AdventureWorks2014 – BI Flow

Overview
This implementation follows a classical **Data Warehouse lifecycle**, starting from system setup and source system preparation, moving through ETL processes, semantic modeling with SSAS Tabular, and ending with analytical reporting in Power BI. The AdventureWorks2014 database is used as the operational source system, and the final output is an interactive BI dashboard answering key business questions related to sales performance.

---

0. Software and Server Setup
The initial step focused on preparing a development environment

- **SQL Server 2025** was installed with the **Analysis Services (SSAS)** feature enabled, allowing both relational database management and tabular analytical modeling.
- **Visual Studio** was installed as the primary development IDE.
- **SSIS and SSAS extensions** were added to Visual Studio to support:
    - SSIS package development for ETL pipelines
    - SSAS Tabular project creation and deployment

---

1. Creating Database Objects
a. Source Database

- The AdventureWorks2014 database was restored in SQL Server Management Studio (SSMS) and used as the transactional source system.

b. Staging Database

- SQL script 1_AWN_STG_Demo was executed to create schemas, tables, and views in the staging database.

c. Data Warehouse Schema

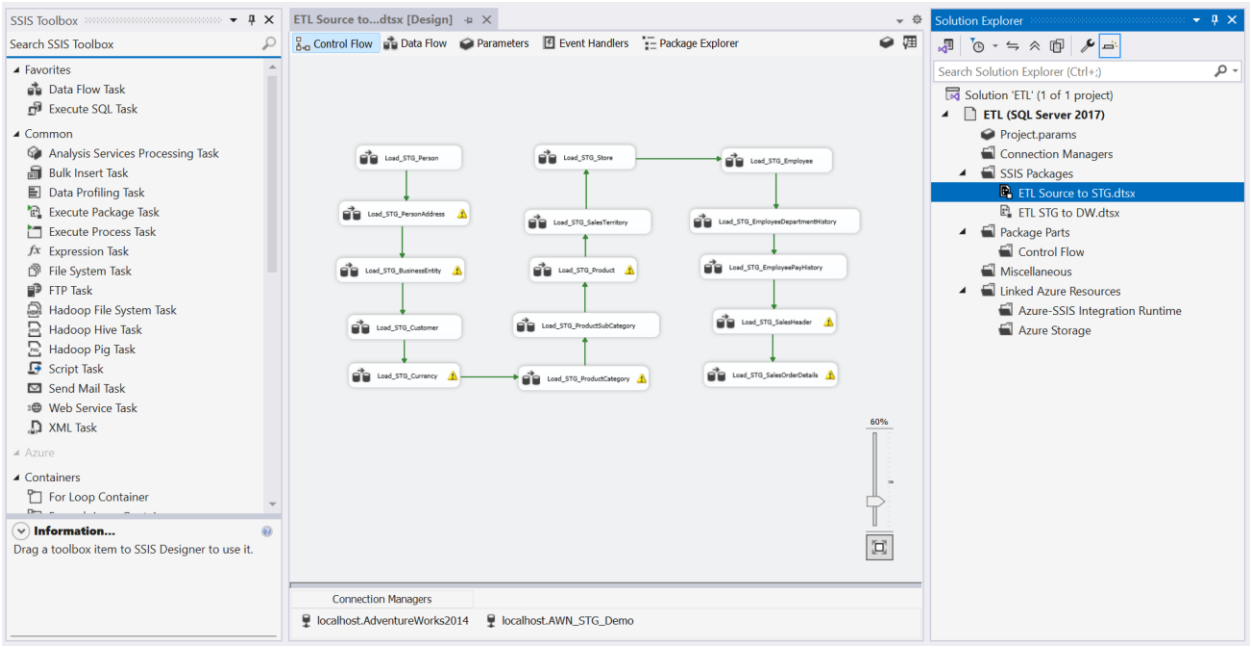
- SQL script 2_AWN_DW_Demo was executed up to line 326 to create the core data warehouse schema.

---

2. ETL: Source to Staging
This stage focused on extracting data from the operational system and loading it into the staging database.
a. SSIS Package Creation

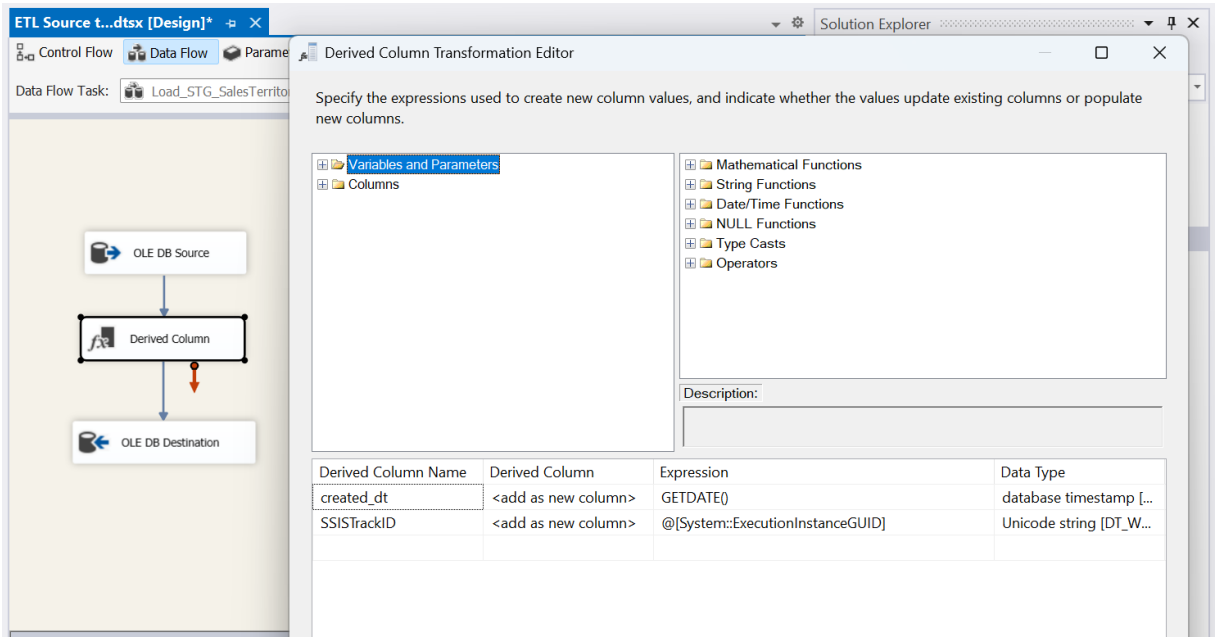
- A new SSIS package was created in Visual Studio.
- Data source: AdventureWorks2014
- Destination: AWN_STG_Demo (staging database)

b. Data Population
The package populated the following staging tables:



- Master data:
  Business_Entity, Currency, Customer, Person, Product, ProductCategory, ProductSubCategory, Store
- Transactional data:
  SalesHeader, SalesOrderDetails
- HR data:
  Employee, EmployeeDepartmentHistory, EmployeePayHistory
- Supporting dimensions:
  PersonAddress, SalesTerritory

Some tables in AWN_STG_Demo require CreatedDate_dt and SSISTrackID, so we used Derived Column to create the according columns.



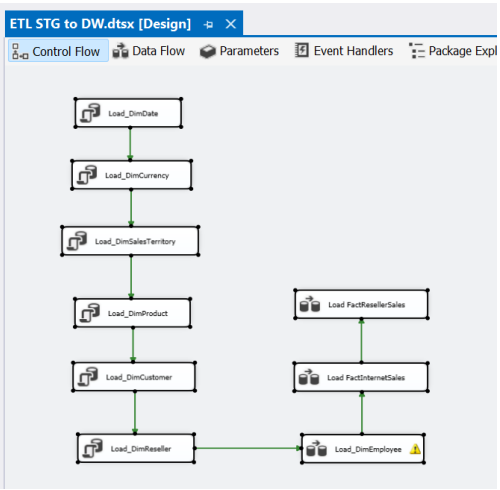
c. Validation

- The package was executed successfully.
- Data integrity and row counts were verified in SSMS to ensure accurate extraction and loading.

---

3. ETL: Staging to Data Warehouse

This stage transformed staged data into **analytics-ready dimensional structures**.



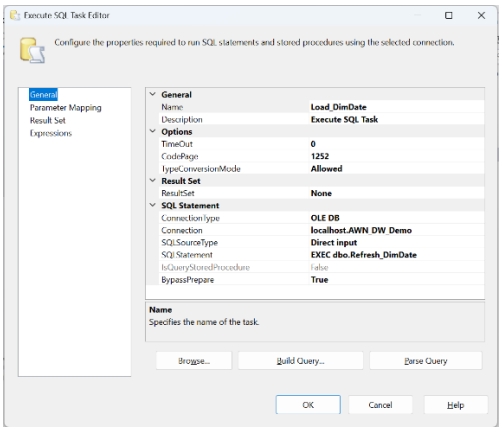
a. Dimension and Procedure Setup

- Stored procedures were created by executing:
  - 2_AWN_DW_Demo (from line 326 to end)
  - 3_AWN_HR_Demo (from line 102 to end)

These procedures encapsulate transformation logic and ensure repeatable, auditable loads.

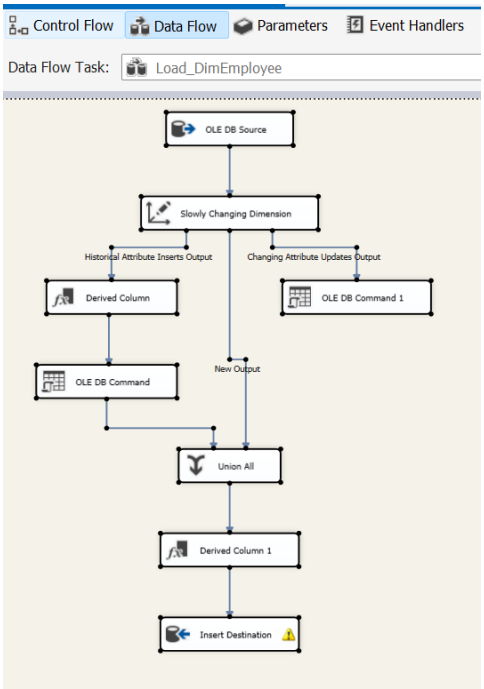
b. Loading Dimensions and FactEmployeePay

- SSIS Data Flow Tasks were created to execute the following procedures:
  - EXEC dbo.Refresh_DimDate
  - EXEC dbo.Refresh_DimCurrency
  - EXEC dbo.Refresh_DimCustomer
  - EXEC dbo.Refresh_DimProduct
  - EXEC dbo.Refresh_DimSalesTerritory
  - EXEC dbo.Refresh_Reseller
  - EXEC dbo.Refresh_FactEmployeePay



c. Slowly Changing Dimension (DimEmployee)

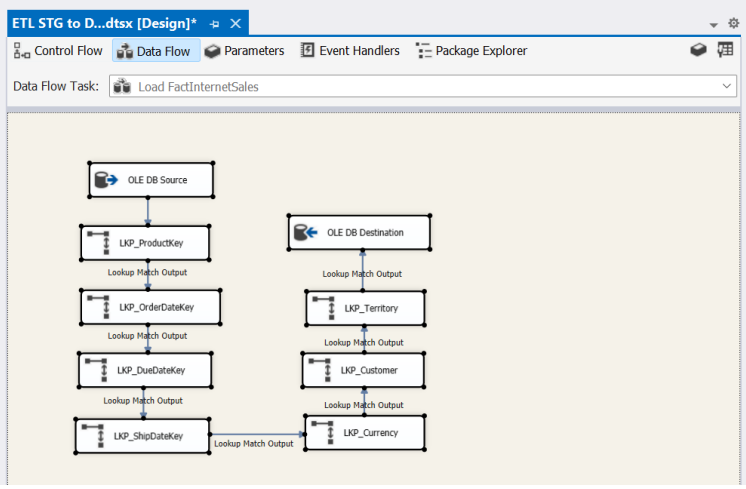
- A dedicated SSIS package was built using the **Slowly Changing Dimension (SCD)** component.
- This allows tracking historical changes in employee attributes (e.g., department changes over time).



d. Fact Table Population

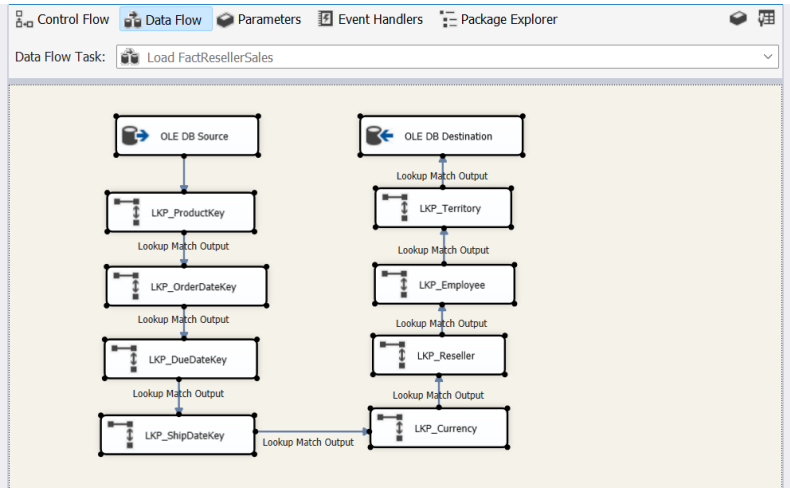
FactInternetSales

- Source: Stg_vw_Erp_Fact_InternetSales
- Lookup transformations were used to resolve foreign keys to dimensions.



FactResellerSales

- Source: Stg_vw_Erp_Fact_ResellerSales
- Lookup transformations mapped dimension keys consistently.



e. Execution and Validation

- All ETL packages were executed.
- Fact and dimension tables were validated in SSMS for completeness and correctness.

---

4. Creating SSAS Tabular Model

This stage introduced the **semantic layer**, optimized for analytical queries.

a. Project Setup

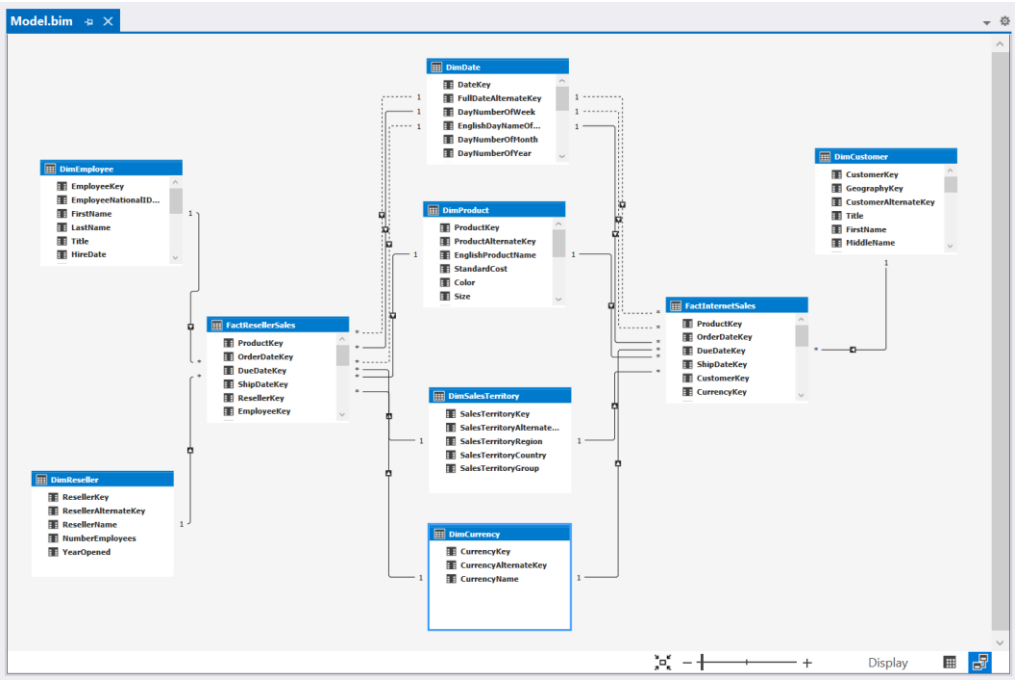
- A new **SSAS Tabular Project** was created in Visual Studio.
- The data warehouse was configured as the data source.

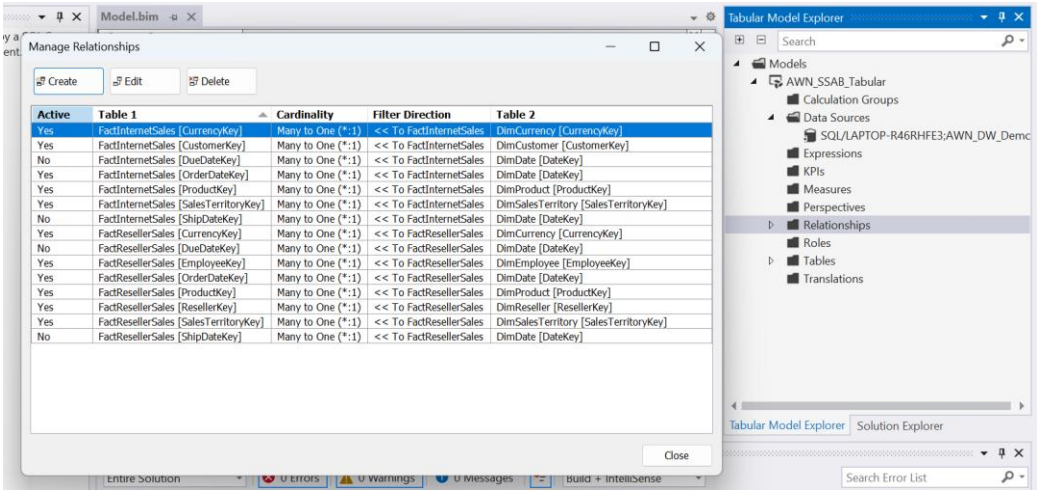
b. Data Import

- All relevant dimension and fact tables were imported.

c. Model Validation

- Relationships were reviewed and manually created where missing.
- Cardinality and filter directions were verified.





d. Deployment

- The tabular model was successfully built and deployed to SQL Server Analysis Services.

---

5. Creating Power BI Reports

The final stage focused on **business analysis and visualization**.

a. Connection

- Power BI connected to SSAS using a **live connection**, ensuring real-time interaction with the semantic model.

b. Business Questions

Key analytical questions included:

- Comparison of sales performance across channels
- Sales trends over time
- Performance by product category and territory

c. Dashboard Design

- A clean dashboard layout was designed with time-series charts, categorical comparisons, and KPIs.
- 3 pages:
  - Overview
  - Reseller Sales
  - Internet Sales

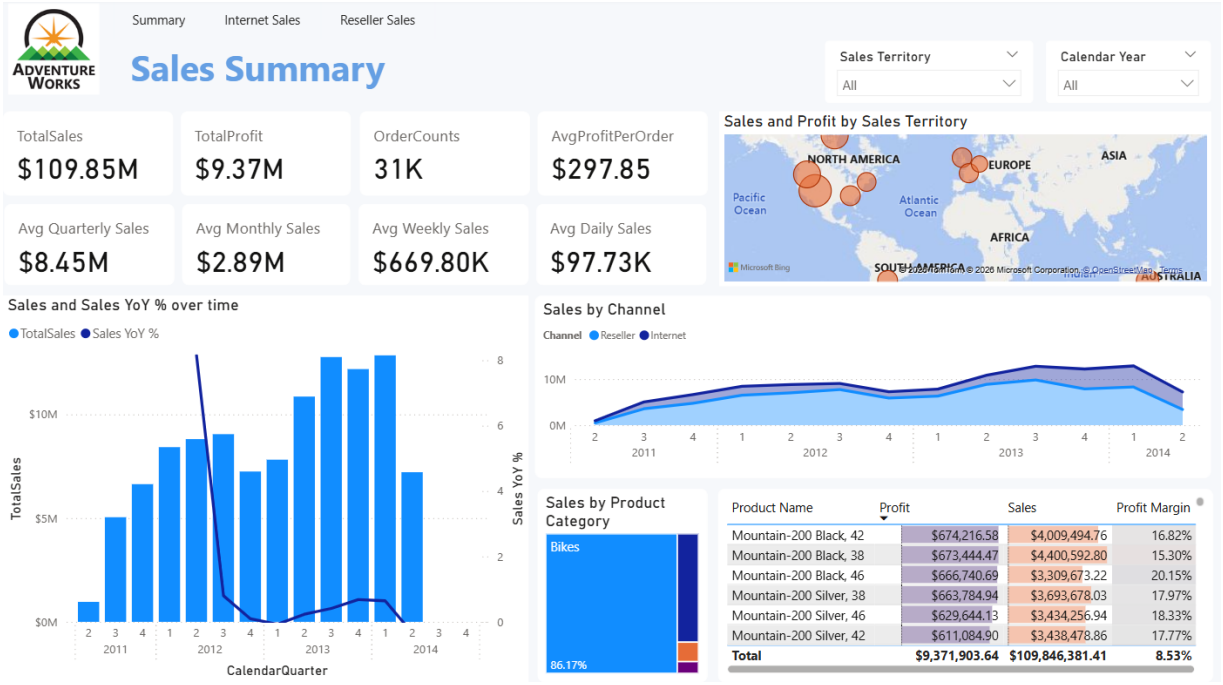
d. DAX Measures

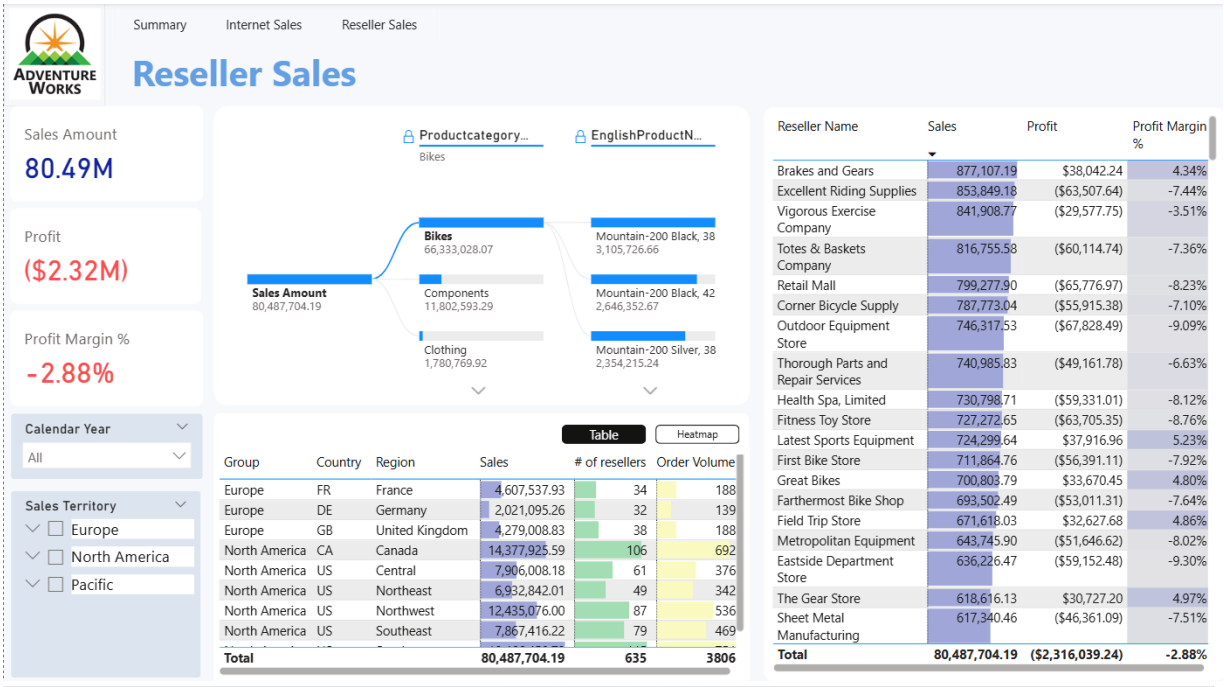
Custom measures were created, including:

- TotalSales
- TotalCost
- TotalProfit
- Sales LY
- Sales YoY%
- ProfitMargin
- OrderCounts
- AvgProfitPerOrder
- AverageQuarterlySales
- AverageMonthlySales
- AverageWeeklySales
- AverageDailySales

e. Final Output

- The dashboard provides interactive, drillable insights suitable for management and analytical decision-making.





Conclusion

This implementation demonstrates a **complete BI pipeline**, from raw operational data to executive-level analytics. Each stage reinforces core data warehousing principles: separation of concerns, controlled transformations, semantic abstraction, and business-oriented reporting. The result is a scalable and maintainable analytical solution aligned with industry best practices.