# ****Report on Sales Analysis- Group 21****

## ****Introduction****

This report details the implementation of a comprehensive Business Intelligence (BI) flow using the AdventureWorks2014 database. The project begins with setting up a Staging database and a Data Warehouse (DW) modeled using a Galaxy Schema. The process involves executing ETL (Extract, Transform, Load) operations to transfer data from the source AdventureWorks2014 database to the Staging database, and then from Staging to the DW. Following this, a tabular model is developed using SQL Server Analysis Services (SSAS) to enable multidimensional data analysis. Finally, Power BI is utilized to create interactive reports, offering insights into sales trends across various dimensions such as years, months, product categories, countries, and sales modalities. This structured approach ensures the delivery of a robust and insightful data analysis framework.

## ****Setting****

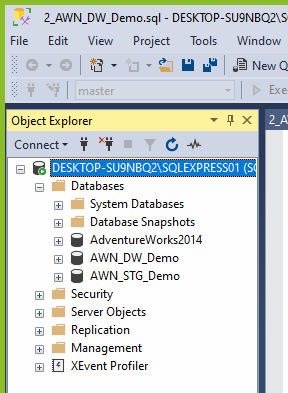
The objective is to implement a full BI flow for AdventureWorks2014, creating a data warehouse with two data marts (Sales and HR) and analyzing sales data through Power BI. The data model includes two fact tables: Internet Sales and Reseller Sales, with shared dimension tables.

## ****Implementation****

II.1. Creating Database Objects

*Figure 1: The created Database Objects*

The first step in the implementation involved creating two databases: a Staging database and a Data Warehouse (DW) database modeled using a Star/Galaxy Schema. The process began with restoring the AdventureWorks2014 database in SQL Server Management Studio (SSMS) to serve as the source for the Staging database. Next, the 1\_AWN\_STG\_Demo SQL script was executed to set up the necessary schema, tables, and views for the Staging database by transforming the source data. Subsequently, the 2\_AWN\_DW\_Demo script was run up to line 326 to establish the schema and tables for the DW.



II.2. ETL Source to Staging

In step 2, two SSIS packages were created: the HR package and the ERP Full package.

* The HR package was designed to populate the tables:
  + Employee,
  + EmployeeDepartmentHistory, and
  + EmployeePayHistory tables.
* The ERP Full package was used to populate the tables of Sales Data Mart:

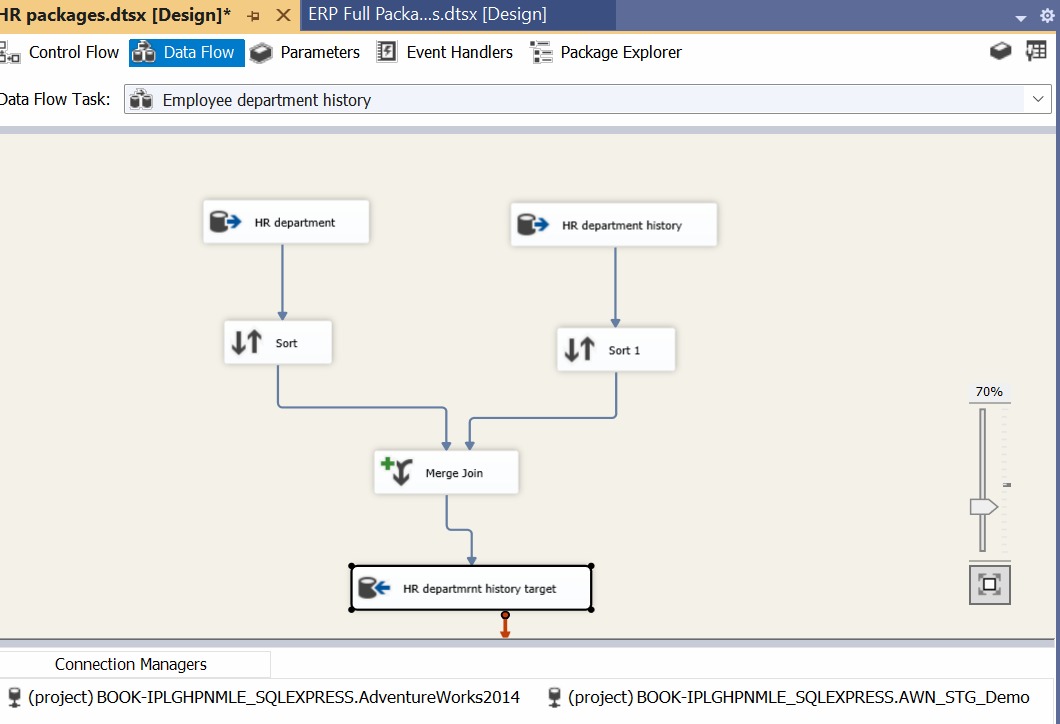
|  |  |  |  |
| --- | --- | --- | --- |
| Business\_Entity | Currency | Customer | Person |
| PersonAddress | Product | ProductCategory | ProductSubCategory |
| SalesHeader | SalesOrderDetails | SalesTerritory | Store. |

Both packages used connection managers to link the **AdventureWorks2014** database as the source and the **AWN\_STG\_Demo** database as the target. Data flow tasks were created for each table in both packages.

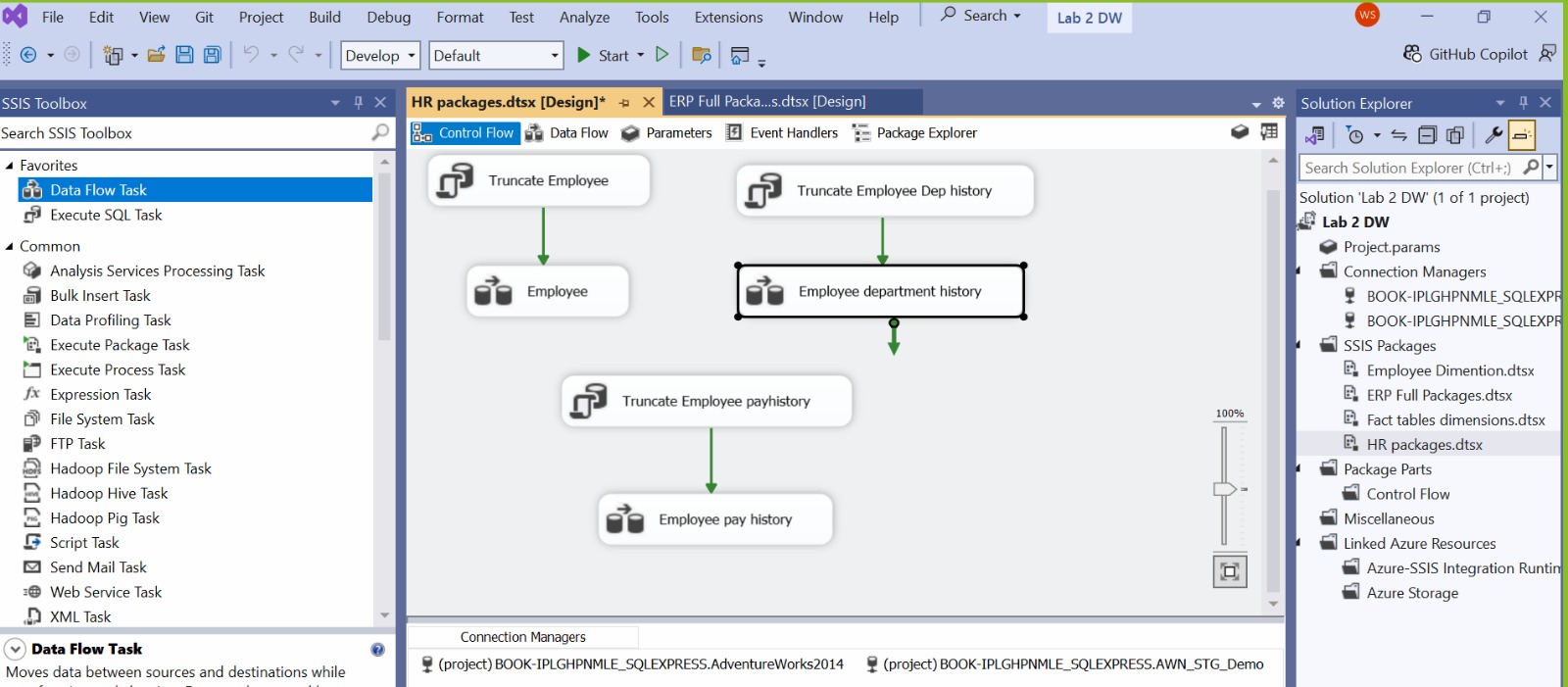
For the [erp].[PersonAddress] table, a merge join was used to combine data from [Person].[Address] and [Person].[BusinessEntityAddress]. Similarly, for [hr].[EmployeeDepartment], a merge join connected [HumanResources].[Department] and [HumanResources].[EmployeeDepartmentHistory].

To prevent duplicates, an Execute SQL Task was included to truncate tables before each load. After executing all tasks, the data was verified in SSMS to confirm successful population.

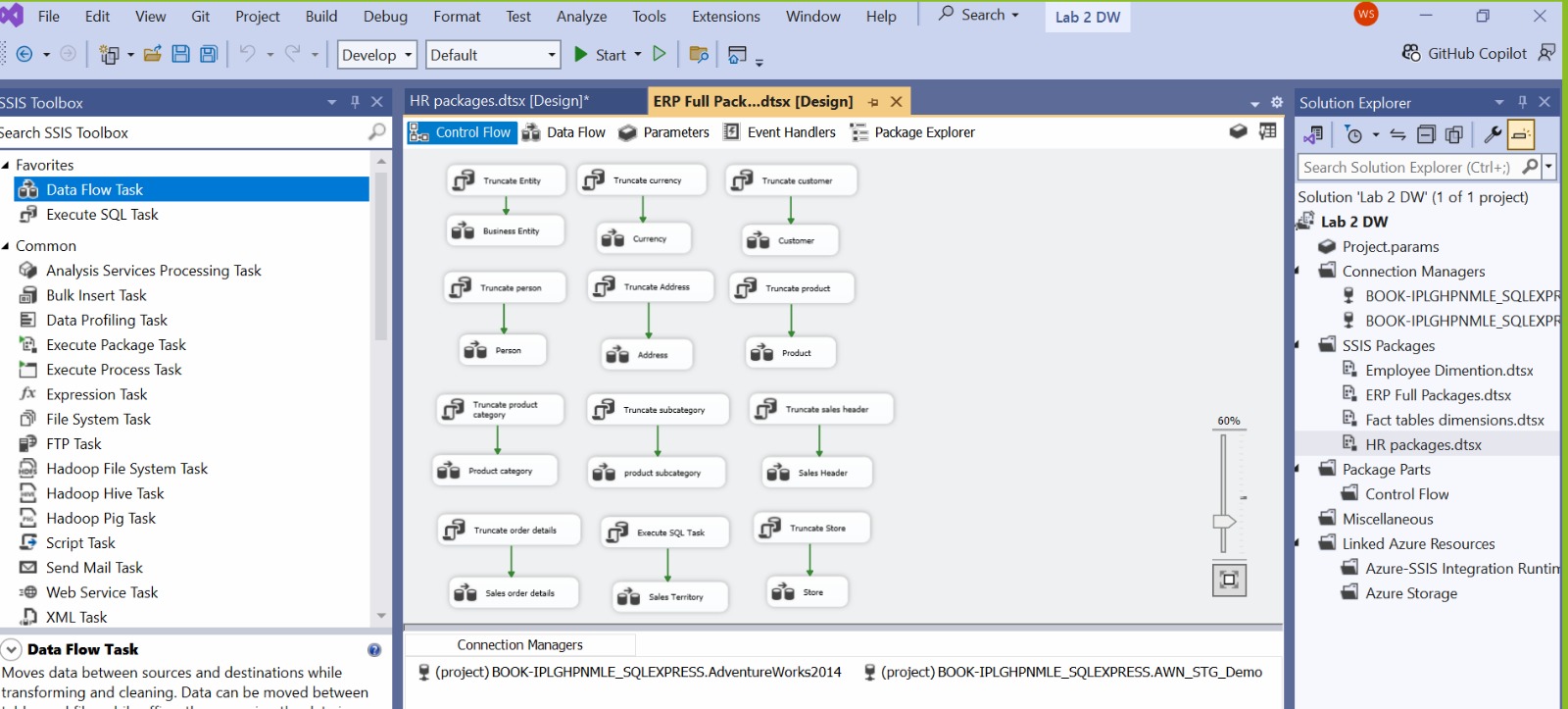
*Figure 2: HR package*



*Figure 3: EmployeeDepartmentHistroy table populated using Merge Join*



*Figure 4: ERP Full Package*



*Figure 5: Execute SQL Task used to prevent duplicates*

A screenshot of a computer

Description automatically generated

II.3. ETL Staging to DW

In this stage, data was loaded from the Staging database to the Data Warehouse (DW) following a chronological order, starting with the dimension tables and then the fact tables. The dimension tables were populated by executing procedures from the 2\_AWN\_DW\_Demo script. The DimEmployee table required a separate SSIS package with a Slowly Changing Dimension (SCD) to maintain historical data, using NationalIDNumber as the business key and tracking changes in MaritalStatus, Title, and Department. We used the Data access mode as SQL command to ensure that the Data types of StartDate and EndDate are DATETIME and we also cross checked it in the Advanced Editor.

*Figure 6 & 7: SQL command and Advanced Editor*

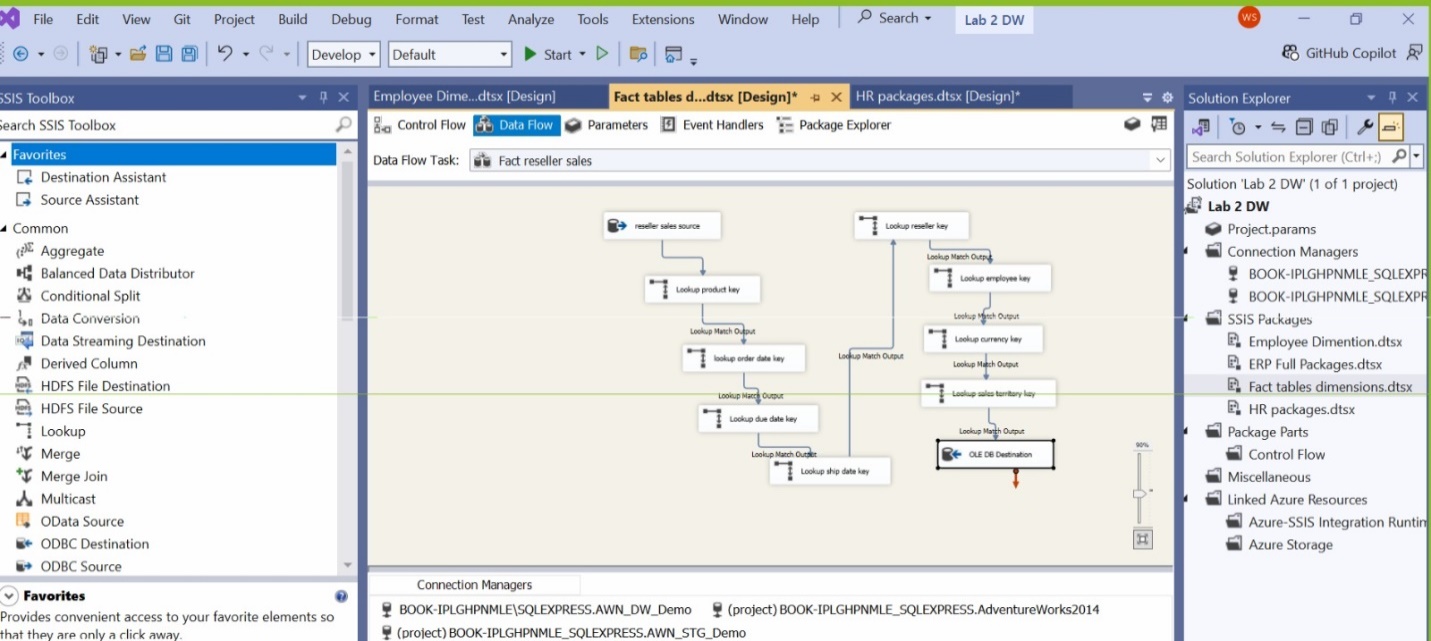
A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

For the fact tables, two Data Flow Tasks were created: one for FactInternetSales and another for FactResellerSales. These tasks involved using OLE DB Sources to read views (Stg\_vw\_Erp\_Fact\_InternetSales and Stg\_vw\_Erp\_Fact\_ResellerSales), and Lookup transformations to join the fact tables with dimension tables via foreign keys. Once configured, the SSIS packages were executed, and the data population was verified in SSMS to ensure successful loading into the DW.

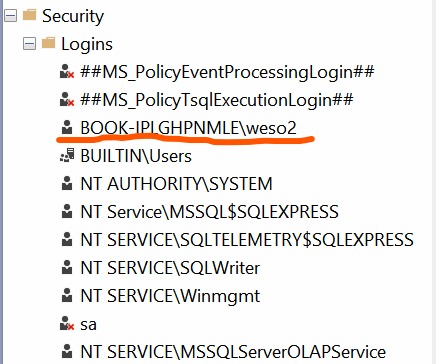
*Figure 8: Lookup Transformations used for Fact Table Reseller Sales*



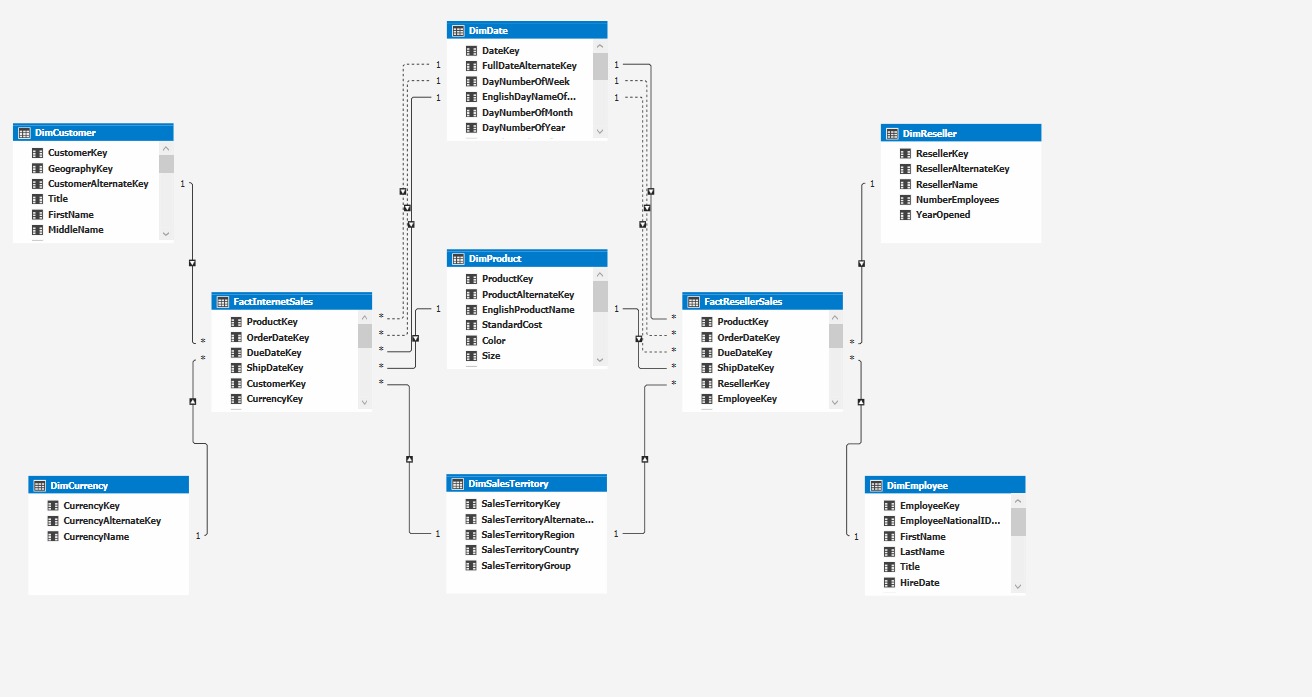
II.4. Creating SSAS tabular

In step 4, a tabular cube was implemented using SQL Server Analysis Services (SSAS). The process began by ensuring that the Analysis Services extension was installed in Visual Studio. A new Analysis Services Tabular Project was then created, and the appropriate Workspace Analysis Server was selected for deployment. Before importing data, the SQL Server Analysis account was added as a new login in SSMS, with permissions enabled to read and write to the Data Warehouse (DW). Data from the DW was then imported into the Tabular Model Explorer, including the fact tables and related dimension tables from the Sales Data Mart, ensuring the Galaxy Schema was correctly mapped. After confirming all necessary tables and views were selected and imported, the tabular model was built and deployed. The deployment results were verified in SSMS by connecting to the Analysis Services Server.

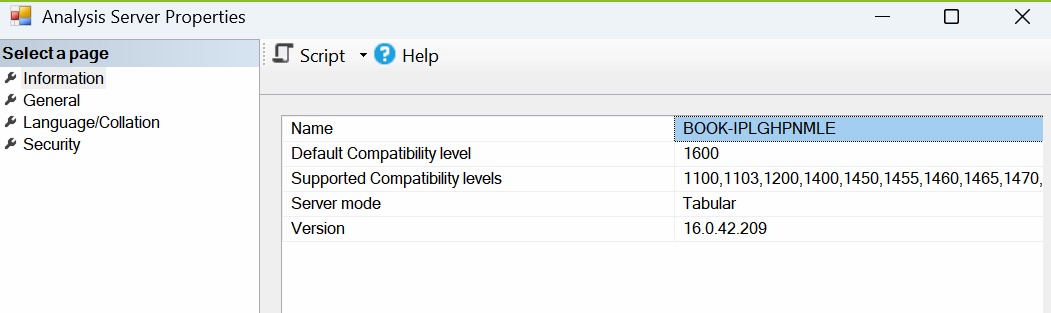
*Figure 9: SQL Server Analysis account added in SSMS*



*Figure 10: Tabular Cube created in SSAS*



*Figure 11: SQL Analysis Server*



II.5. Creating Power BI Reports

We connected the SSAS Tabular Cube to Power BI using the Get Data function and selected the appropriate database. DAX queries were utilized to create measures for display on the dashboard. Additionally, slicers were incorporated to enable efficient data filtering.

*Figure 12: DAX query to create measure*

A screenshot of a computer

Description automatically generated

The analysis of Sales was done for the following features: 3 years, 12 months, 4 Product Categories, Product subcategories, 6 Countries, 3 Continents and 2 Modalities (Internet and Reseller Sales). The statistics generated were Total Sales, Average Monthly Sales, Average Quarterly Sales, Average Weekly  
Sales, and Average Daily Sales.

## ****Conclusion****

This report captures the end-to-end process of building a data-driven BI solution using AdventureWorks2014. The implementation includes creating a robust data warehouse, performing ETL operations, building a tabular model for analysis, and visualizing insights through Power BI reports. The solution enables comprehensive sales analysis across multiple dimensions, supporting strategic decision-making.