CustomerID; insert only new members.
- **Product:** carry ProductName, Category; Lookup on ProductID; insert only new members.
- **Date:** insert distinct dates with DateKey = YEAR*10000 + MONTH*100 + DAY and Quarter = ((Month-1)/3)+1.
- **Staging.Sales → FactSales**
    1. **Source**: single OLE DB Source from Staging.Sales.
    2. **Derived Columns:**
        - cast CustomerID/ProductID to DT_WSTR(50) for lookups,
        - DateKey = YEAR(SaleDate)*10000 + MONTH*100 + DAY,
        - Quantity = (Quantity <= 0 ? 1 : Quantity).
    3. **Lookups:**
        - DimCustomer → CustomerKey (No-Match → rejects)
        - DimProduct → ProductKey (No-Match → rejects)
        - DimDate → validate DateKey (No-Match → rejects)
        - Staging.Product → Price for measure (No-Match → rejects)
    4. **Measure:** TotalSaleAmount = Price * Quantity cast to decimal(19,4).
    5. **Conditional Split:** only rows with non-NULL keys and price flow to destination.
    6. **Destination:** FactSales via FastLoad; do not map identity SalesKey.

Results & Validation

- **Dim tables populated** with unique business keys.
- **FactSales populated** with sample rows such as: (SalesKey, CustomerKey, ProductKey, DateKey, Quantity, TotalSaleAmount) → (1,150,75,20250331,16,14608.0000) etc.
- **Sanity checks** performed:
    - Recomputed Price × Quantity equals TotalSaleAmount.
    - Left joins to dims return zero missing keys.
    - Quantity contains no values ≤ 0 after transformation.

Risks / Assumptions

- Price at time of sale is sourced from Staging.Product. If historical pricing is required (e.g., SCD-2), we should persist price snapshots or implement effective-dated products.
- Current load inserts only to dims. If source attributes change, dimension updates are not yet applied.

Recommendations (Next Steps)

1. Reject management: Persist all No-Match outputs to tables with error codes, row lineage, and load timestamp; include a daily review process.

2. Upsert enhancements for dims: Add update path (Conditional Split + OLE DB Command) to refresh CustomerName/Region and ProductName/Category when they change.
3. Scheduling & observability: Wrap packages with logging (row counts, timings), create SQL Agent schedules, and add alerts on non-zero rejects.
4. Data dictionary: Document column definitions, derivations (DateKey, TotalSaleAmount), and constraints (Quantity > 0) for BI consumers.
5. Performance tuning (as data grows): Consider partial-cache lookups, incremental staging, and surrogate-key caching if row volume increases.

Deliverables Provided

- SSIS solution with dataflows described above.
- SQL scripts for DimDate generation and post-load validation.
- This memo is an implementation summary and runbook direction.

Please let me know if you'd like me to enable dimension updates or formalize the reject pipelines next.