Database: WideWorldImporters

Wide World Importers is a global wholesale and retail distribution company that specializes in sourcing and delivering a wide range of goods—from novelty items to everyday products—to customers around the world. It manages relationships with suppliers, processes customer orders, handles logistics and inventory, and ensures timely delivery through an efficient supply chain network.

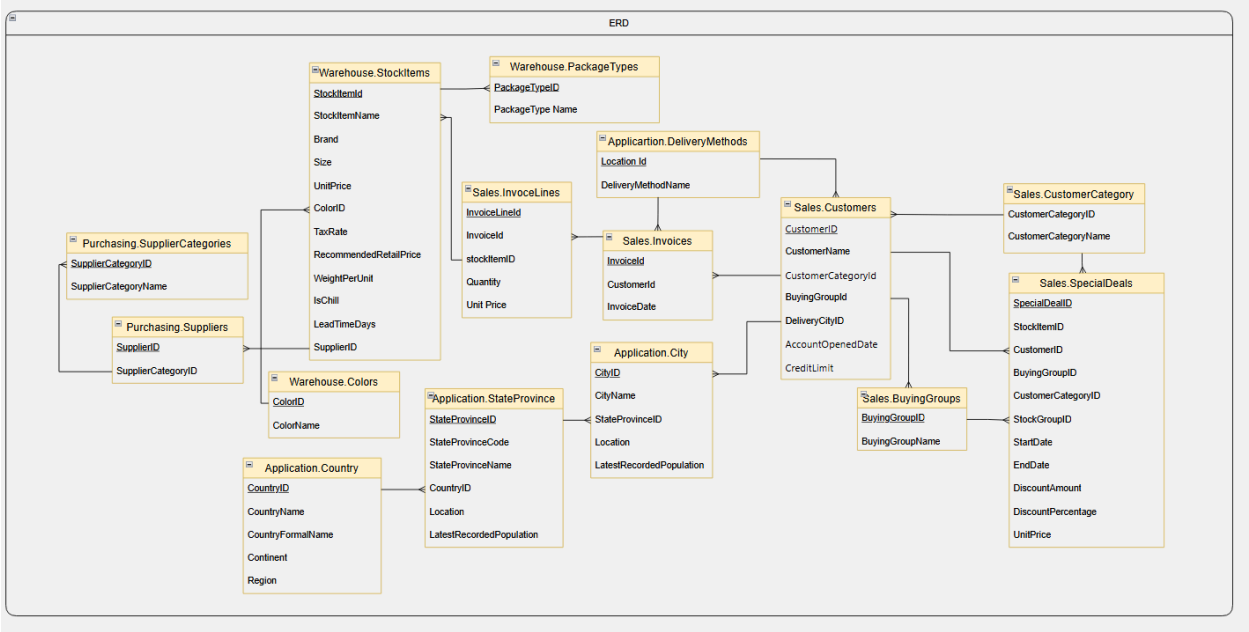
Business Area: Sales

Business Goals:

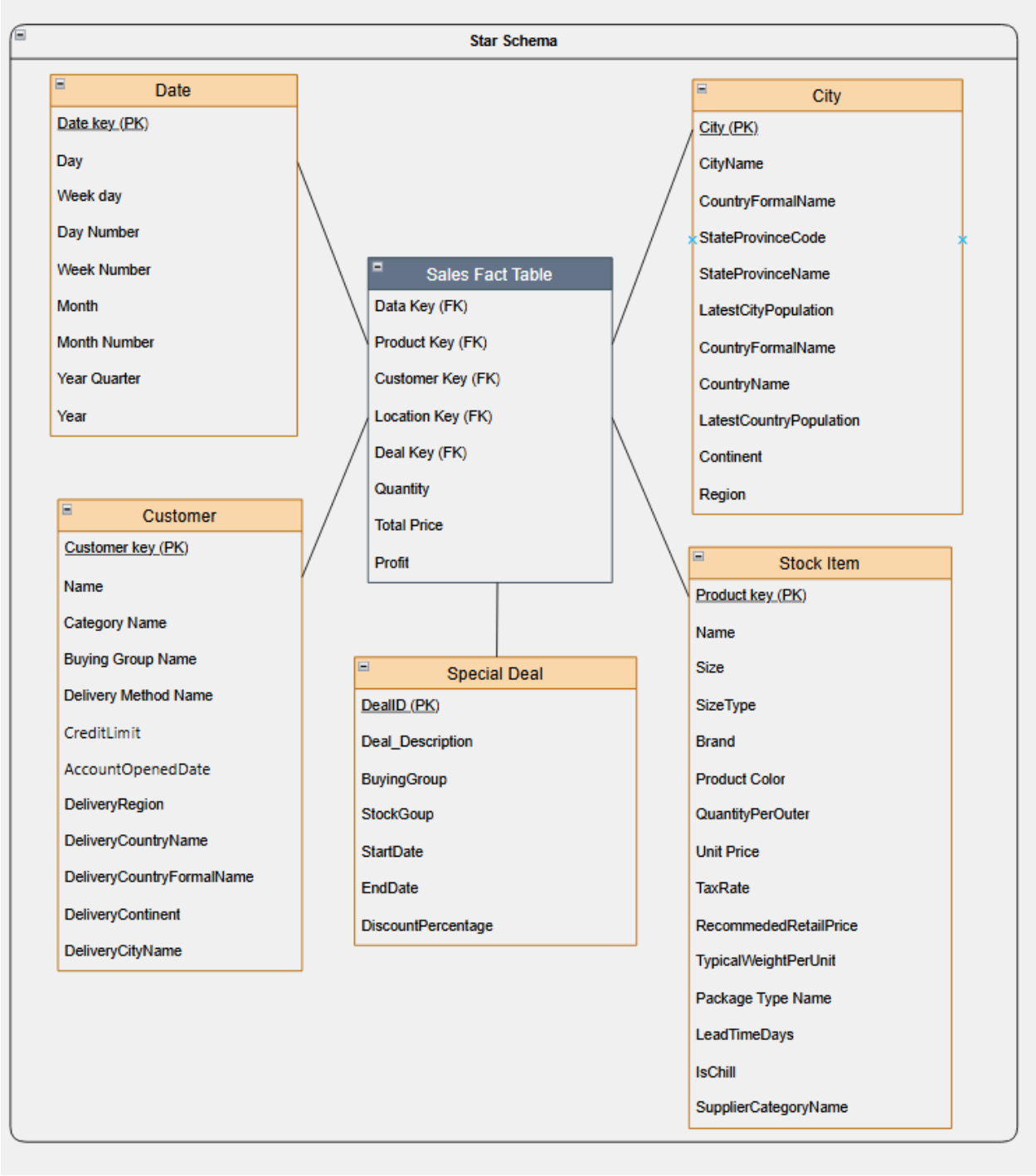
1. Track Customer Transactions: Analyze transaction details, including amounts, dates, and payment methods, to understand purchasing behaviors and identify trends.

2. Analyze Product Sales: Examine sales data by product to identify top-selling items, seasonal trends, and opportunities for product line optimization.

Source ERD:


ERD
Warehouse.StockItems
StockItemId
StockItemName
Brand
Size
UnitPrice
ColorID
TaxRate
RecommendedRetailPrice
WeightPerUnit
IsChill
LeadTimeDays
SupplierID
Warehouse.PackageTypes
PackageTypeID
PackageType Name
Applicartion.DeliveryMethods
Location Id
DeliveryMethodName
Sales.InvoceLines
InvoiceLineId
InvoiceId
stockItemID
Quantity
Unit Price
Sales.Invoices
InvoiceId
CustomerId
InvoiceDate
Sales.Customers
CustomerID
CustomerName
CustomerCategoryId
BuyingGroupId
DeliveryCityID
AccountOpenedDate
CreditLimit
Sales.CustomerCategory
CustomerCategoryID
CustomerCategoryName
Purchasing.SupplierCategories
SupplierCategoryID
SupplierCategoryName
Purchasing.Suppliers
SupplierID
SupplierCategoryID
Warehouse.Colors
ColorID
ColorName
Application.City
CityID
CityName
StateProvinceID
Location
LatestRecordedPopulation
Application.StateProvince
StateProvinceID
StateProvinceCode
StateProvinceName
CountryID
Location
LatestRecordedPopulation
Application.Country
CountryID
CountryName
CountryFormalName
Continent
Region
Sales.BuyingGroups
BuyingGroupID
BuyingGroupName
Sales.SpecialDeals
SpecialDealID
StockItemID
CustomerID
BuyingGroupID
CustomerCategoryID
StockGroupID
StartDate
EndDate
DiscountAmount
DiscountPercentage
UnitPrice

Star Schema:



Schema Description:

Grain**:**

One row per purchased stock item by a customer per day

Dimensions**:**

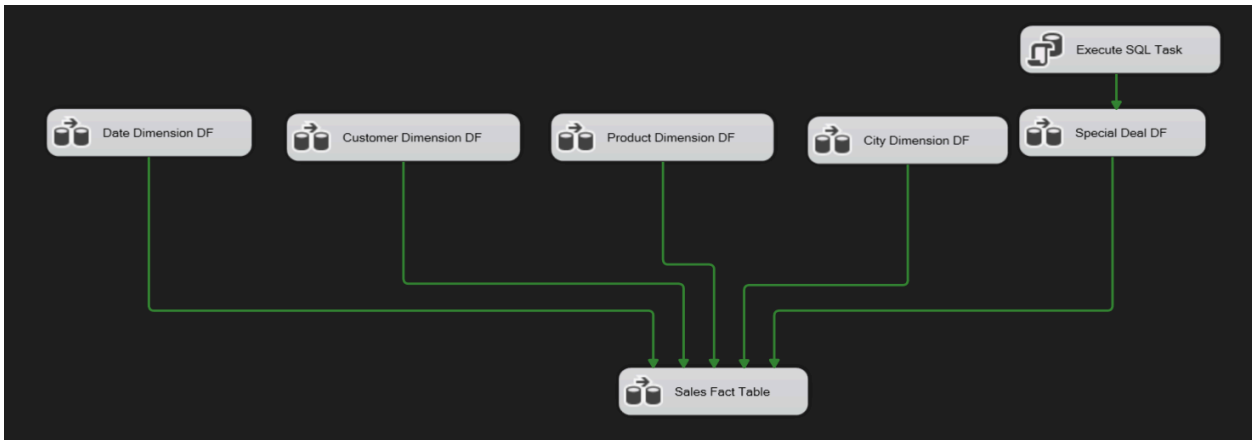
- **Customer Dimension**

- **Stock Item Dimension**
- **Location Dimension**
- **Special Deal Dimension**
- **Date Dimension**

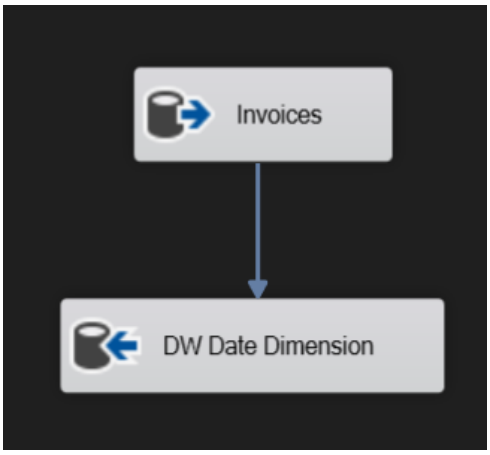
Measures**: Quantity, Profit, Total Price**

SSIS ETL Process

overview:



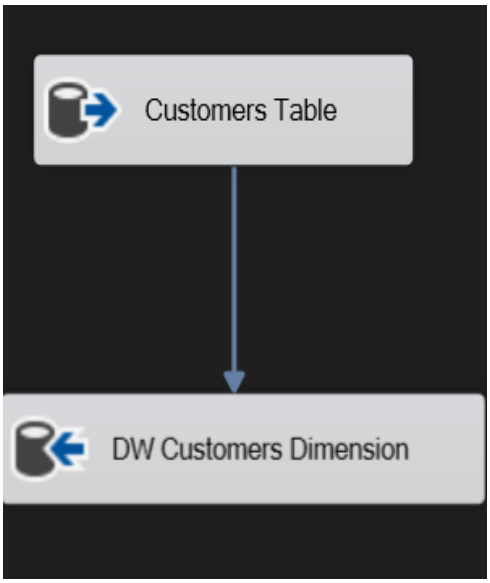
1. Date dimension data flow




```sql
-- Date dimension construction
SELECT DISTINCT
    CONVERT(INT, FORMAT(InvoiceDate, 'yyyyMMdd')) AS DateKey,
    DATENAME(WEEKDAY, InvoiceDate) AS DayName,
    DATEPART(WEEKDAY, InvoiceDate) AS WeekDay,
    DATEPART(DAY, InvoiceDate) AS DayNumber,
    DATEPART(WEEK, InvoiceDate) AS WeekNumber,
    DATENAME(MONTH, InvoiceDate) AS MonthName,
    DATEPART(MONTH, InvoiceDate) AS MonthNumber,
    DATEPART(QUARTER, InvoiceDate) AS YearQuarter,
    YEAR(InvoiceDate) AS Year
FROM Sales.Invoices
WHERE InvoiceDate IS NOT NULL;
```

The above query is used to extract date details from the date field in invoices table.

2. Customer dimension data flow




```sql
-- Customer dimension construction
SELECT
    c.CustomerID,
    c.CustomerName,
    scc.CustomerCategoryName as CategoryName,
    ISNULL(bg.BuyingGroupName, 'None') as BuyingGroupName,
    dm.DeliveryMethodName,
    ISNULL(c.CreditLimit, 0)  as CreditLimit,
    c.AccountOpenedDate,
    ci.CityName AS DeliveryCityName,
    sp.StateProvinceName AS DeliveryStateProvinceName,
    sp.StateProvinceCode AS DeliveryStateProvinceCode,
    co.CountryName AS DeliveryCountryName,
    co.FormalName AS DeliveryCountryFormalName,
    co.Continent AS DeliveryContinent,
    co.region as region
FROM Sales.Customers c
LEFT JOIN Application.Cities ci ON c.DeliveryCityID = ci.CityID
LEFT JOIN Application.StateProvinces sp ON ci.StateProvinceID = sp.StateProvinceID
LEFT JOIN Application.Countries co ON sp.CountryID = co.CountryID
LEFT JOIN Sales.CustomerCategories scc ON  scc.CustomerCategoryID = c.CustomerCategoryID
LEFT JOIN Sales.BuyingGroups bg ON bg.BuyingGroupID = c.BuyingGroupID
LEFT JOIN Application.DeliveryMethods dm on dm.DeliveryMethodID = c.DeliveryMethodID
WHERE c.CustomerID IS NOT NULL
```

The above query:

- extracts customer details by denormalizing the related tables through simple joins
- sets ***buying group*** and ***Credit limit*** fields to neutral values to avoid nulls.

3. Product dimension data flow

```sql
-- Product dimension construction
SELECT
    si.StockItemID,
    si.StockItemName AS Name,
    ISNULL(si.Size, 'Unknown') AS Size,
    ISNULL(si.Brand, 'Unbranded') AS Brand,
    ISNULL(c.ColorName, 'No Color') AS ProductColor,
    si.QuantityPerOuter,
    si.UnitPrice,
    si.TaxRate,
    si.RecommendedRetailPrice,
    si.TypicalWeightPerUnit,
    ISNULL(pt.PackageTypeName, 'Unknown Package') AS PackageTypeName,
    si.LeadTimeDays,
    si.IsChillerStock AS IsChill,

    CASE
        WHEN RTRIM(sc.SupplierCategoryName) LIKE '% supplier'
            THEN LEFT(RTRIM(sc.SupplierCategoryName), LEN(RTRIM(sc.SupplierCategoryName)) - 9)
        WHEN RTRIM(sc.SupplierCategoryName) LIKE '% suppliers'
            THEN LEFT(RTRIM(sc.SupplierCategoryName), LEN(RTRIM(sc.SupplierCategoryName)) - 10)
        ELSE sc.SupplierCategoryName
    END AS SupplierCategoryName,

    CASE
        WHEN si.Size IS NULL THEN 'Unknown'
        WHEN (si.Size LIKE '%ml' OR si.Size LIKE '%L') AND si.Size NOT IN ('L', 'XL') THEN 'Volume'
        WHEN si.Size IN ('S', 'M', 'L') OR si.Size LIKE '%XL' OR si.Size LIKE '%XS' THEN 'Clothing'
        WHEN si.Size LIKE '%kg' OR si.Size LIKE '%g' THEN 'Weight'
        WHEN si.Size LIKE '%x%' THEN 'Dimensions'
        WHEN si.Size LIKE '%cm' OR si.Size LIKE '%m' THEN 'Length'
        ELSE 'Unknown'
    END AS SizeType

FROM Warehouse.StockItems si
LEFT JOIN Warehouse.Colors c ON si.ColorID = c.ColorID
LEFT JOIN Warehouse.PackageTypes pt ON si.UnitPackageID = pt.PackageTypeID
LEFT JOIN Purchasing.Suppliers sp ON si.supplierID = sp.SupplierID
LEFT JOIN Purchasing.SupplierCategories sc ON sc.SupplierCategoryID= sp.SupplierCategoryID;
```

The above query:

- Constructs product dimension from various normalized tables through simple joins.
- Sets ***PackageType***, ***Brand***, ***Color***, ***Size*** to neutral values to avoid nulls.
- Trims “supplier(s)” from the end of supplier category name
- Adds a ***SizeType*** column to solve inconsistencies in size units (volume in Litre/ml, Clothing sizes (S, M, L, XL, …), Weight and others.

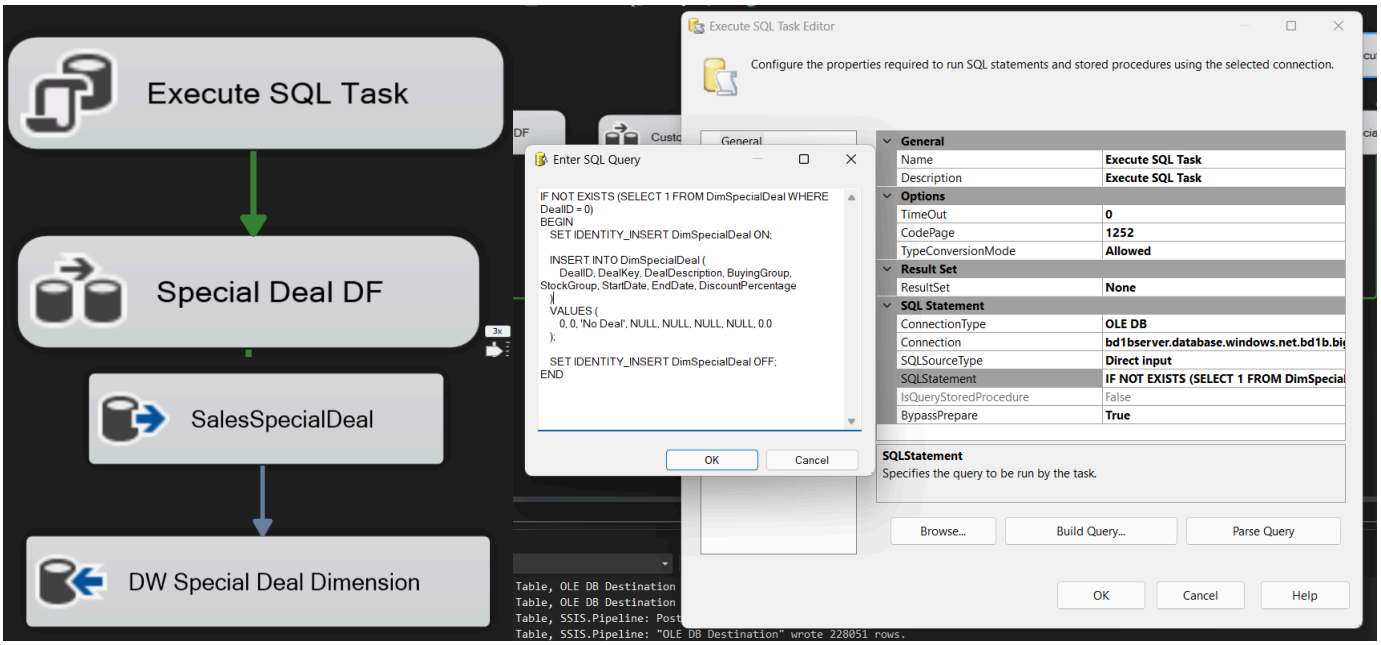
4. City Dimension data flow

```sql
-- City dimension construction
SELECT
    c.CityID AS CityKey,
    c.CityName,
    ISNULL(c.LatestRecordedPopulation, 0) AS LatestCityPopulation,
    sp.StateProvinceCode,
    sp.StateProvinceName,
    co.CountryName,
    co.FormalName AS CountryFormalName,
    ISNULL(co.LatestRecordedPopulation, 0) AS LatestCountryPopulation,
    co.Continent,
    co.Region
FROM Application.Cities c
JOIN Application.StateProvinces sp ON c.StateProvinceID = sp.StateProvinceID
JOIN Application.Countries co ON sp.CountryID = co.CountryID
```

The above query:

- Gets City dimension details from Application.cities, StateProvinces, and countries tables
- Sets null fields to neutral values.

5. Special deal dimension data flow

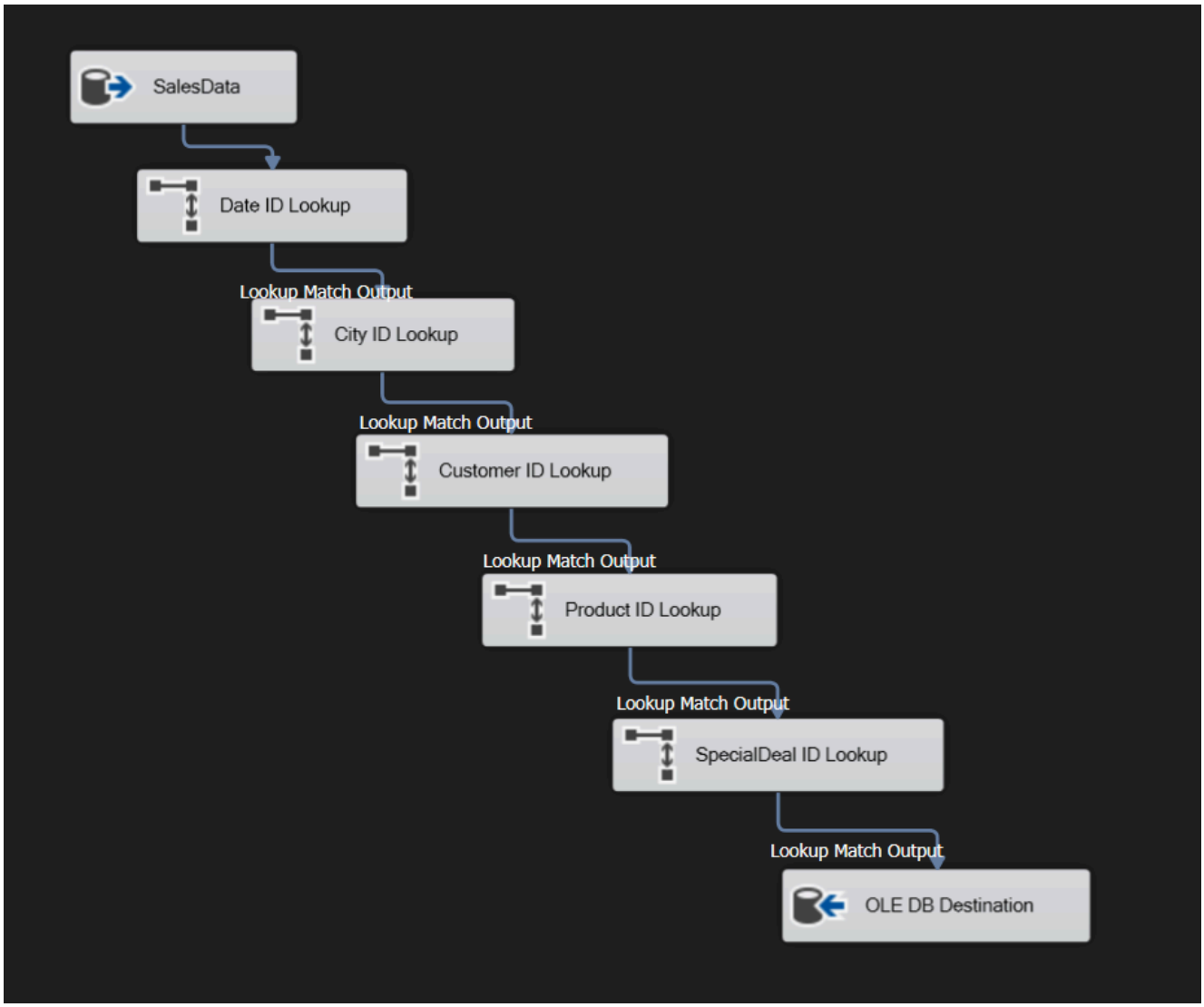



```sql
SELECT
 sd.SpecialDealId,
     sd.DealDescription,
     ISNULL(bg.BuyingGroupName, 'All Customers') AS BuyingGroup,
     ISNULL(sg.StockGroupName, 'All Products') AS StockGroup,
     sd.StartDate,
     sd.EndDate,
     ISNULL(sd.DiscountPercentage, 0.00) AS DiscountPercentage
 FROM Sales.SpecialDeals sd
 LEFT JOIN Sales.BuyingGroups bg ON sd.BuyingGroupID = bg.BuyingGroupID
 LEFT JOIN Warehouse.StockGroups sg ON sd.StockGroupID = sg.StockGroupID
```

Special Deal Dimension construction happens on two steps:

1. Adding a row with id = 0 representing no deal to include products which wasn't a part of a special deal without having nulls in the fact table composite primary key.
2. A simple data flow query

6. Sales fact table data flow



Data Flow Query:

```sql
WITH AggregatedSource AS (
    SELECT
        CAST(CONVERT(char(8), i.InvoiceDate, 112) AS INT) AS DateKey,
        c.CustomerID AS CustomerKey,
        il.StockItemID AS ProductKey,
        c.DeliveryCityID AS LocationKey,
        ISNULL(sd.SpecialDealID, 0) AS DealKey,
        il.Quantity,
        il.Quantity * il.UnitPrice AS TotalPrice,
        il.LineProfit AS Profit
    FROM Sales.InvoiceLines il
    JOIN Sales.Invoices i ON il.InvoiceID = i.InvoiceID
    JOIN Sales.Customers c ON i.CustomerID = c.CustomerID
    LEFT JOIN Sales.SpecialDeals sd ON sd.StockItemID = il.StockItemID
)
SELECT
    DateKey,
    CustomerKey,
    ProductKey,
    LocationKey,
    DealKey,
    SUM(Quantity) AS Quantity,
    SUM(TotalPrice) AS TotalPrice,
    SUM(Profit) AS Profit
FROM AggregatedSource
GROUP BY DateKey, CustomerKey, ProductKey, LocationKey, DealKey;
```

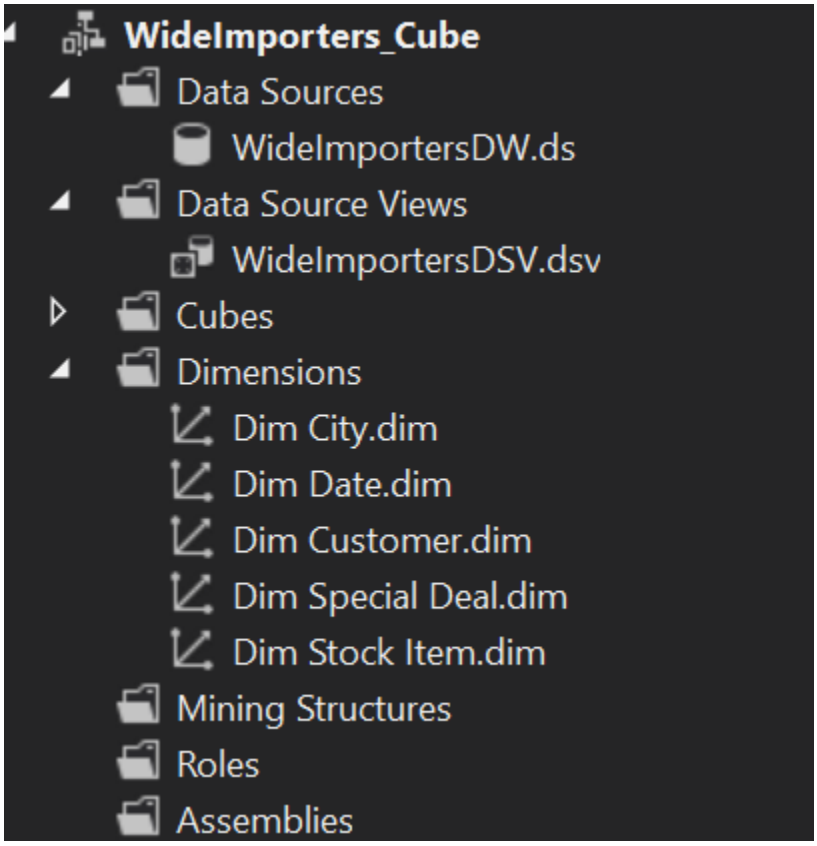
The above query:

- Gets Dimensions natural keys through joins.
- Gets measures from ***invoice lines*** table and aggregate over dimension keys.

SSAS Process

1. Adding Dimensions

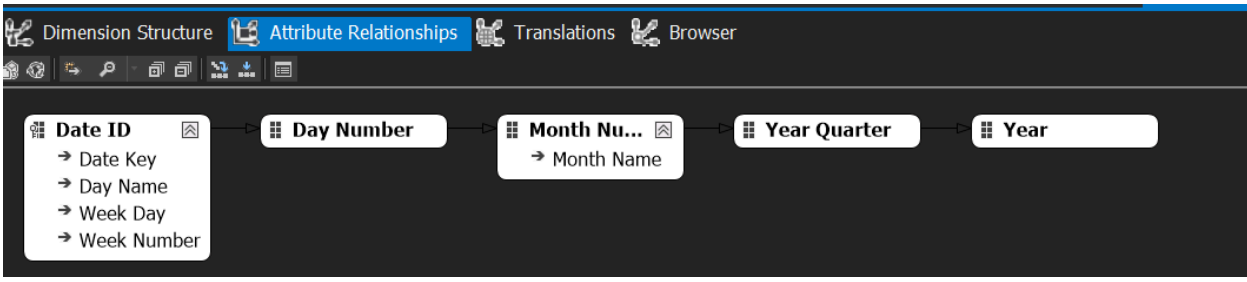
After the ETL stage, all the required dimensions were added to the Data Source View in SSAS.



2. Creating needed hierarchies

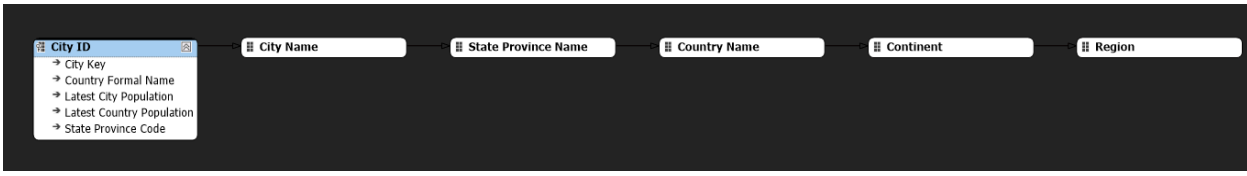
a. Time Dimension

This dimension includes day numbers, months, quarters, and years. To improve query performance and navigation, a time hierarchy was created.



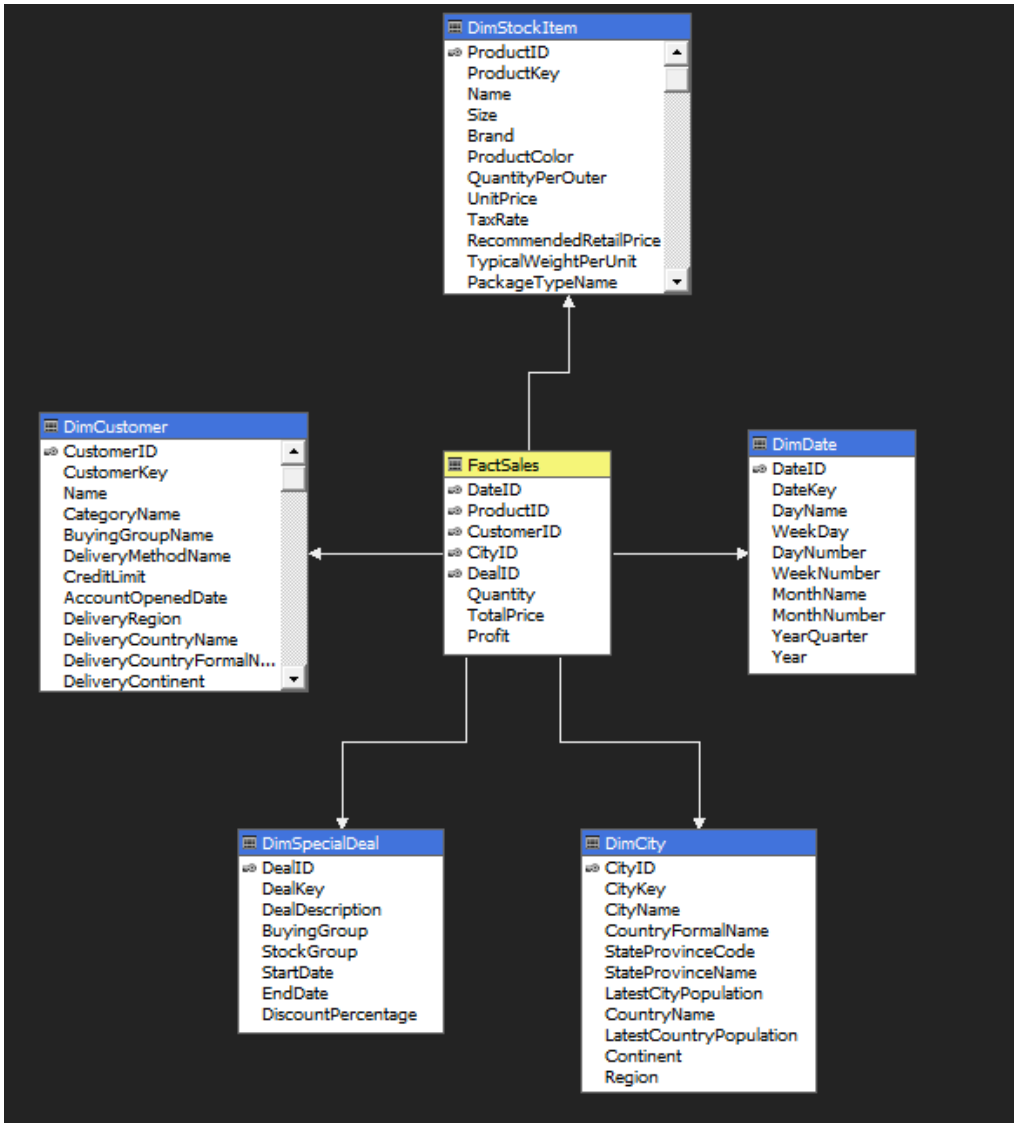
b. City Dimension

The City dimension includes city, state, country, continent, and region. A hierarchy was also created for this dimension to make it easier to drill down and analyze by location.



3. Creating the cube

The cube was created using the fact table, and the dimensions were added. Measures were defined as needed (Quantity, Total Price, Profit).



Analysis Using the Cube with MDX queries

Here are some samples of queries using the cube and their results

```
-- find out the quantity and profit sold ove the years
SELECT
[Dim Date].[Time].[Year] on rows,
{[Measures].[Quantity] , [Measures].[Profit]} on columns
FROM [WideImportersCube]
```

39 %

Messages | Results

	Quantity	Profit
2013	2401657	22768352.25
2014	2567401	24828462.45
2015	2740266	26957600.65
2016	1241304	11174765.55

```
-- what are the top 3 selling categories sold over the years
SELECT
[Dim Date].[Time].[Year] on rows,
topcount([Dim Stock Item].[Supplier Category Name].members,4,[Measures].[Quantity]) on columns
FROM [WideImportersCube]
```

89 %

Messages | Results

	All	Packaging	Clothing	Novelty Goods
2013	2401657	1557071	710689	106207
2014	2567401	1649422	771357	116488
2015	2740266	1765233	817394	124554
2016	1241304	724621	324898	178529

```
-- Top 5 most profitable stock items
SELECT
  TopCount([Dim Stock Item].[Name].Members, 5, [Measures].[Profit]) on rows,
  [Measures].[Profit] on columns
FROM [WideImportersCube]
```

108 %

Messages | Results

	Profit
All	85729180.9000001
20 mm Double sided bubble wrap 50m	5293680
Air cushion machine (Blue)	4439391
32 mm Anti static bubble wrap (Blue) 50m	3526400
10 mm Anti static bubble wrap (Blue) 50m	3452220

```
-- most profitable month of a specific year
SELECT
  TopCount(
    [Dim Date].[Time].[Month Number].Members,
    1,
    [Measures].[Profit]
  ) ON ROWS,
  {[Measures].[Profit]} ON COLUMNS
FROM [WideImportersCube]
where ([Dim Date].[Year].[2013])
```

108 %

Messages | Results

	Profit
5	2231604.45

```
-- most profitable quarter of a specific year
SELECT
  TopCount(
    [Dim Date].[Time].[Year Quarter].Members,
    1,
    [Measures].[Profit]
  ) ON ROWS,
  {[Measures].[Profit]} ON COLUMNS
FROM [WideImportersCube]
where ([Dim Date].[Year].[2014])
```

108 %

Messages | Results

	Profit
2	6435079.5

```
-- Total sales of each state
SELECT
  [Dim City].[Hierarchy].[State Province Name].Members on rows,
  [Measures].[Profit] on columns
FROM [WideImportersCube]
```

108 %

Messages | Results

	Profit
Hawaii	161952
Idaho	911020.8
Illinois	2251077.05
Indiana	1645318.8
Iowa	1563214.95
Kansas	1709816
Kentucky	1269557.7
Louisiana	1710731.7
Maine	903079.95
Maryland	1074535.3
Massachusetts[E]	1366291.65
Michigan	1760800.1
Minnesota	2498576.1
Mississippi	639864.15

```
-- Total Sales of each State over the years
SELECT
  non empty [Dim City].[City Name].Members on rows,
  non empty [Dim Date].[Time].[Year].Members on columns
FROM [WideImportersCube]
WHERE ([Measures].[Profit])
```

108 %

Messages | Results

	2013	2014	2015	2016
All	22768352.25	24828462.45	26957600.65	11174765.55
Abbottsburg	49414.85	39632.7	62523	22376.4
Absecon	39040.7	44474.4	32445.8	13397.45
Accomac	50445.5	57324.5	30164	19834.4
Aceitunas	37205.85	36577.2	33696.55	11803.4
Airport Drive	53217.7	47086.7	41538.3	20657.3
Akhiok	69254.2	79376.9	77201.35	33721.85
Alcester	45824.75	46559.1	27106.6	7549.8
Alden Bridge	45866.1	51558.75	44754.6	9958.4
Alstead	30998.65	25450.3	29259	20439
Amado	48017.25	34678.55	41389.1	12632.9
Amanda Park	39419.9	24043.6	38364.85	15615.5
Andrix	37103.1	36304.25	39254.1	18048.55

The Power BI Dashboard