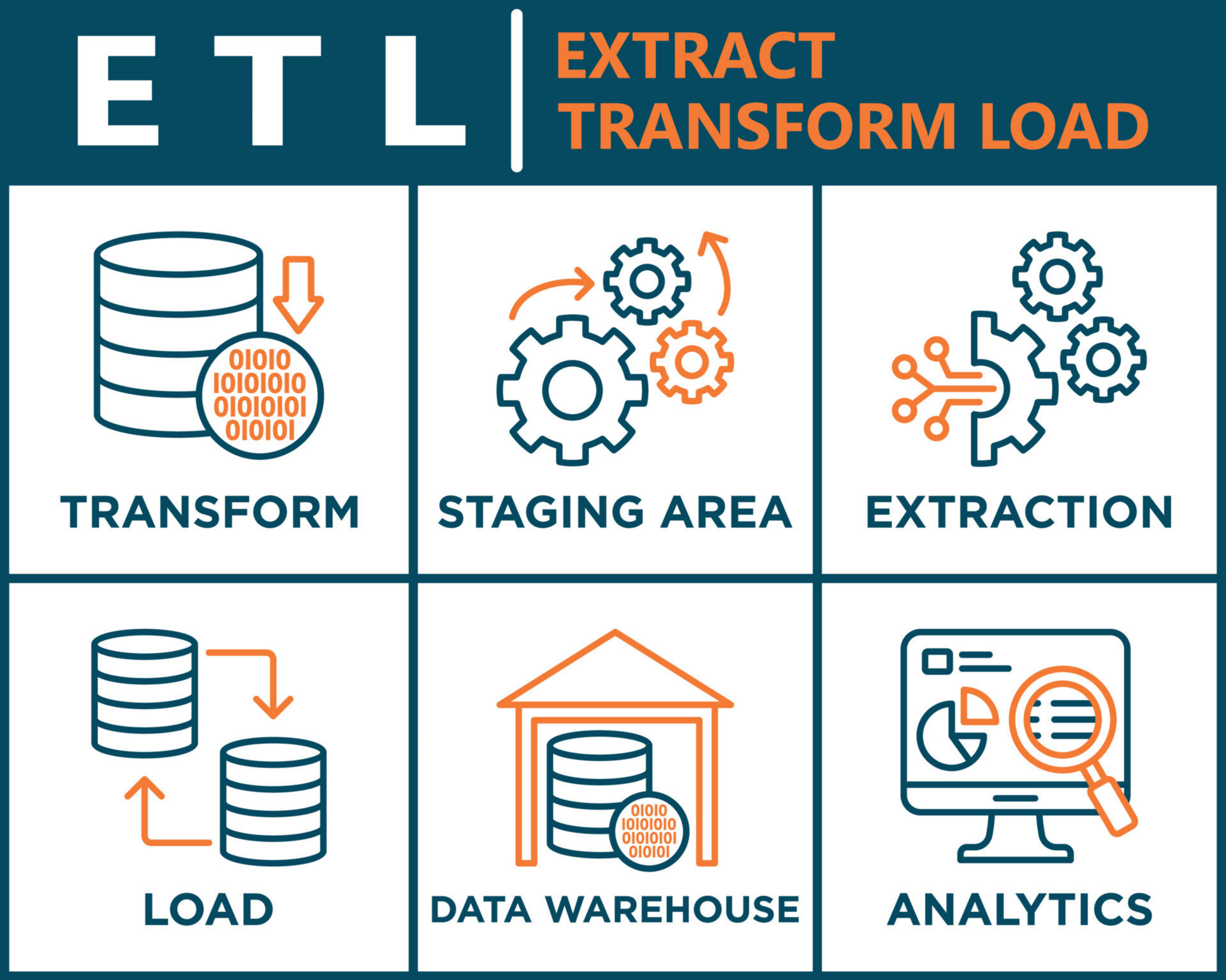


DESIGNING ETL PIPELINE

For My Own AdventureWorks DW (Sales)

Presented to:	Prof. Usman Shahzeb
Course:	DWDM
Presented by:	Asad Ali, SP21-BCS-007
Presented by:	Haroon Shahzad, SP21-BCS-017
Presented by:	Asad Ur Rehman, SP21-BCS-003

COMSATS University
Islamabad Lahore
Campus

Introduction to Data Warehouse and ETL Process:

A data warehouse is a centralized repository of integrated data from various sources within an organization. It is designed to support business intelligence and reporting activities by providing a structured and optimized environment for data analysis. Data warehouses are typically used to store historical and current data, enabling users to gain insights and make informed decisions based on the data.

ETL (Extract, Transform, Load) is a process used to extract data from multiple sources, transform it into a consistent format, and load it into a data warehouse. The ETL process involves extracting data from operational systems, applying various data transformations, and loading the transformed data into the data warehouse. This process ensures that data is cleansed, standardized, and organized in a way that is suitable for analysis and reporting.

Adventure Works Database and Adventure Works DW:

The Adventure Works database and Adventure Works data warehouse are provided by Microsoft as sample databases for learning and practicing data warehousing and business intelligence concepts.

1. **Adventure Works Database:**
   The Adventure Works database is a fictional sample database that represents a company selling bicycles and related products. It contains tables and data representing various aspects of the company's operations, such as customers, products, sales orders, and more. The Adventure Works database is designed to showcase the functionalities of a transactional database and serves as the source of data for populating the Adventure Works data warehouse.
2. **Adventure Works Data Warehouse:**
   The Adventure Works data warehouse is a pre-built data warehouse based on the Adventure Works database. It is designed to demonstrate best practices in data warehousing and provides a structured and optimized environment for data analysis and reporting. The Adventure Works data warehouse contains several dimension tables and a fact table, which together form the foundation for business intelligence and decision-making processes.

Dimensions and Fact Table in My AdventureWorks DW (Sales):

Date Dimension:
The Date Dimension table stores detailed information about dates. It includes attributes such as the date, year, month, day, day name, month name, and full date. This dimension allows for analyzing sales and other metrics based on specific dates or date ranges. It provides the ability to aggregate data at various levels, such as year, month, or day, and supports time-based analysis and trend identification.

Category Dimension:
The Category Dimension table represents different categories of products. It captures information about product categories, such as electronics, furniture, clothing, or accessories. Each category is assigned a unique category ID and is associated with products through relationships. This dimension is useful for analyzing sales, revenue, and other metrics based on different product categories, allowing for product category-based performance analysis.

Subcategory Dimension:
The Subcategory Dimension table represents subcategories of products, which are further classified within each category in the Category Dimension. Subcategories provide more granular information about the products. For example, within the electronics category, subcategories may include televisions, smartphones, or laptops. The Subcategory Dimension establishes a hierarchical relationship with the Category Dimension, allowing for deeper analysis and segmentation of products.

Product Dimension:
The Product Dimension table contains detailed information about products. It includes attributes such as the product ID, product name, and the subcategory to which the product belongs. This dimension provides comprehensive data about individual products, enabling analysis based on specific products or groups of products. It supports product-centric analysis, such as identifying top-selling products, analyzing product performance over time, and understanding the relationship between products and sales.

Vendor Dimension:
The Vendor Dimension table stores information about vendors or suppliers who provide the products. It includes attributes such as the vendor ID and vendor name. This dimension allows for analyzing vendor performance, tracking vendor contributions to sales, and evaluating vendor relationships. It enables businesses to gain insights into vendor performance, identify top vendors, and optimize vendor management strategies.

Country Dimension:
The Country Dimension table represents countries related to sales territories. It captures information about different countries, such as the country ID and country name. This dimension is useful for analyzing sales and performance across different countries, identifying regional trends, and understanding the impact of geography on sales. It enables businesses to analyze sales performance by country, compare sales across regions, and support international business analysis.

State Dimension:
The State Dimension table stores information about states or regions within countries. It includes attributes such as the state ID, country ID, and state name. This dimension enables regional analysis within specific countries or territories. It allows for grouping and analyzing sales data at the state or regional level, providing insights into localized trends, regional variations, and market dynamics. The State Dimension table establishes a relationship with the Country Dimension, enabling drill-down analysis from country to state-level data.

Sales Fact:

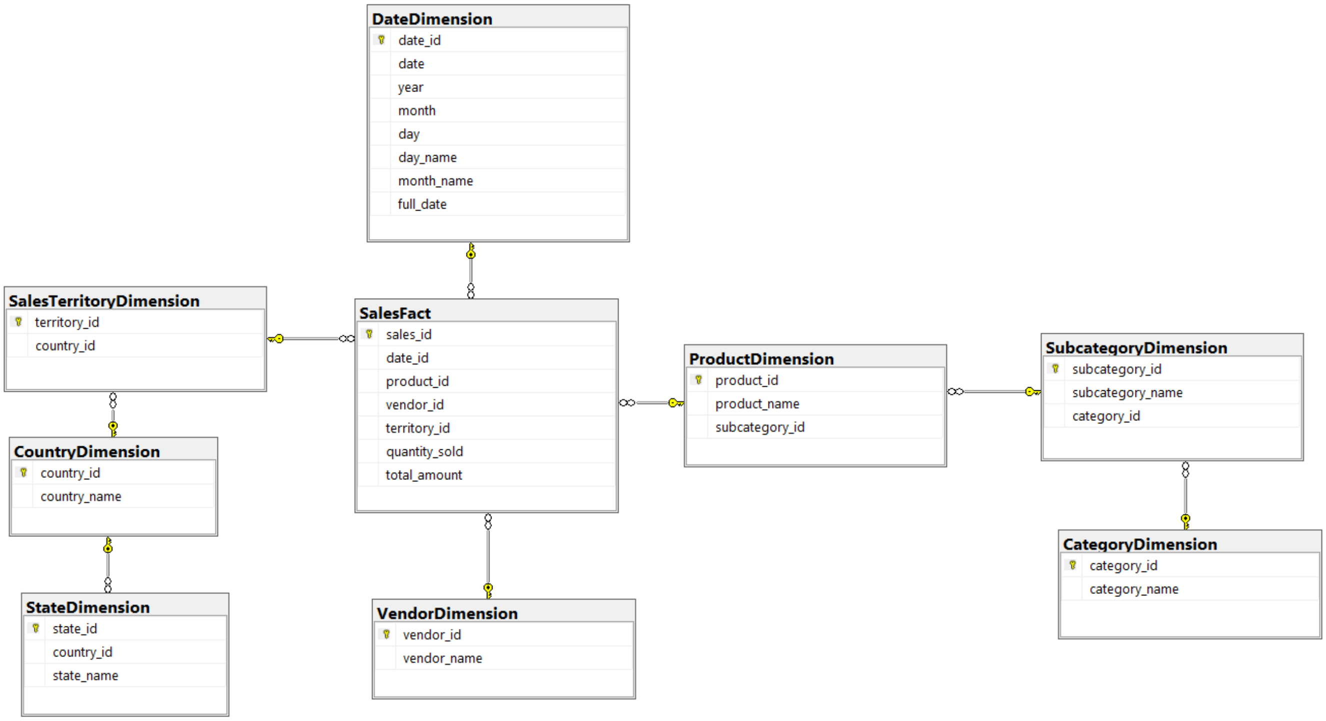
The Sales Fact table is a fundamental component of a data warehouse and serves as the central repository for capturing sales-related information. It contains detailed transactional data and measures related to sales activities. The Sales Fact table enables businesses to analyze and gain insights into their sales performance, identify trends, and make informed decisions. The Sales Fact table includes the following key attributes:

1. **Date ID:** This attribute references the Date Dimension and represents the specific date on which the sale occurred. It allows for time-based analysis, such as tracking sales trends over different periods, comparing sales performance on specific dates, or aggregating sales by year, month, or day.
2. **Product ID:** This attribute references the Product Dimension and identifies the specific product that was sold. It enables product-centric analysis, such as identifying the best-selling products, analyzing sales performance by product category or subcategory.

3. **Vendor ID:** This attribute references the Vendor Dimension and identifies the vendor or supplier associated with the sale. It allows businesses to track sales performance and relationships with different vendors, assess vendor contributions to overall sales, and optimize vendor management strategies.
4. **Sales Territory ID:** This attribute references the Sales Territory Dimension and represents the sales territory or geographical region associated with the sale. It allows for analyzing sales performance across different territories, identifying high-performing regions, and making data-driven decisions regarding sales expansion or resource allocation.
5. **Quantity Sold:** This measure captures the quantity or units of the product sold in a specific transaction. It provides insights into sales volume and allows for analyzing sales trends, forecasting demand, and identifying patterns related to product popularity or seasonality.
6. **Total Amount:** This measure represents the total monetary value or revenue generated from the sale. It provides a comprehensive view of the financial impact of sales activities and enables businesses to track revenue, assess profitability, and perform financial analysis.

The Sales Fact table serves as a key analytical asset within a data warehouse environment. It allows for aggregating and summarizing sales data based on different dimensions, such as date, product, vendor, or sales territory. With the Sales Fact table, businesses can perform various types of analysis, including but not limited to sales performance analysis, profitability analysis, market segmentation, forecasting, and trend identification. It forms the basis for generating meaningful reports, dashboards, and visualizations that aid in decision-making and strategy formulation.

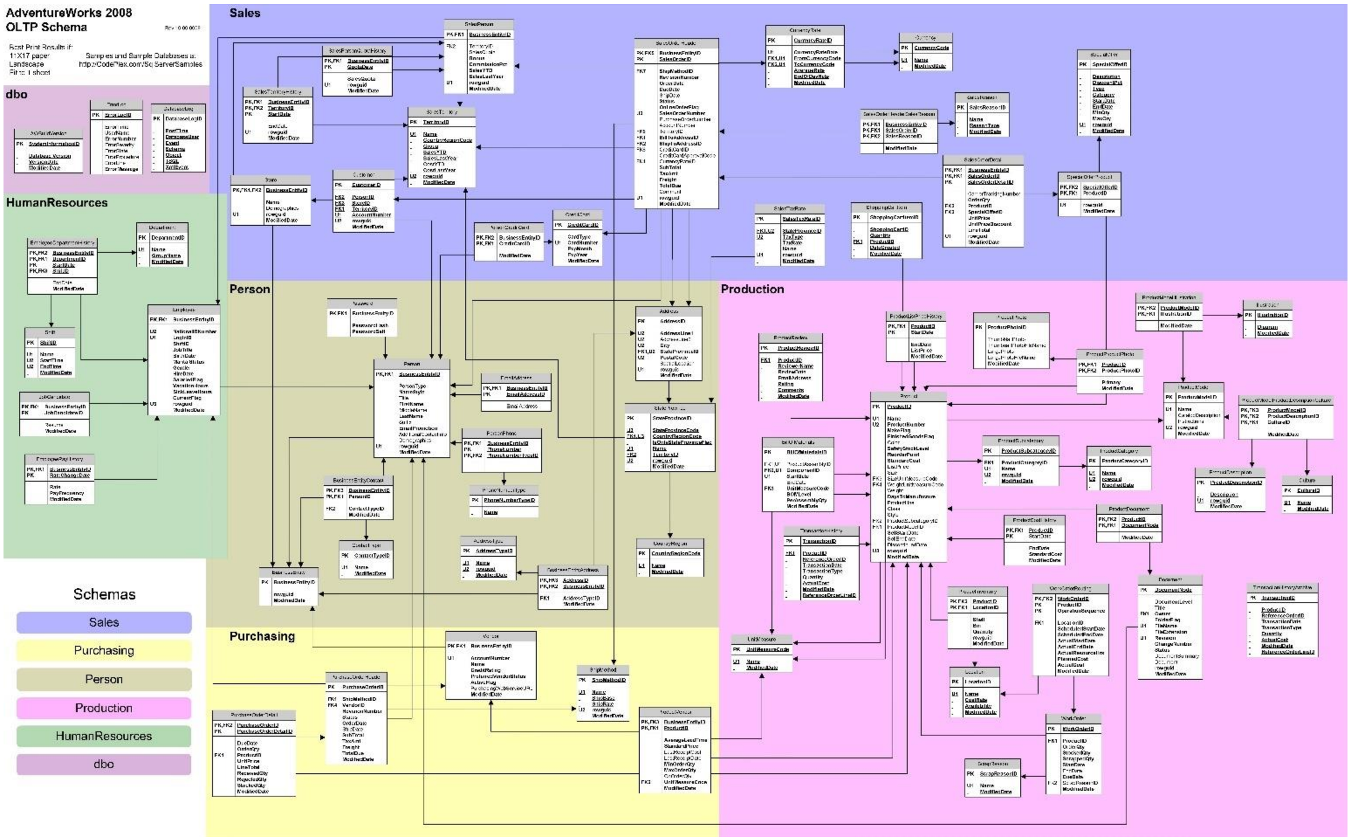
Star Schema of My Adventure Works DW:



These dimensions and the fact table provide a foundation for analyzing and reporting on sales data, allowing users to gain insights into product sales, vendor performance, and regional sales performance within the Adventure Works organization.

Creation Stages:

Adventure Works DB Used to Design my Adventure Works DW (Sales):



Script to Design My Warehouse’s Star Schema on MS SQL Server:

```sql
-- Drop Sales Fact Table
DROP TABLE IF EXISTS SalesFact;

-- Drop Product Dimension Table
DROP TABLE IF EXISTS ProductDimension;

-- Drop Vendor Dimension Table
DROP TABLE IF EXISTS VendorDimension;

-- Drop Subcategory Dimension Table
DROP TABLE IF EXISTS SubcategoryDimension;

-- Drop Category Dimension Table
DROP TABLE IF EXISTS CategoryDimension;

-- Drop Date Dimension Table
DROP TABLE IF EXISTS DateDimension;

-- Drop SalesTerritoryDimension Table
DROP TABLE IF EXISTS SalesTerritoryDimension;
```

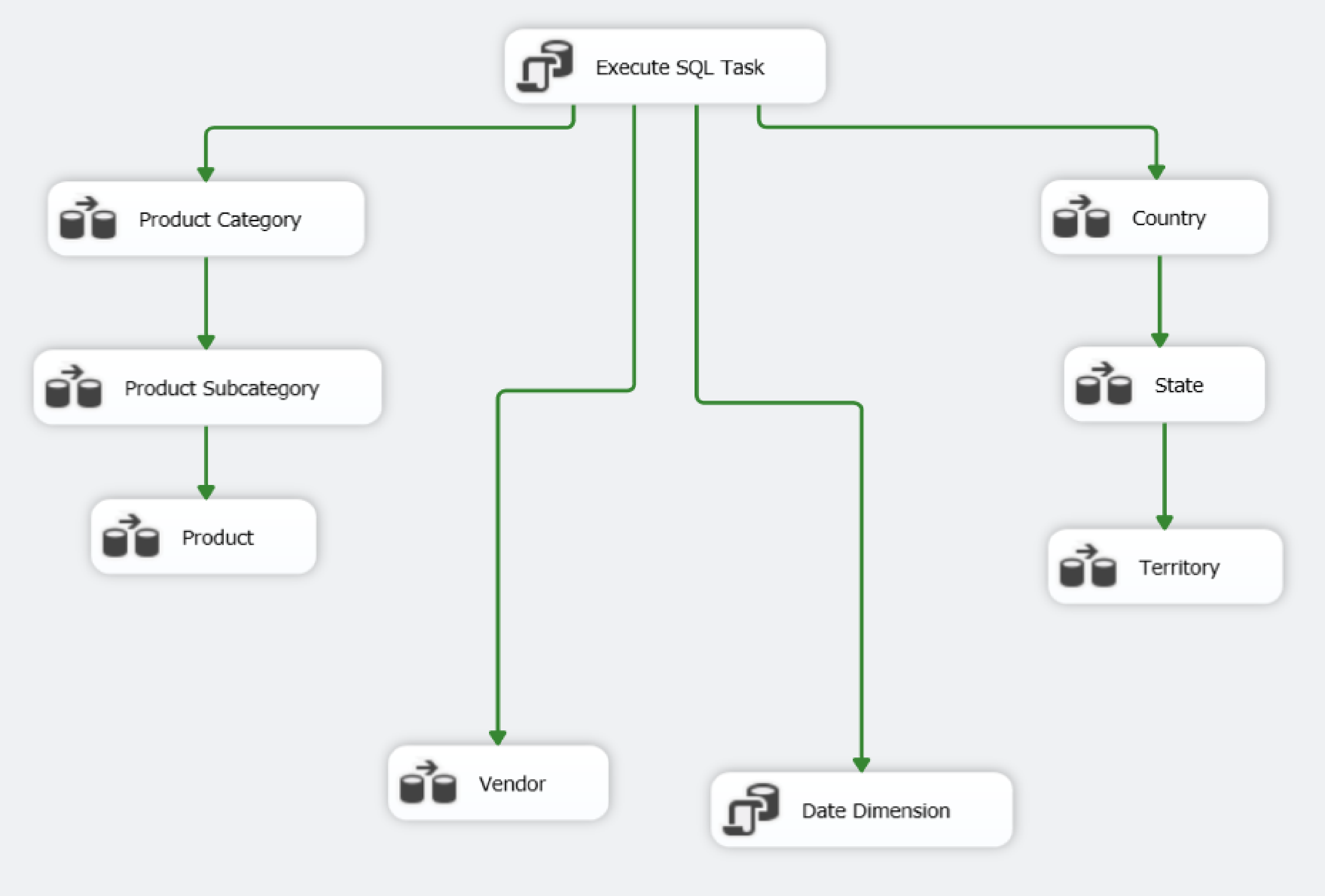
```sql
-- Drop StateDimension Table
DROP TABLE IF EXISTS StateDimension;

-- Drop CountryDimension Table
DROP TABLE IF EXISTS CountryDimension;

-- Date Dimension Table
CREATE TABLE DateDimension (
    date_id INT PRIMARY KEY,
    date DATE,
    year INT,
    month INT,
    day INT,
    day_name VARCHAR(20),
    month_name VARCHAR(20),
    full_date VARCHAR(50)
);

CREATE INDEX IDX_DateDimension_Year ON DateDimension (year);
CREATE INDEX IDX_DateDimension_Month ON DateDimension (month);
CREATE INDEX IDX_DateDimension_Day ON DateDimension (day);
CREATE INDEX IDX_DateDimension_MonthName ON DateDimension (month_name);

-- Category Dimension Table
CREATE TABLE CategoryDimension (
    category_id INT PRIMARY KEY,
    category_name NVARCHAR(255)
);

CREATE INDEX IDX_CategoryDimension_CategoryName ON CategoryDimension (category_name);

-- Subcategory Dimension Table
CREATE TABLE SubcategoryDimension (
    subcategory_id INT PRIMARY KEY,
    subcategory_name NVARCHAR(255),
    category_id INT,
    FOREIGN KEY (category_id) REFERENCES CategoryDimension(category_id)
);

CREATE INDEX IDX_SubcategoryDimension_SubcategoryName ON SubcategoryDimension
(subcategory_name);

-- Product Dimension Table
CREATE TABLE ProductDimension (
    product_id INT PRIMARY KEY,
    product_name NVARCHAR(255),
    subcategory_id INT,
    FOREIGN KEY (subcategory_id) REFERENCES SubcategoryDimension(subcategory_id)
);

CREATE INDEX IDX_ProductDimension_ProductName ON ProductDimension (product_name);
```

```sql
-- Vendor Dimension Table
CREATE TABLE VendorDimension (
    vendor_id INT PRIMARY KEY,
    vendor_name NVARCHAR(255)
);

CREATE INDEX IDX_VendorDimension_VendorName ON VendorDimension (vendor_name);

CREATE TABLE CountryDimension (
    country_id NVARCHAR(5) PRIMARY KEY,
    country_name NVARCHAR(255)
);

CREATE INDEX IDX_CountryDimension_country_name ON CountryDimension (country_name);

CREATE TABLE StateDimension (
    state_id NVARCHAR(5) PRIMARY KEY,
    country_id NVARCHAR(5),
    state_name NVARCHAR(255),

    FOREIGN KEY (country_id) REFERENCES CountryDimension(country_id),
);

CREATE INDEX IDX_country_id ON StateDimension (country_id);
CREATE INDEX IDX_state_name ON StateDimension (state_name);

-- Sales Territory Dimension Table
CREATE TABLE SalesTerritoryDimension (
    territory_id INT PRIMARY KEY,
    country_id NVARCHAR(5),

    FOREIGN KEY (country_id) REFERENCES CountryDimension(country_id),
);

CREATE INDEX IDX_SalesTerritoryDimension_state_country_id ON SalesTerritoryDimension
(country_id);

-- Sales Fact Table
CREATE TABLE SalesFact (
    sales_id INT IDENTITY(1, 1) PRIMARY KEY,
    date_id INT,
    product_id INT,
    vendor_id INT,
    territory_id INT,
    quantity_sold INT,
    total_amount DECIMAL(10, 2),
    FOREIGN KEY (date_id) REFERENCES DateDimension(date_id),
    FOREIGN KEY (product_id) REFERENCES ProductDimension(product_id),
    FOREIGN KEY (vendor_id) REFERENCES VendorDimension(vendor_id),
    FOREIGN KEY (territory_id) REFERENCES SalesTerritoryDimension(territory_id)
);
```

```
CREATE INDEX IDX_SalesFact_DateId ON SalesFact (date_id);
CREATE INDEX IDX_SalesFact_ProductId ON SalesFact (product_id);
CREATE INDEX IDX_SalesFact_VendorId ON SalesFact (vendor_id);
CREATE INDEX IDX_SalesFact_QuantitySold ON SalesFact (quantity_sold);
CREATE INDEX IDX_SalesFact_TotalAmount ON SalesFact (total_amount);
```

Some Screenshots of ETL Pipeline from SSIS Project:

All Dimensions:

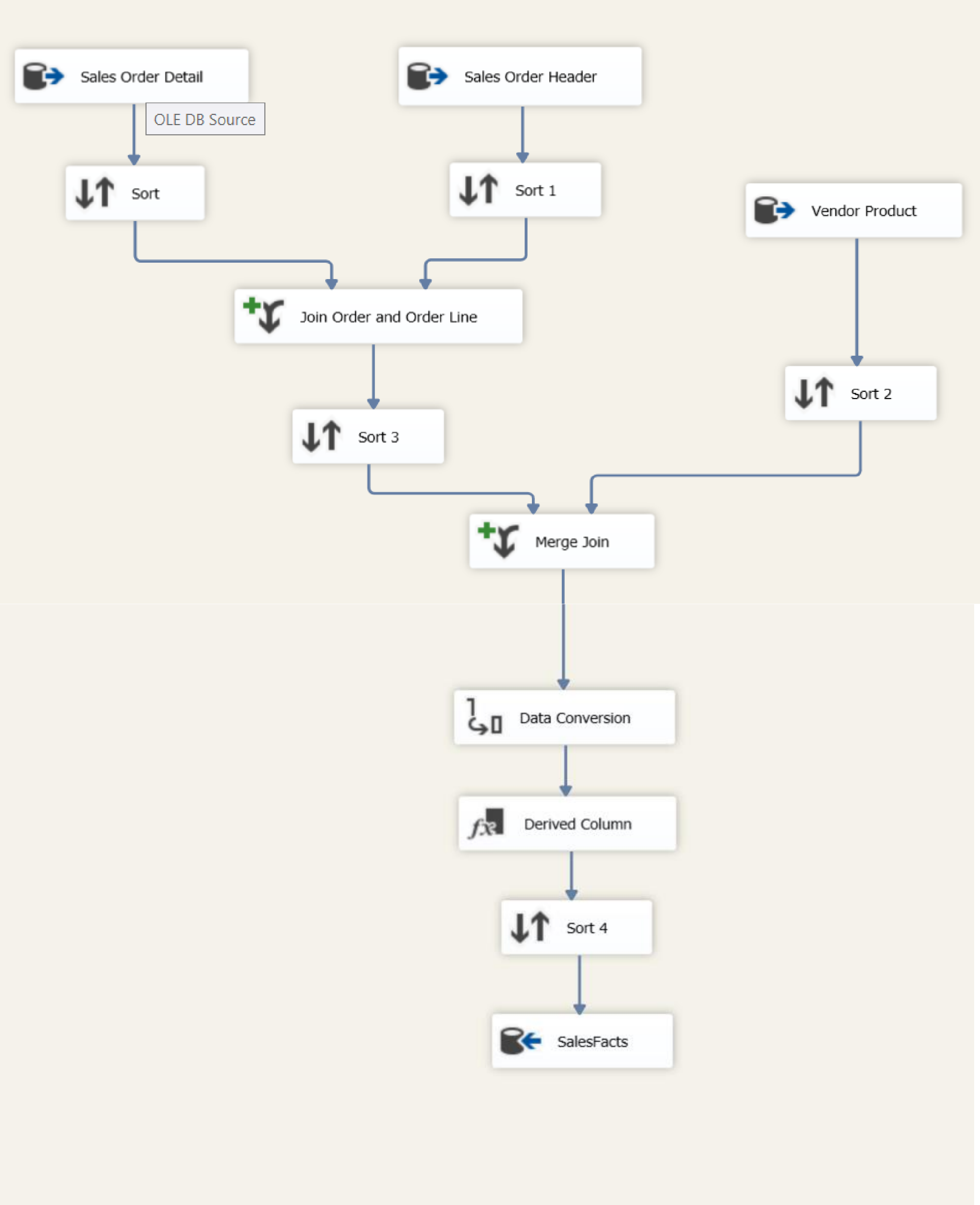


Execute SQL Task executes the SQL script given in previous point. This will use destructive merge i.e., will delete all the previous data in each table and refresh the whole DW on every load.

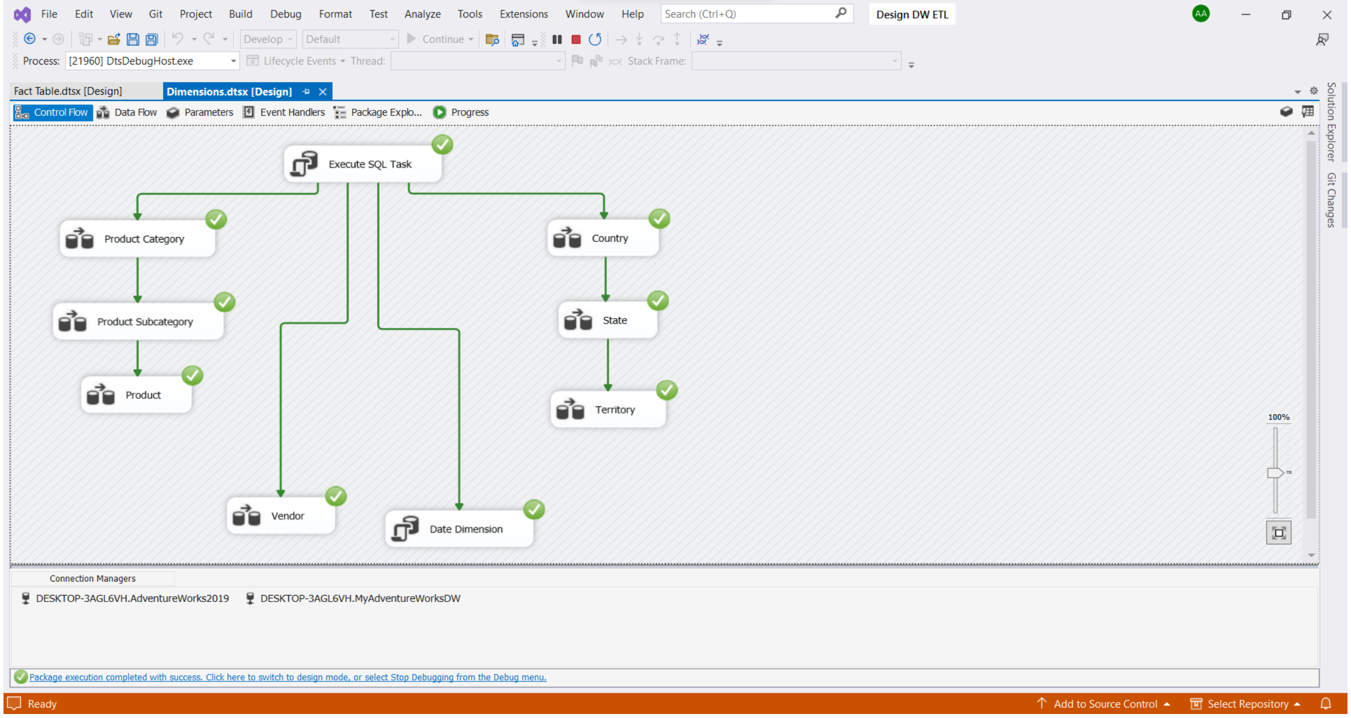
All dimensions are created by just loading data from the different database table using Sort, Merge and Calculated Columns to data warehouse table.

While the Date Dimension is created by using a script which fills up the table by using a loop to enter each date between a specific two dates in my case, I fill the table for six years (2011 to 2016).

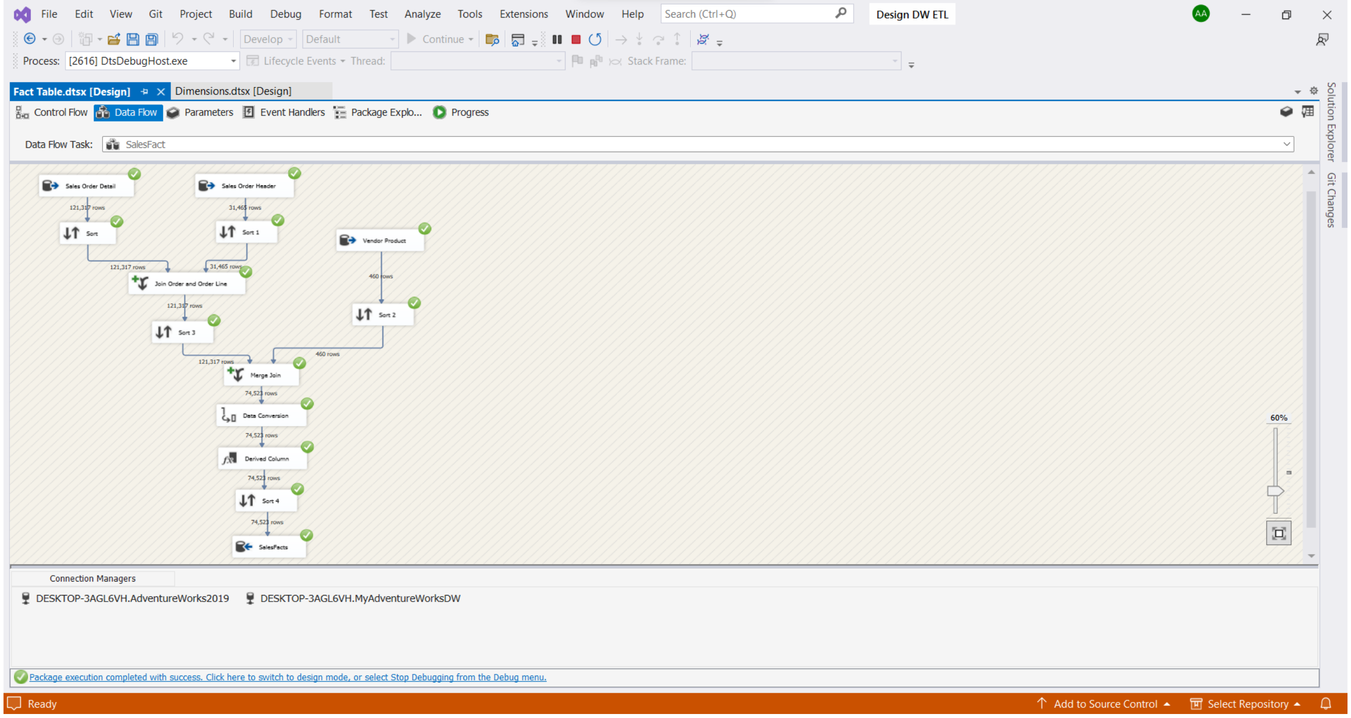
Sales Fact Table:



Debugging in Visual Studio Code:



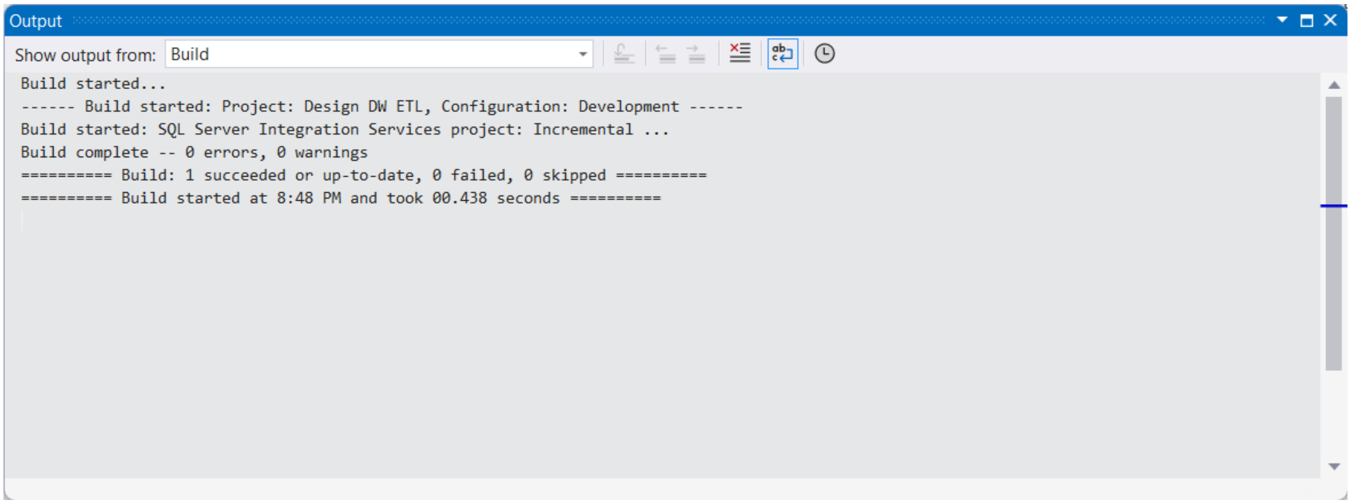
Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.



Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

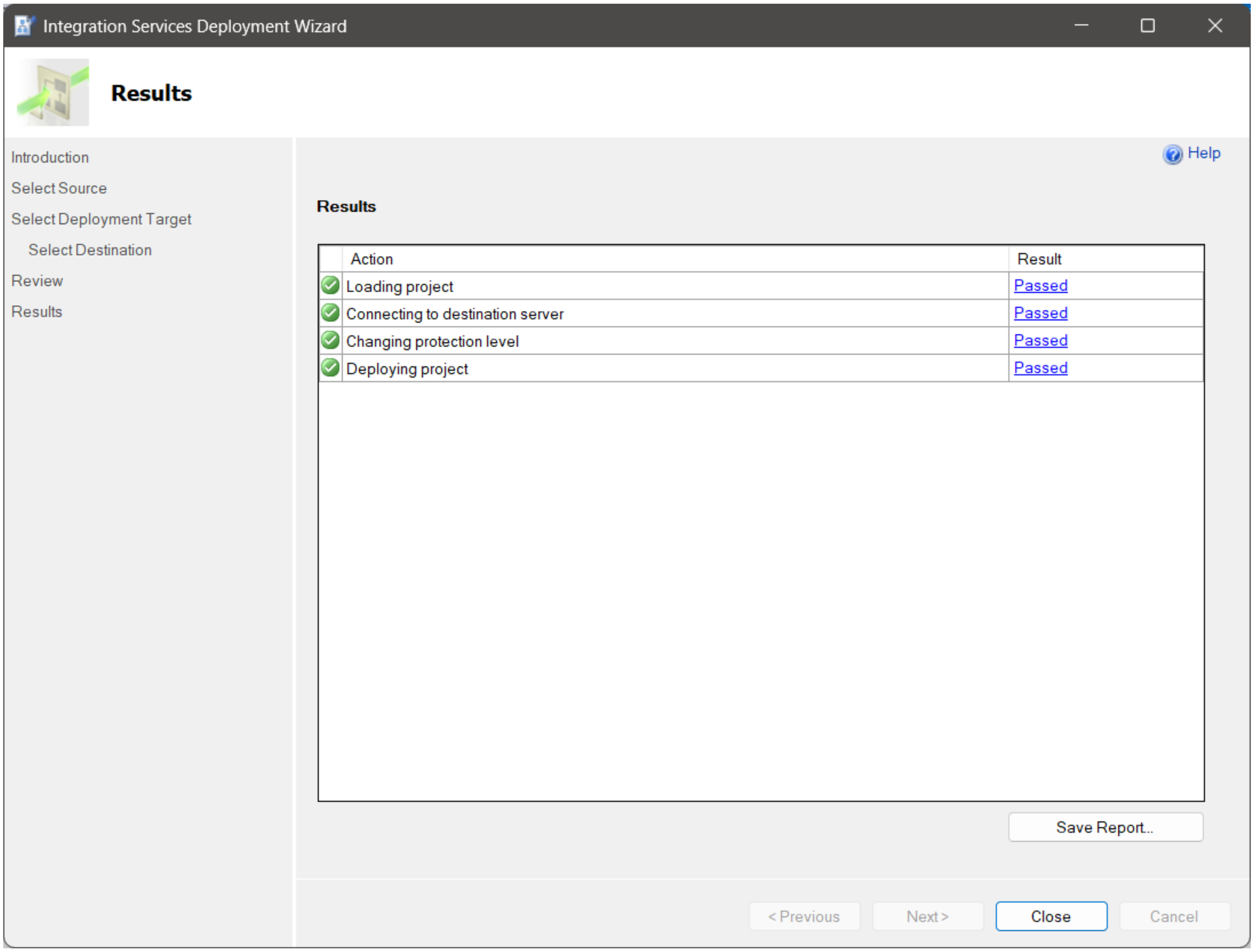
As you can see, we don’t have any errors in any of our ETL packages and these pipelines are filling in the data without any bug. After testing and debugging we will deploy our project on Microsoft SQL Server and will automate our work by scheduling a task using SQL Server Agent.

Build and Deploy on MS SQL Server:



```
Build started...
------ Build started: Project: Design DW ETL, Configuration: Development ------
Build started: SQL Server Integration Services project: Incremental ...
Build complete -- 0 errors, 0 warnings
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
========== Build started at 8:48 PM and took 00.438 seconds ==========
```

Project build with 0 errors and 0 warnings. Now we will deploy it on MS SQL Server.

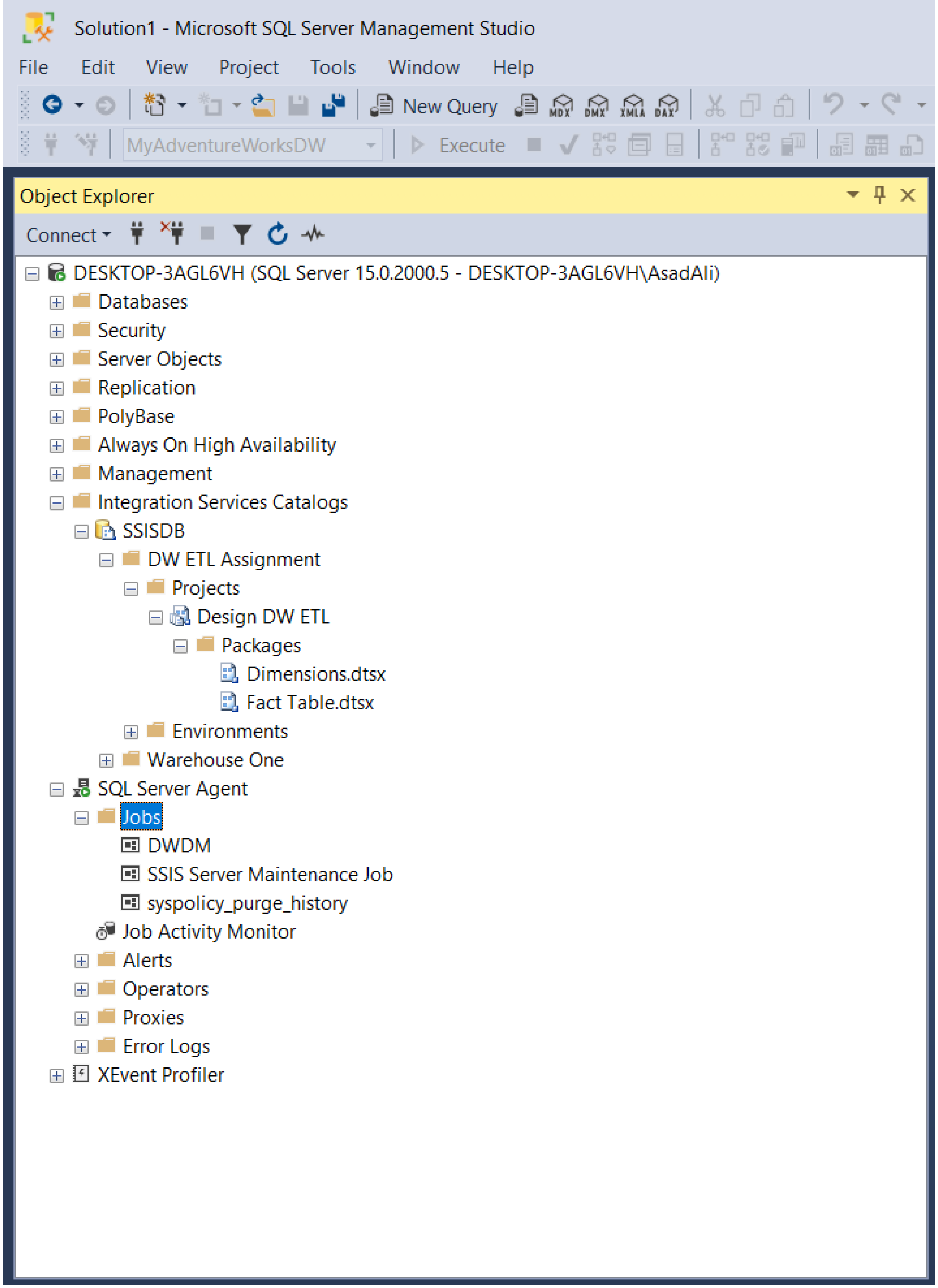


Results

Action	Result
✓ Loading project	Passed
✓ Connecting to destination server	Passed
✓ Changing protection level	Passed
✓ Deploying project	Passed

Project has been deployed successfully at SSISDB/DW ETL Assignment.

Here is the deployed project with two Packages We have designed.

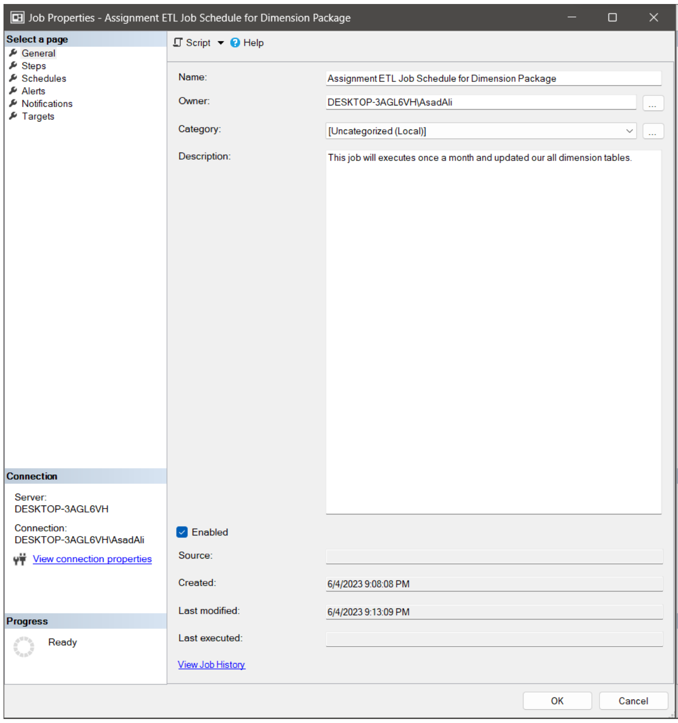


```
⊟ DESKTOP-3AGL6VH (SQL Server 15.0.2000.5 - DESKTOP-3AGL6VH\AsadAli)
    ⊞ Databases
    ⊞ Security
    ⊞ Server Objects
    ⊞ Replication
    ⊞ PolyBase
    ⊞ Always On High Availability
    ⊞ Management
    ⊟ Integration Services Catalogs
        ⊟ SSISDB
            ⊟ DW ETL Assignment
                ⊟ Projects
                    ⊟ Design DW ETL
                        ⊟ Packages
                              Dimensions.dtsx
                              Fact Table.dtsx
                ⊞ Environments
            ⊞ Warehouse One
    ⊟ SQL Server Agent
        ⊟ Jobs
              DWDM
              SSIS Server Maintenance Job
              syspolicy_purge_history
          Job Activity Monitor
        ⊞ Alerts
        ⊞ Operators
        ⊞ Proxies
        ⊞ Error Logs
    ⊞ XEvent Profiler
```

Assigning and Scheduling a Job Using SQL Server Agent:

Our scheduled jobs will be of two types, one for our dimensions packages which will be executed once a month and second will be for our sales fact packages which will be executed once a day.

Here are the screenshots above both jobs side by side:



Job Properties - Assignment ETL Job Schedule for Dimension Package

Name: Assignment ETL Job Schedule for Dimension Package

Owner: DESKTOP-3AGL6VH\AsadAli

Category: [Uncategorized (Local)]

Description: This job will executes once a month and updated our all dimension tables.

☑ Enabled

Source:

Created: 6/4/2023 9:08:08 PM

Last modified: 6/4/2023 9:13:09 PM

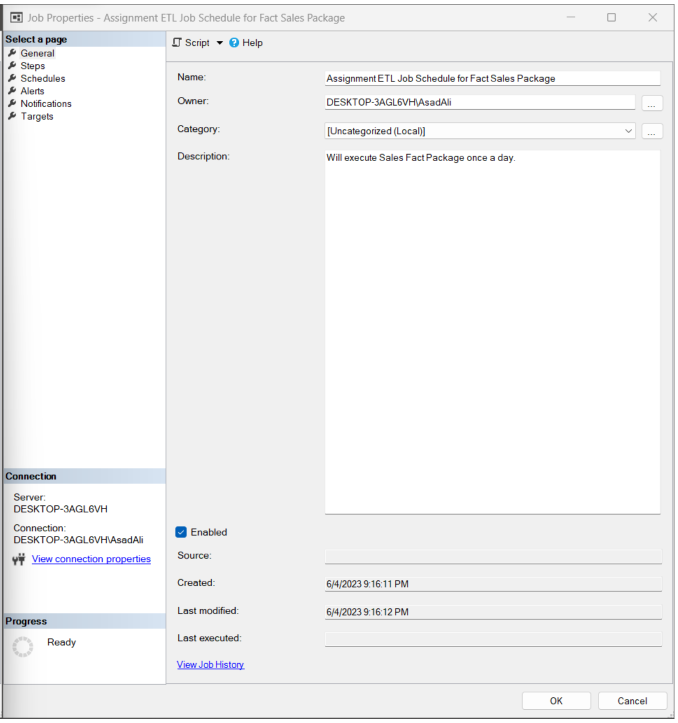
Last executed:

Connection

Server:
DESKTOP-3AGL6VH

Connection:
DESKTOP-3AGL6VH\AsadAli

Progress: Ready



Job Properties - Assignment ETL Job Schedule for Fact Sales Package

Name: Assignment ETL Job Schedule for Fact Sales Package

Owner: DESKTOP-3AGL6VH\AsadAli

Category: [Uncategorized (Local)]

Description: Will execute Sales Fact Package once a day.

☑ Enabled

Source:

Created: 6/4/2023 9:16:11 PM

Last modified: 6/4/2023 9:16:12 PM

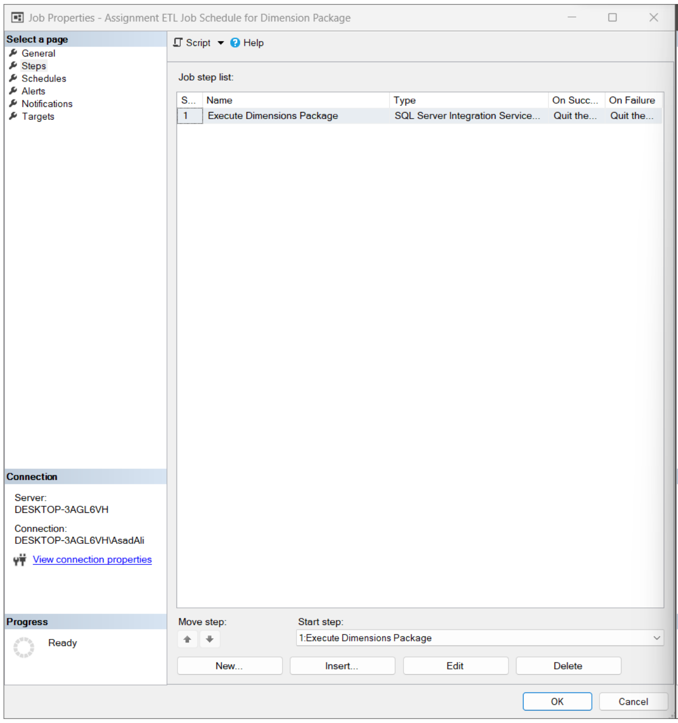
Last executed:

Connection

Server:
DESKTOP-3AGL6VH

Connection:
DESKTOP-3AGL6VH\AsadAli

Progress: Ready



Job Properties - Assignment ETL Job Schedule for Dimension Package

Job step list:

S...	Name	Type	On Succ...	On Failure
1	Execute Dimensions Package	SQL Server Integration Service...	Quit the...	Quit the...

Move step:

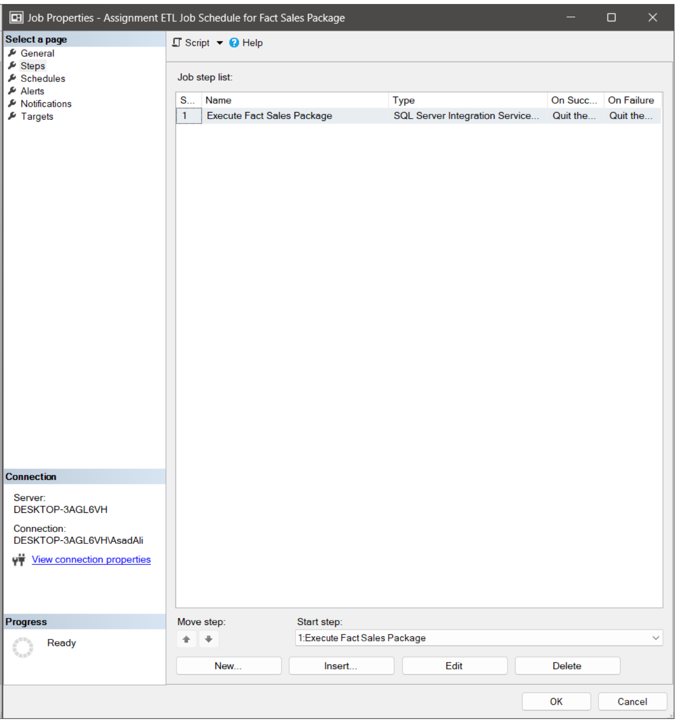
Start step: 1:Execute Dimensions Package

Connection

Server:
DESKTOP-3AGL6VH

Connection:
DESKTOP-3AGL6VH\AsadAli

Progress: Ready



Job Properties - Assignment ETL Job Schedule for Fact Sales Package

Job step list:

S...	Name	Type	On Succ...	On Failure
1	Execute Fact Sales Package	SQL Server Integration Service...	Quit the...	Quit the...

Move step:

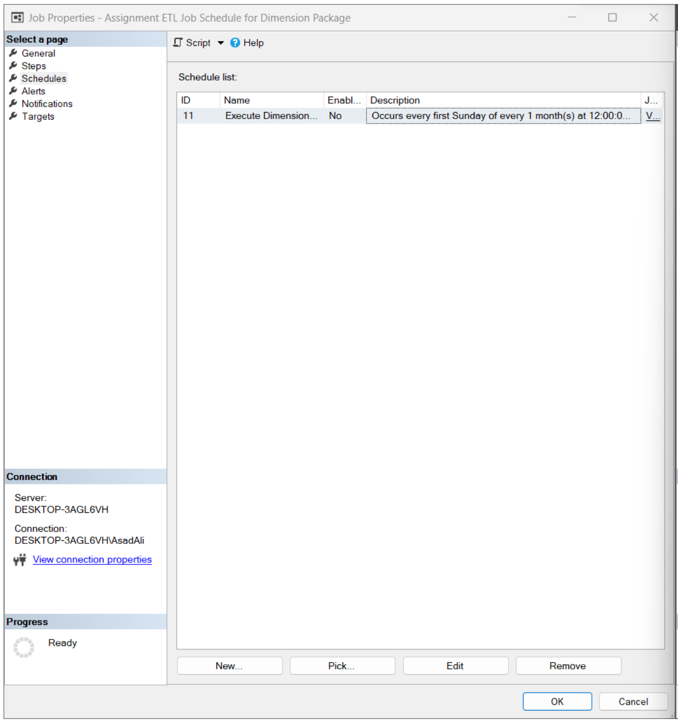
Start step: 1:Execute Fact Sales Package

Connection

Server:
DESKTOP-3AGL6VH

Connection:
DESKTOP-3AGL6VH\AsadAli

Progress: Ready



Job Properties - Assignment ETL Job Schedule for Dimension Package

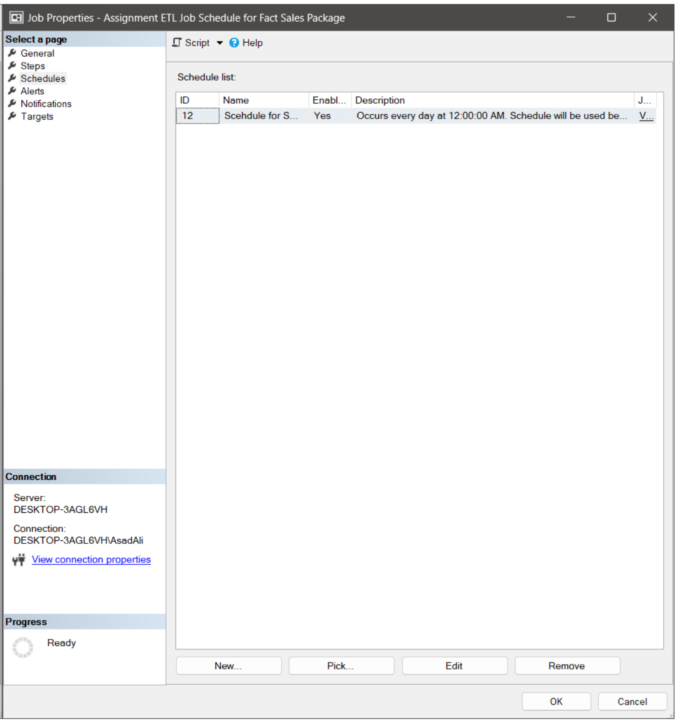
Schedule list:

ID	Name	Enabl...	Description	J...
11	Execute Dimension...	No	Occurs every first Sunday of every 1 month(s) at 12:00:0...	V...

Connection

Server:
DESKTOP-3AGL6VH

Connection:
DESKTOP-3AGL6VH\AsadAli



Job Properties - Assignment ETL Job Schedule for Fact Sales Package

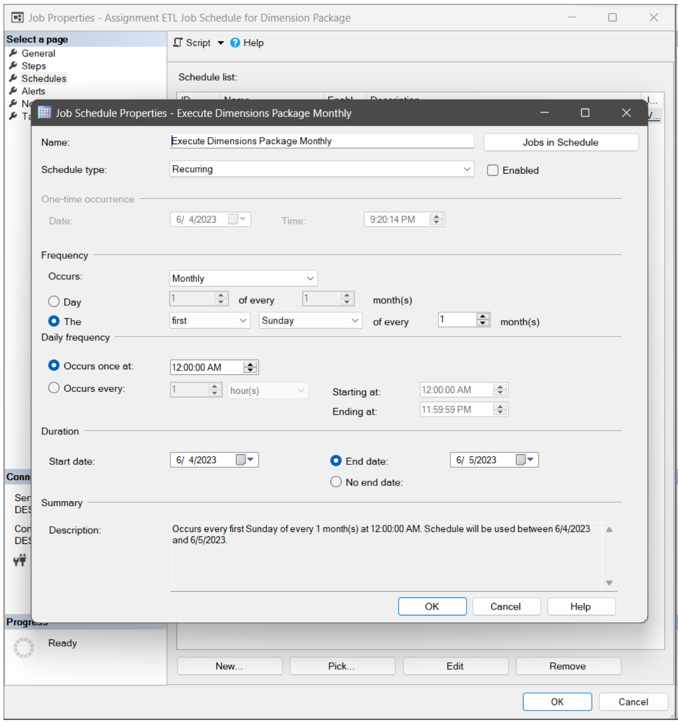
Schedule list:

ID	Name	Enabl...	Description	J...
12	Scehdule for S...	Yes	Occurs every day at 12:00:00 AM. Schedule will be used be...	V...

Connection

Server:
DESKTOP-3AGL6VH

Connection:
DESKTOP-3AGL6VH\AsadAli



Job Properties - Assignment ETL Job Schedule for Dimension Package

Job Schedule Properties - Execute Dimensions Package Monthly

Name: Execute Dimensions Package Monthly

Schedule type: Recurring □ Enabled

One-time occurrence

Date: 6/ 4/2023 Time: 9:20:14 PM

Frequency

Occurs: Monthly

○ Day 1 of every 1 month(s)

● The first Sunday of every 1 month(s)

Daily frequency

● Occurs once at: 12:00:00 AM

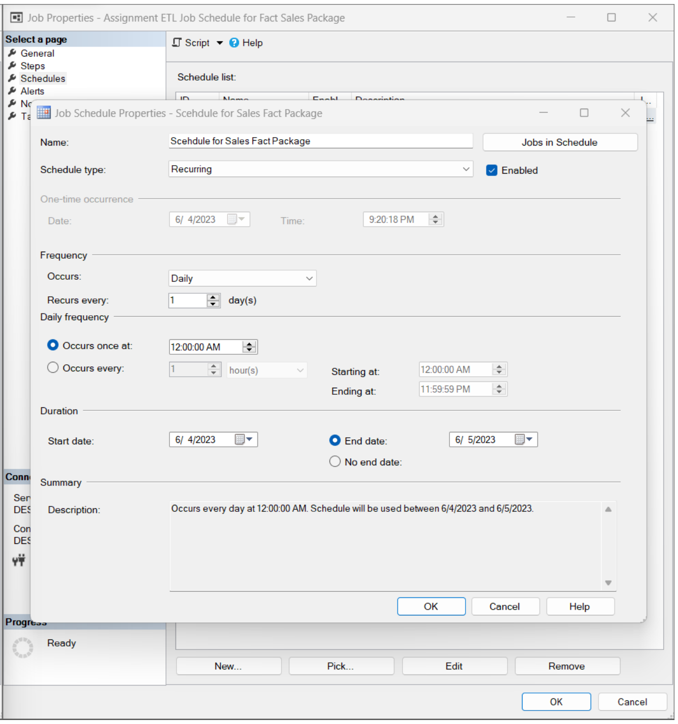
○ Occurs every: 1 hour(s) Starting at: 12:00:00 AM
Ending at: 11:59:59 PM

Duration

Start date: 6/ 4/2023 ● End date: 6/ 5/2023
○ No end date:

Summary

Description: Occurs every first Sunday of every 1 month(s) at 12:00:00 AM. Schedule will be used between 6/4/2023 and 6/5/2023.



Job Properties - Assignment ETL Job Schedule for Fact Sales Package

Job Schedule Properties - Scehdule for Sales Fact Package

Name: Scehdule for Sales Fact Package

Schedule type: Recurring ☑ Enabled

One-time occurrence

Date: 6/ 4/2023 Time: 9:20:18 PM

Frequency

Occurs: Daily

Recurs every: 1 day(s)

Daily frequency

● Occurs once at: 12:00:00 AM

○ Occurs every: 1 hour(s) Starting at: 12:00:00 AM
Ending at: 11:59:59 PM

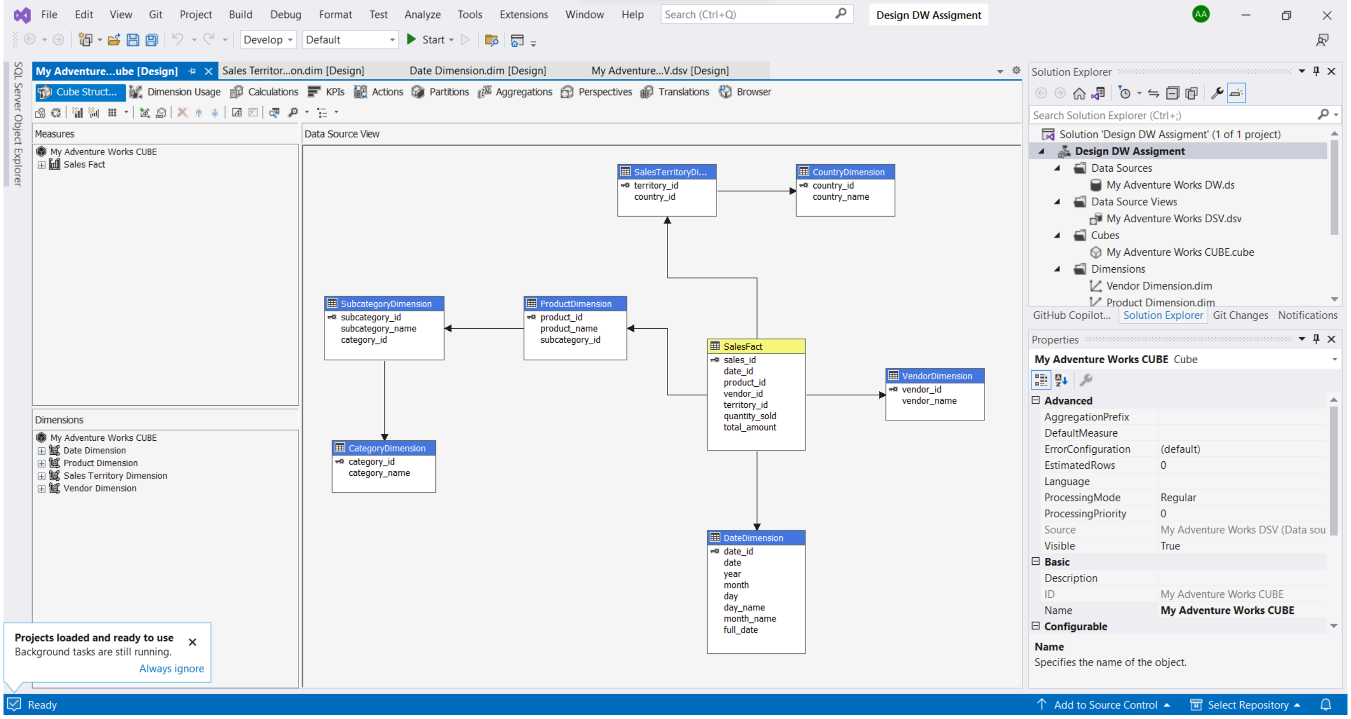
Duration

Start date: 6/ 4/2023 ● End date: 6/ 5/2023
○ No end date:

Summary

Description: Occurs every day at 12:00:00 AM. Schedule will be used between 6/4/2023 and 6/5/2023.

Sales Cube Using My Adventure Works DW:



Projects loaded and ready to use
Background tasks are still running.
Always ignore

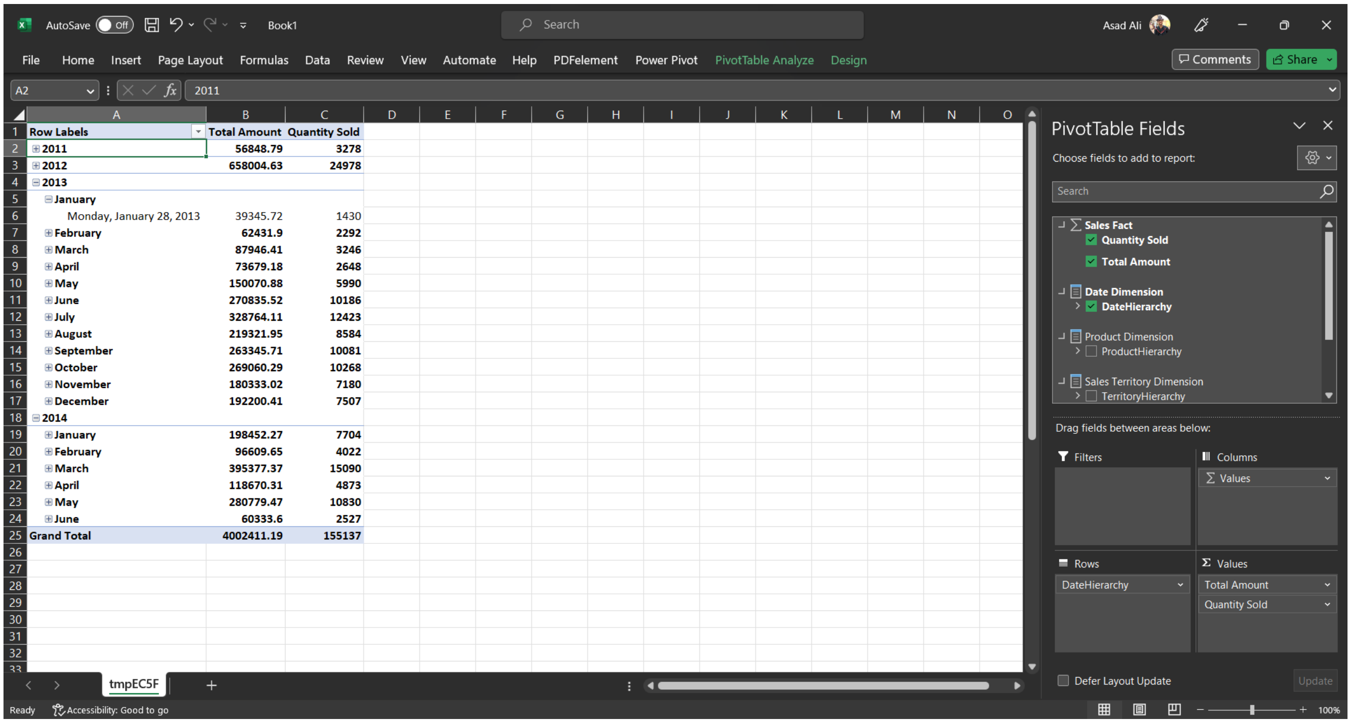
My Adventure Works CUBE Cube

Advanced	
AggregationPrefix	
DefaultMeasure	
ErrorConfiguration	(default)
EstimatedRows	0
Language	
ProcessingMode	Regular
ProcessingPriority	0
Source	My Adventure Works DSV (Data sou
Visible	True
Basic	
Description	
ID	My Adventure Works CUBE
Name	**My Adventure Works CUBE**
Configurable	

Name
Specifies the name of the object.

Some Analysis on My Own DW Sales Cube:

Sales by Date:



Row Labels	Total Amount	Quantity Sold
⊞2011	56848.79	3278
⊞2012	658004.63	24978
⊟2013		
⊟January		
Monday, January 28, 2013	39345.72	1430
⊞February	62431.9	2292
⊞March	87946.41	3246
⊞April	73679.18	2648
⊞May	150070.88	5990
⊞June	270835.52	10186
⊞July	328764.11	12423
⊞August	219321.95	8584
⊞September	263345.71	10081
⊞October	269060.29	10268
⊞November	180333.02	7180
⊞December	192200.41	7507
⊟2014		
⊞January	198452.27	7704
⊞February	96609.65	4022
⊞March	395377.37	15090
⊞April	118670.31	4873
⊞May	280779.47	10830
⊞June	60333.6	2527
Grand Total	4002411.19	155137

PivotTable Fields
Choose fields to add to report:

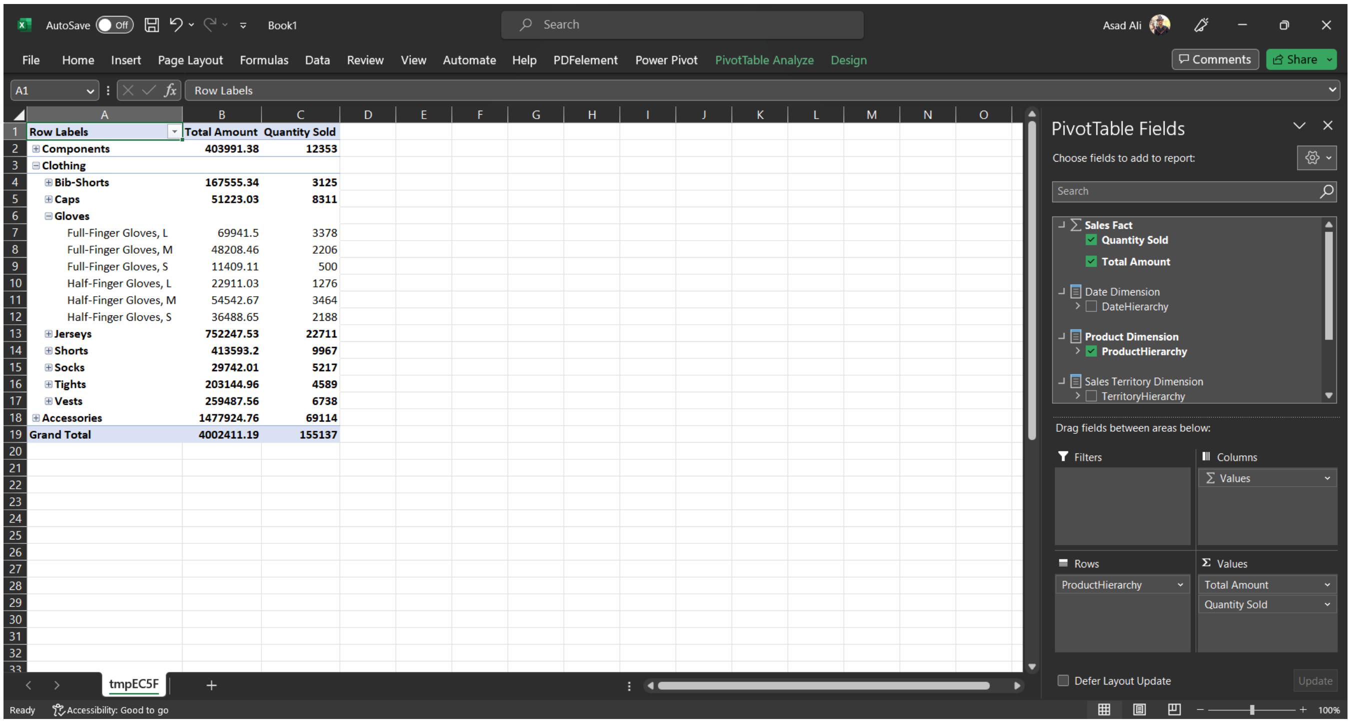
- Sales Fact
  - ☑ Quantity Sold
  - ☑ Total Amount
- Date Dimension
  - ☑ DateHierarchy
- Product Dimension
  - ☐ ProductHierarchy
- Sales Territory Dimension
  - ☐ TerritoryHierarchy

Drag fields between areas below:

Filters:
Columns: Values
Rows: DateHierarchy
Values: Total Amount, Quantity Sold

☐ Defer Layout Update
Update

Sales by Category and Subcategory:



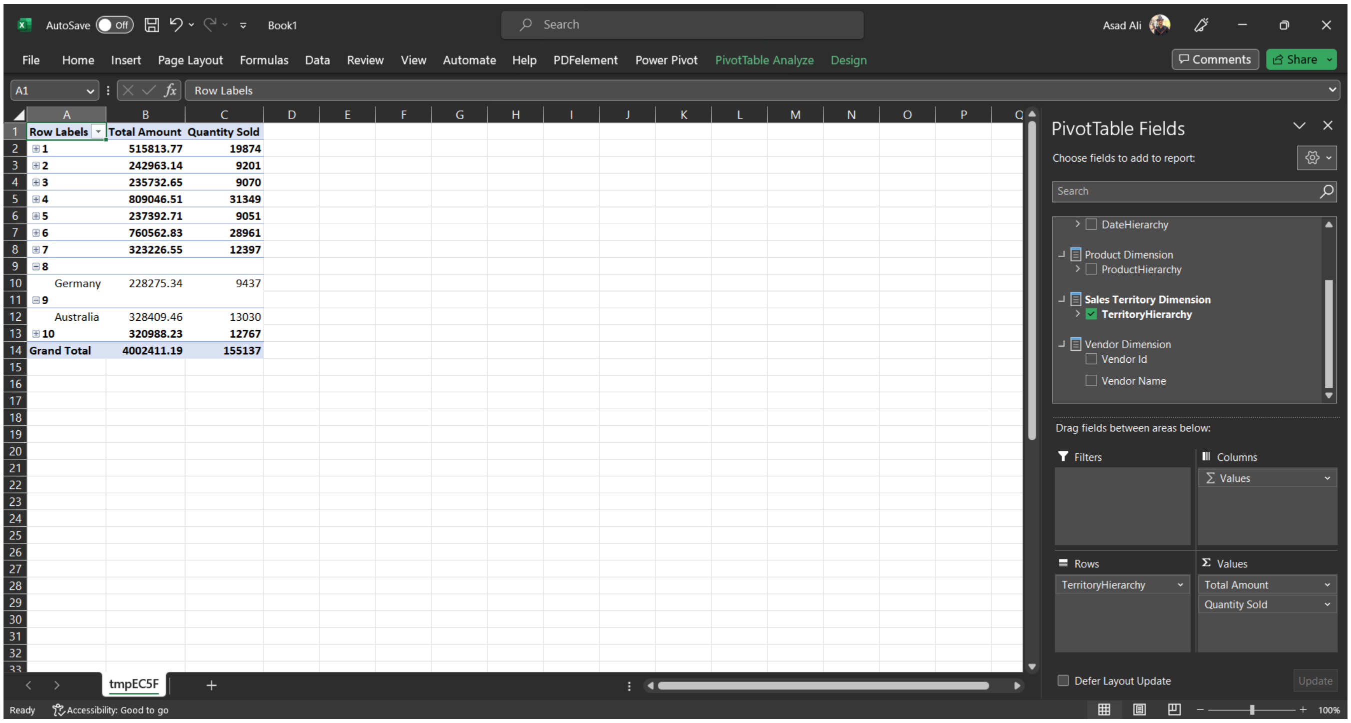
Row Labels	Total Amount	Quantity Sold
⊞ **Components**	**403991.38**	**12353**
⊟ **Clothing**		
⊞ **Bib-Shorts**	**167555.34**	**3125**
⊞ **Caps**	**51223.03**	**8311**
⊟ **Gloves**		
Full-Finger Gloves, L	69941.5	3378
Full-Finger Gloves, M	48208.46	2206
Full-Finger Gloves, S	11409.11	500
Half-Finger Gloves, L	22911.03	1276
Half-Finger Gloves, M	54542.67	3464
Half-Finger Gloves, S	36488.65	2188
⊞ **Jerseys**	**752247.53**	**22711**
⊞ **Shorts**	**413593.2**	**9967**
⊞ **Socks**	**29742.01**	**5217**
⊞ **Tights**	**203144.96**	**4589**
⊞ **Vests**	**259487.56**	**6738**
⊞ **Accessories**	**1477924.76**	**69114**
Grand Total	**4002411.19**	**155137**

PivotTable Fields

Sales Fact: ☑ Quantity Sold, ☑ Total Amount
Date Dimension: □ DateHierarchy
Product Dimension: ☑ ProductHierarchy
Sales Territory Dimension: □ TerritoryHierarchy

Columns: Σ Values
Rows: ProductHierarchy
Values: Total Amount, Quantity Sold

Sales by Territory and Country:



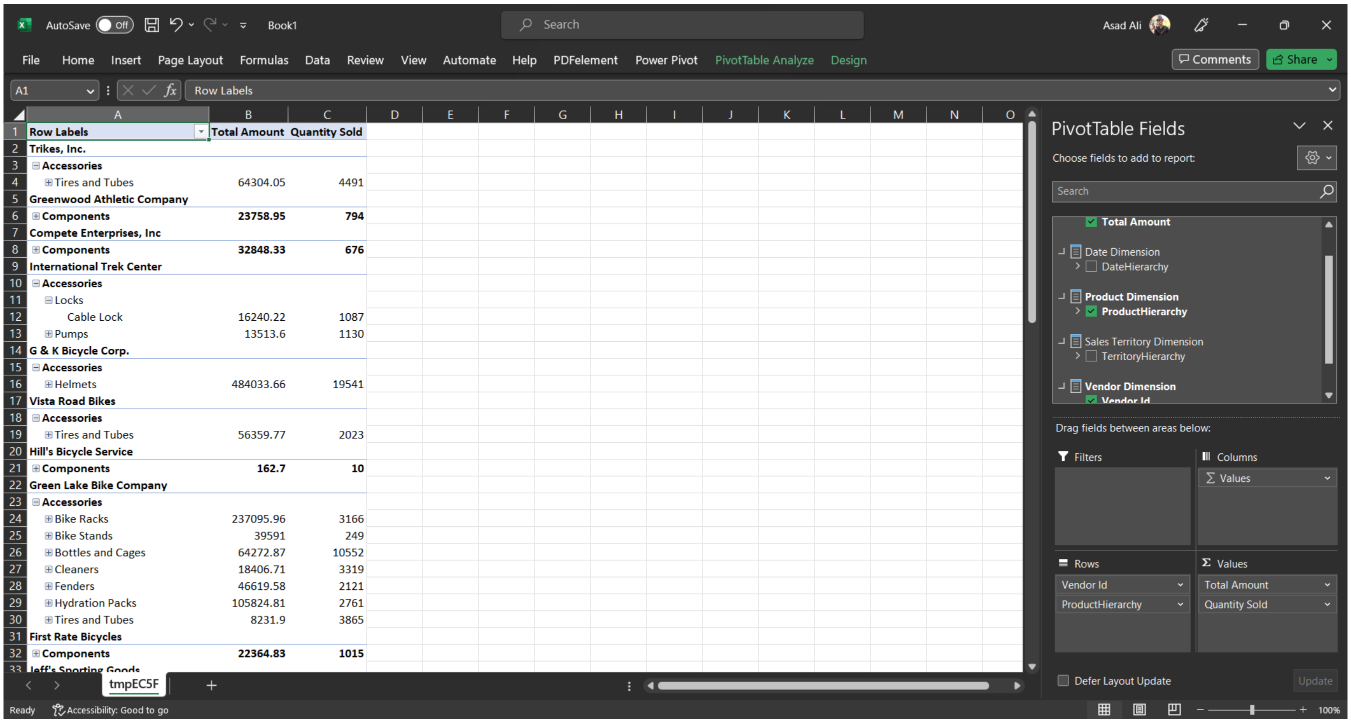
Row Labels	Total Amount	Quantity Sold
⊞ **1**	**515813.77**	**19874**
⊞ **2**	**242963.14**	**9201**
⊞ **3**	**235732.65**	**9070**
⊞ **4**	**809046.51**	**31349**
⊞ **5**	**237392.71**	**9051**
⊞ **6**	**760562.83**	**28961**
⊞ **7**	**323226.55**	**12397**
⊟ **8**		
Germany	228275.34	9437
⊟ **9**		
Australia	328409.46	13030
⊞ **10**	**320988.23**	**12767**
Grand Total	**4002411.19**	**155137**

PivotTable Fields

□ DateHierarchy
Product Dimension: □ ProductHierarchy
Sales Territory Dimension: ☑ TerritoryHierarchy
Vendor Dimension: □ Vendor Id, □ Vendor Name

Columns: Σ Values
Rows: TerritoryHierarchy
Values: Total Amount, Quantity Sold

Sales by Vendor and Category Hierarchy:



Row Labels	Total Amount	Quantity Sold
Trikes, Inc.		
⊟ Accessories		
⊞ Tires and Tubes	64304.05	4491
Greenwood Athletic Company		
⊞ Components	**23758.95**	**794**
Compete Enterprises, Inc		
⊞ Components	**32848.33**	**676**
International Trek Center		
⊟ Accessories		
⊟ Locks		
Cable Lock	16240.22	1087
⊞ Pumps	13513.6	1130
G & K Bicycle Corp.		
⊟ Accessories		
⊞ Helmets	484033.66	19541
Vista Road Bikes		
⊟ Accessories		
⊞ Tires and Tubes	56359.77	2023
Hill's Bicycle Service		
⊞ Components	**162.7**	**10**
Green Lake Bike Company		
⊟ Accessories		
⊞ Bike Racks	237095.96	3166
⊞ Bike Stands	39591	249
⊞ Bottles and Cages	64272.87	10552
⊞ Cleaners	18406.71	3319
⊞ Fenders	46619.58	2121
⊞ Hydration Packs	105824.81	2761
⊞ Tires and Tubes	8231.9	3865
First Rate Bicycles		
⊞ Components	**22364.83**	**1015**
Jeff's Sporting Goods		

PivotTable Fields

☑ Total Amount
Date Dimension
□ DateHierarchy
Product Dimension
☑ ProductHierarchy
Sales Territory Dimension
□ TerritoryHierarchy
Vendor Dimension
☑ Vendor Id

Columns: Σ Values
Rows: Vendor Id, ProductHierarchy
Values: Total Amount, Quantity Sold

Conclusion:

In conclusion, undertaking the Adventure Works data warehouse project has been a valuable and enriching experience for practice and learning. Throughout the project, we have gained practical knowledge and hands-on experience in designing and implementing a data warehouse, as well as performing Extract, Transform, Load (ETL) processes.

Throughout the project, we have learned how to create and manipulate database tables, define primary keys and foreign key relationships, and optimize performance using indexes. We have also developed a deeper understanding of dimensional modeling principles and the importance of data consistency, integrity, and accuracy in a data warehouse environment.

By practicing ETL processes, including data extraction, transformation, and loading, we have gained hands-on experience in preparing and loading data into the data warehouse tables. This has allowed us to consolidate data from various sources, transform it into a consistent format, and load it into the appropriate dimensions and fact table.