Datawarehousing & ETL

Project Report

Table of Contents:

Introduction

1. **The Selected Dataset and Rationale for Its Choice**
    a. **Dataset Overview**
    b. **Rationale for Dataset Choice**
    c. **Initial Exploration and Filtering**
2. **Data Dictionary**
3. **The Star Schema**
    a. **Fact Table**
    b. **Dimension tables**
4. **Table Creation Process Using SQL Scripts**
    a. **Stages Tables (STA)**
    b. **ODS Tables**
    c. **Dimension and Fact Tables (DWH)**
    d. **Error logging table (ADM)**
    e. **Referential Integrity**
    f. **DimDate Special Case**
5. **ETL Process and Key Transformations**
    a. **ETL Architecture Overview**
    b. **SSIS Implementation Summary**
    c. **Key Transformations**
    d. **Error Handling, Logging, and Control Flow Design**
6. **Data Analysis & Insights**
    a. **Business Questions and Analytical Goals**
    b. **Analytical Queries and Results**

Conclusion

Introduction

In the modern era of data-driven decision-making, organizations rely heavily on structured data warehouses that support analytical processing and business intelligence. This report outlines the development of an end-to-end Extract, Transform, and Load (ETL) process and a structured data warehouse solution using SQL Server Integration Services (SSIS), SQL Server Management Studio (SSMS), and Visual Studio.

The purpose of this project is to guide the reader through a hands-on implementation of a traditional data warehouse architecture, starting from raw flat-file data to a refined star schema capable of supporting complex business queries. Each step in the pipeline is documented, and all transformations are meticulously explained. Insights generated from the warehouse data are included to demonstrate the analytical capabilities of the final solution.

The project is structured using the classic three-tier architecture of a data warehouse:

1. **Staging Area (STA)** – For raw data ingestion.
2. **Operational Data Store (ODS)** – For data cleansing and intermediate transformations.
3. **Data Warehouse (DWH)** – Final star schema design, optimized for querying.

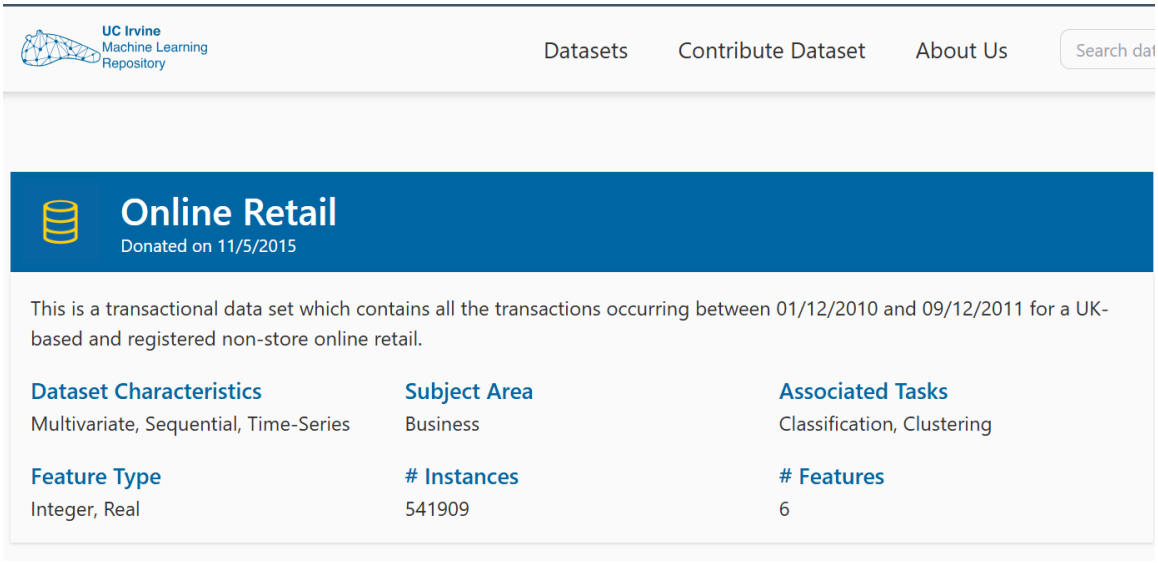
This report follows a detailed, structured format under the following sections:

- The selected dataset and rationale for its choice, along with a comprehensive data dictionary
- The star schema design, including fact and dimension tables
- ETL process and key transformations across STA, ODS, and DWH
- Insights and conclusions drawn from analysis

1. The Selected Dataset and Rationale for Its Choice

a. Dataset Overview

The dataset selected for this project is the **Online Retail Dataset** sourced from the UCI Machine Learning Repository and Kaggle. It consists of over 500,000 transactions made by customers of a UK-based online retailer between December 1, 2010 and December 9, 2011. Each transaction includes invoice information, product details, customer IDs, quantities, and pricing.



➔ Dataset Link

b. Rationale for Dataset Choice

This dataset was chosen for multiple reasons:

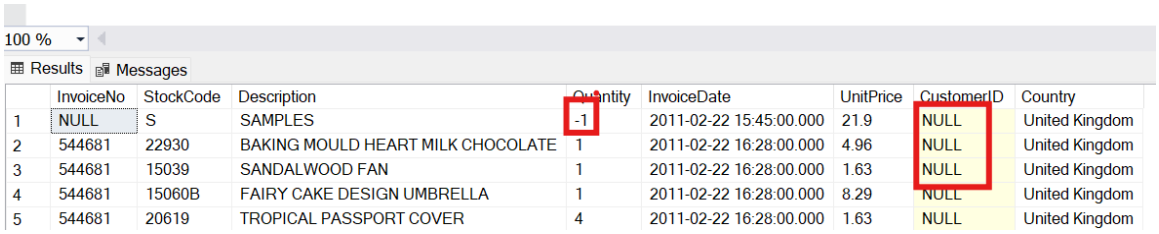
- It contains **transactional sales data**, making it an ideal candidate for a fact table.
- The dataset includes several **dimension-defining attributes** such as customers, products, countries, and dates.
- It allows for **time-based analysis**, customer behavior analysis, and product-level insights.
- It mimics a real-world retail scenario, which is both relevant and educational for ETL and data warehousing training.

c. Initial Exploration and Filtering

Before building the ETL pipeline, we conducted an exploration of the CSV file in Excel and SSMS. We identified key issues such as:

- Null values in CustomerID
- Duplicate transactions
- Negative or zero Quantity values
- Incomplete product descriptions

These insights informed the transformations and filtering logic we implemented later.



100 %

Results | Messages

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	NULL	S	SAMPLES	-1	2011-02-22 15:45:00.000	21.9	NULL	United Kingdom
2	544681	22930	BAKING MOULD HEART MILK CHOCOLATE	1	2011-02-22 16:28:00.000	4.96	NULL	United Kingdom
3	544681	15039	SANDALWOOD FAN	1	2011-02-22 16:28:00.000	1.63	NULL	United Kingdom
4	544681	15060B	FAIRY CAKE DESIGN UMBRELLA	1	2011-02-22 16:28:00.000	8.29	NULL	United Kingdom
5	544681	20619	TROPICAL PASSPORT COVER	4	2011-02-22 16:28:00.000	1.63	NULL	United Kingdom

2. Data Dictionary

The following table summarizes the most important columns extracted and used from the raw dataset:

Column name	Description	Data type
Invoiceno	Unique invoice number for each transaction	String
Stockcode	Unique product code	String
Description	Product description	String
Quantity	Quantity of product sold	Integer
Invoicedate	Timestamp of transaction	DateTime
Unitprice	Unit price of the product	Decimal
Customerid	Unique identifier for the customer	Integer
Country	Country of the customer	String

3. The Star Schema Design

The star schema is designed with a central Fact Sales table and four surrounding dimension tables:

a. Fact Table: FactSales

This table records each transaction and links to each dimension via surrogate keys.

Column	Description
SalesKey	Primary key (surrogate)
CustomerKey	Foreign key to dim_Customer
ProductKey	Foreign key to dim_Product
CountryKey	Foreign key to dim_Country
DateKey	Foreign key to dim_Date (YYYYMMDD)
Quantity	Number of items sold
UnitPrice	Price per unit
TotalAmount	Computed as Quantity * UnitPrice

b. Dimension Tables:

- **DimCustomer:** Stores unique customer identifiers.

Column	Description
CustomerKey	Surrogate primary key
CustomerID	Business identifier

- **DimProduct**: Stores unique products and their descriptions.

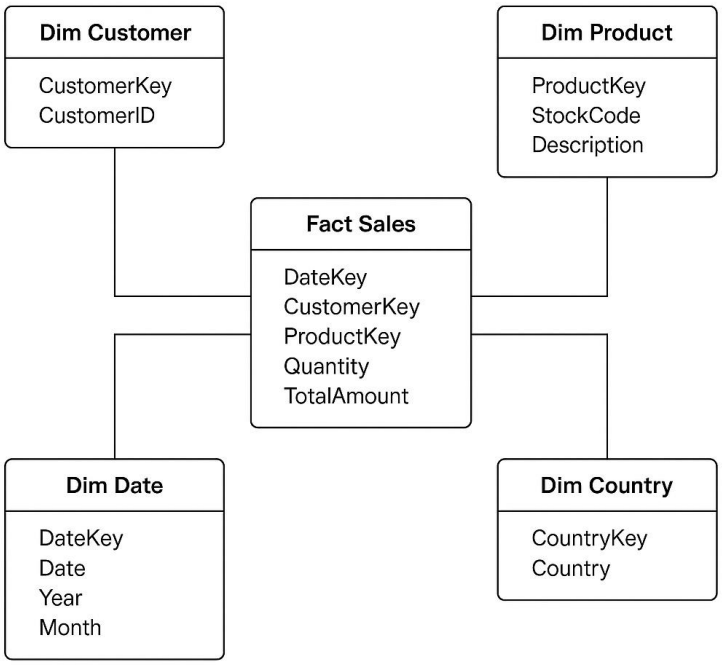
Column	Description
ProductKey	Surrogate primary key
StockCode	Business product code
Description	Product name

- **DimCountry:** Contains a unique list of countries.

Column	Description
CountryKey	Surrogate primary key
Country	Country name

- **DimDate:** Created using a T-SQL script. Allows filtering and analysis by date components.

Column	Description
DateKey	Format YYYYMMDD
FullDate	Actual date
Year	Year
Month	Month number
MonthName	Month name (e.g., January)
Quarter	Calendar quarter (1–4)



This star schema structure is designed to optimize query performance while maintaining simplicity and clarity for business users and analysts. The clear separation between fact and dimension tables ensures that business logic is easy to follow and scalable for future enhancements.

4. Table Creation Process Using SQL Scripts

To set up the data warehouse schema, we manually wrote and executed T-SQL scripts in **SQL Server Management Studio (SSMS)**. This process was structured in three main stages: staging (STA), operational data store (ODS), and data warehouse (DWH). Here's how we approached it:

a. Staging Tables (STA)

- The first step was to create a table to receive raw data directly from the source (CSV file).
- The staging table mirrored the structure of the input file to avoid conversion issues and simplify loading via SSIS.
- No constraints or transformations were applied at this stage; it served as a landing zone.

b. ODS Tables

- Once the raw data was in the staging table, we created an intermediate ODS table to perform cleansing and normalization.
- The table included filters to exclude invalid rows (null customers, negative quantities).
- At this point, we standardized data types and prepared the data to be compatible with dimension and fact tables.

c. Dimension and Fact Tables (DWH)

- We then created the dimension tables (DimCustomer, DimProduct, DimCountry, DimDate) with surrogate primary keys.
- These tables were designed to store unique, cleaned values and enable lookups during the ETL process.
- The fact table (FactSales) was built to store the core sales data, referencing all dimension tables through foreign keys.
- We added calculated columns such as TotalPrice to enrich the dataset and support business analytics.

d. Error Logging Table (ADM)

- An ErrorLog table was set up to **capture rows that failed during transformation or lookups** (unmatched keys, conversion errors).
- This table holds:
  - Original data row
  - Error description
  - Timestamp of failure
- It's used for **data quality review and debugging**.

e. Referential Integrity

- Constraints like primary keys and foreign keys were added to ensure data consistency.
- Indexes were created where necessary to optimize query performance.

f. DimDate Special Case

- Unlike the others, the DimDate table was generated entirely via a T-SQL script that populated dates across a defined range.
- It included extra columns like month names, quarters, and year for better time-based analysis.

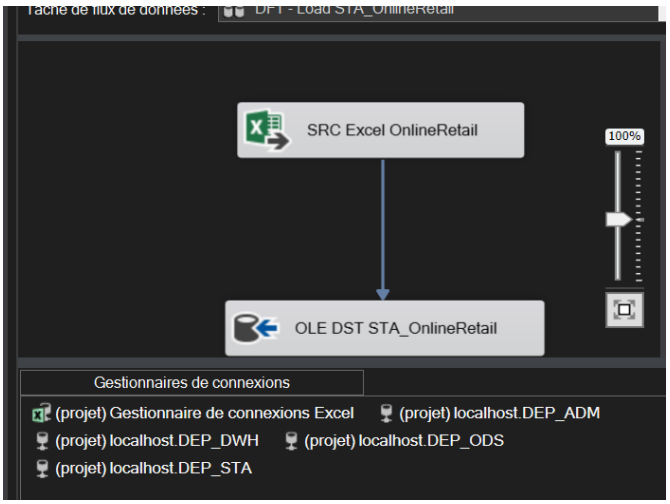
---

5. ETL Process and Key Transformations

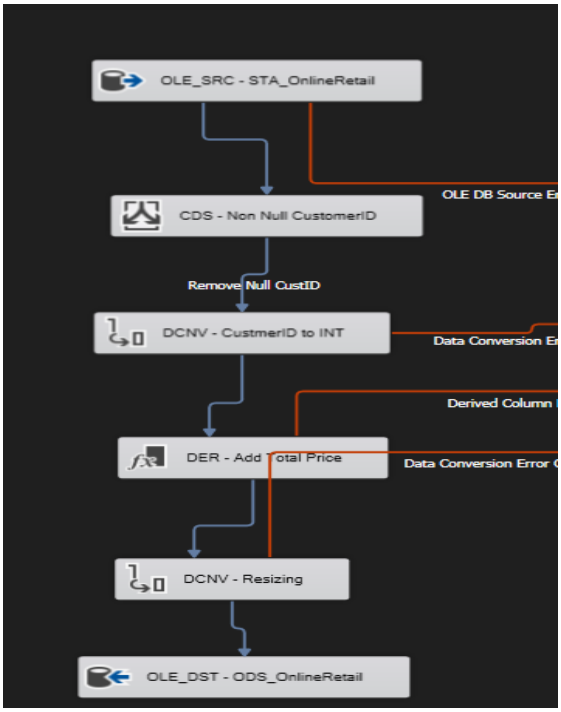
a. ETL Architecture Overview

The ETL process follows the classical 3-layered architecture:

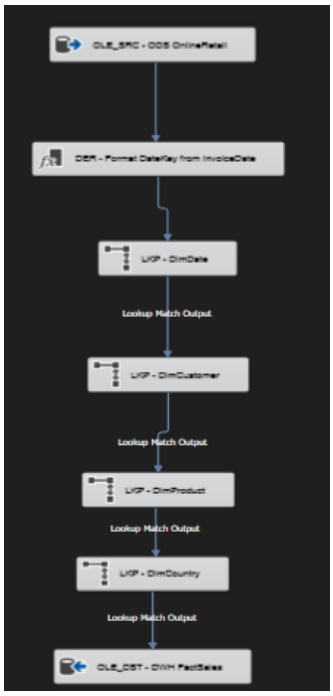
- **Staging (STA):** Raw flat file is imported using an SSIS Data Flow Task. Data is inserted into the STA_OnlineRetail table.



- **Operational Data Store (ODS):** Data cleansing and transformation occurs here. We filter invalid records, replace nulls, and standardize formats.



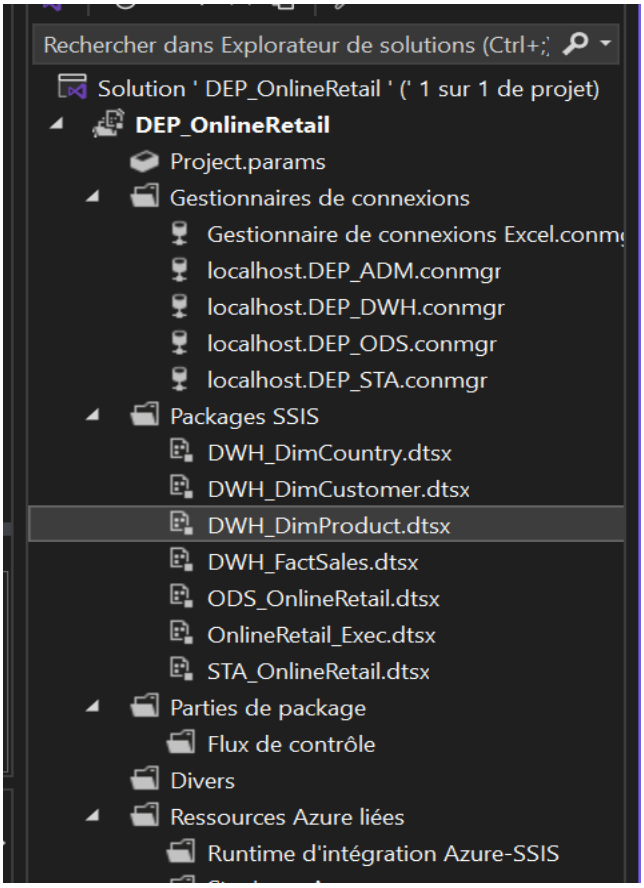
- **Data Warehouse (DWH):** Final transformation step involves creating surrogate keys using Lookups and populating fact and dimension tables.



b. SSIS Implementation Summary

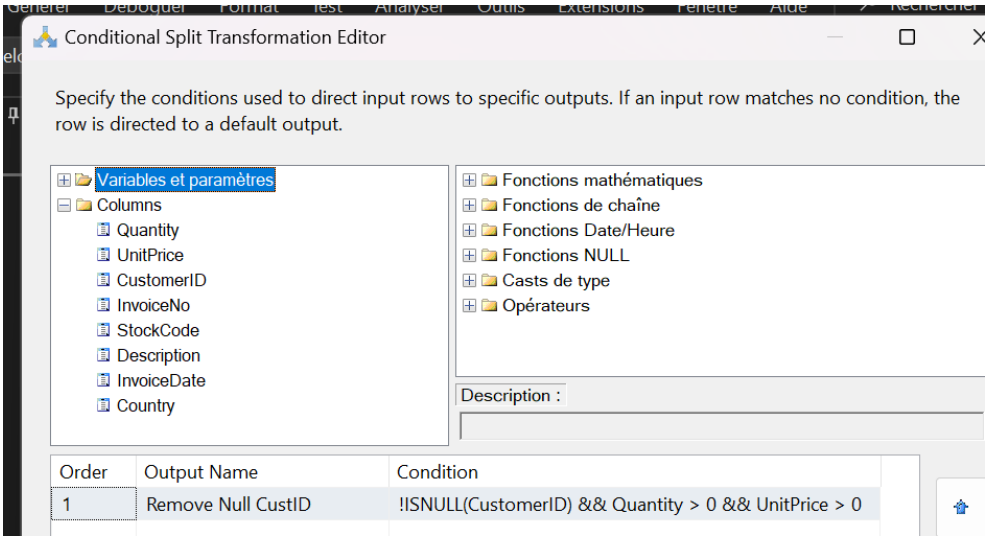
Each ETL stage is encapsulated in separate SSIS packages:

Package name	Purpose
Sta_Onlineretail	Load raw CSV to STA_OnlineRetail table
Ods_Onlineretail	Clean and transform for ODS
Dimcustomer	Load data into DimCustomer
Dimproduct	Load data into DimProduct
DimCountry	Load data into DimCountry
FactSales	Populate FactSales with lookups

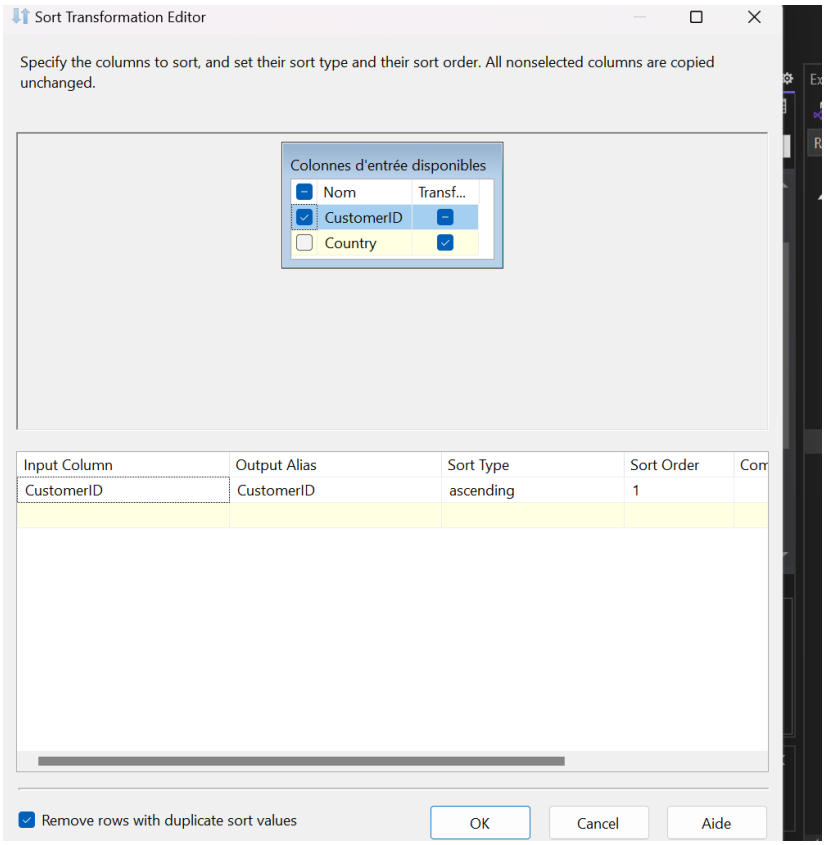


c. Key Transformations

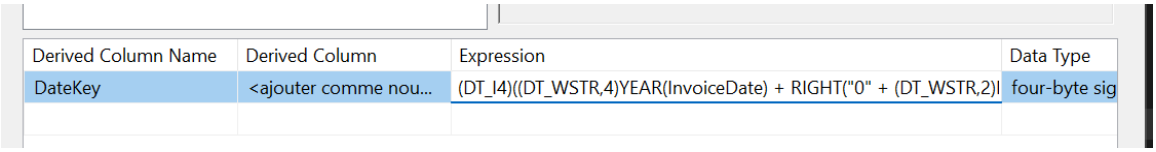
- **Filtering Rules:**
  - Remove rows with NULL CustomerID
  - Exclude transactions with negative or zero Quantity



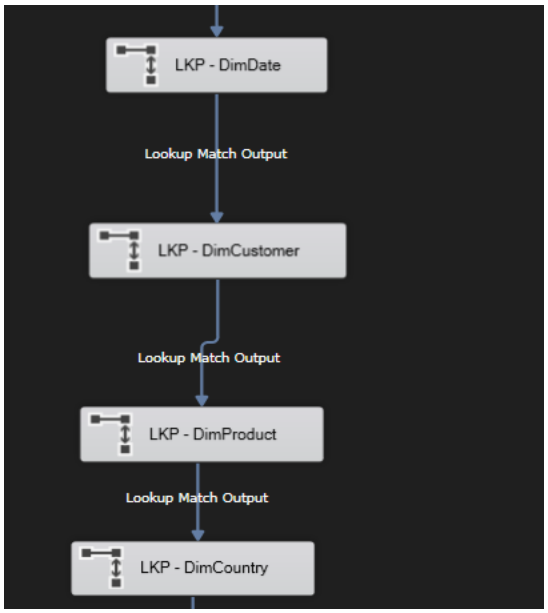
- Remove duplicated rows



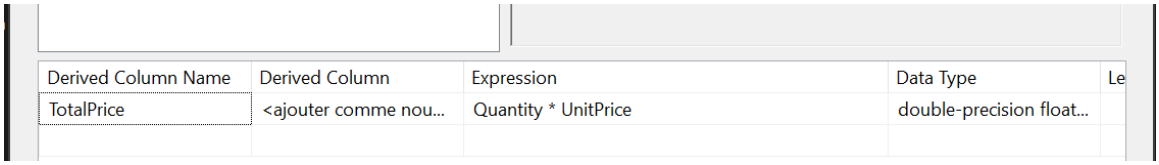
- **Date Formatting:**
  - Derived Column used to extract DateKey in format YYYYMMDD



- **Lookup Tasks:**
  - Used to retrieve surrogate keys from dimension tables based on business identifiers (CustomerID, StockCode, etc.)



- **Derived Columns:**
    - Calculate TotalPrice = Quantity * UnitPrice



Derived Column Name	Derived Column	Expression	Data Type	Le
TotalPrice	<ajouter comme nou...	Quantity * UnitPrice	double-precision float...	

- **Date Dimension Creation (DimDate):**
  Instead of loading DimDate using SSIS, we generated it entirely using a **T-SQL script**. The script programmatically created a calendar table that spans a defined date range (from 2010 to 2012). It includes important attributes such as DateKey (in YYYYMMDD format), FullDate, Year, Month, MonthName, and Quarter. This static date table ensures consistency in time-based analysis and supports efficient filtering, grouping, and trend identification.

```sql
DECLARE @StartDate DATE = '2010-01-01';
DECLARE @EndDate DATE = '2022-12-31';

WHILE @StartDate <= @EndDate
BEGIN
    INSERT INTO dbo.DimDate (
        DateKey,
        FullDate,
        Day,
        Month,
        MonthName,
        Quarter,
        Year,
        DayOfWeekName,
        IsWeekend
    )
    VALUES (
        CONVERT(INT, FORMAT(@StartDate, 'yyyyMMdd')),
        @StartDate,
        DAY(@StartDate),
        MONTH(@StartDate),
        DATENAME(MONTH, @StartDate),
        DATEPART(QUARTER, @StartDate),
        YEAR(@StartDate),
        DATENAME(WEEKDAY, @StartDate),
        CASE WHEN DATENAME(WEEKDAY, @StartDate) IN ('Saturday', 'Sunday') THEN 1 ELSE 0 END
    );

    SET @StartDate = DATEADD(DAY, 1, @StartDate);
END;
```

Results | Messages

	DateKey	FullDate	Day	Month	MonthName	Quarter	Year	DayOfWeekName	IsWeekend
1	20100101	2010-01-01	1	1	January	1	2010	Friday	0
2	20100102	2010-01-02	2	1	January	1	2010	Saturday	1
3	20100103	2010-01-03	3	1	January	1	2010	Sunday	1
4	20100104	2010-01-04	4	1	January	1	2010	Monday	0
5	20100105	2010-01-05	5	1	January	1	2010	Tuesday	0
6	20100106	2010-01-06	6	1	January	1	2010	Wednesday	0

d. Error Handling, Logging, and Control Flow Design

In an enterprise-grade ETL pipeline, robustness, monitoring, and traceability are essential. This ETL solution incorporates several strategies for error handling, logging, and modular control using SSIS features to ensure the pipeline performs reliably and can be debugged and audited effectively.

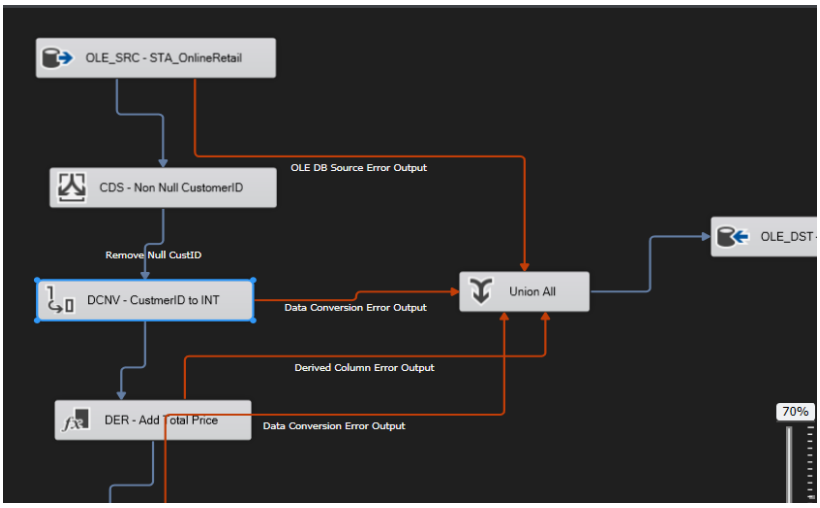
Error Handling Strategies

Each SSIS Data Flow Task includes built-in error paths and mechanisms to deal with data quality issues:

- Lookup Failures:

- In the FactSales package, if a CustomerID, StockCode, or Country fails to match a record in the corresponding dimension table, the row is redirected to an error output using a red arrow.
- Redirected rows are inserted into an ErrorLog table in the staging area for review and correction.

- Data Conversion Issues:
  - When converting values (e.g., Quantity, UnitPrice), the pipeline includes Data Conversion tasks.
  - Invalid or improperly formatted values are redirected, and errors are logged.



Data Quality Assurance

Before populating the ODS and DWH layers, several validation steps ensure the reliability of the data:

- Conditional Split:
  - Filters out:
    - Rows with NULL or missing CustomerID
    - Rows with Quantity <= 0
    - Transactions with invalid or empty product descriptions
  - Ensures only clean, relevant rows continue in the pipeline.

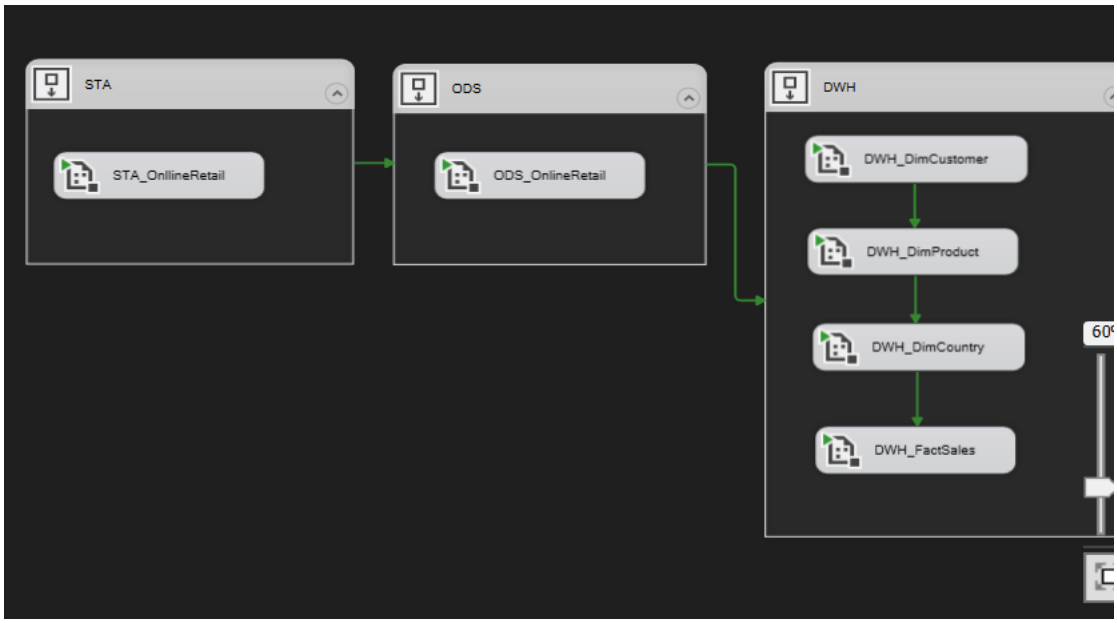
- Surrogate Key Integrity:

- Lookup transformations ensure that each fact record can be joined correctly to all related dimensions.
- If a lookup fails, the fact record is excluded from the final load to maintain referential integrity.

Control Flow and Package Orchestration

To manage the sequence and dependencies of package execution, a Master SSIS Package (0_OnlineRetail_Exec) is created using Sequence Containers and Execute Package Tasks.

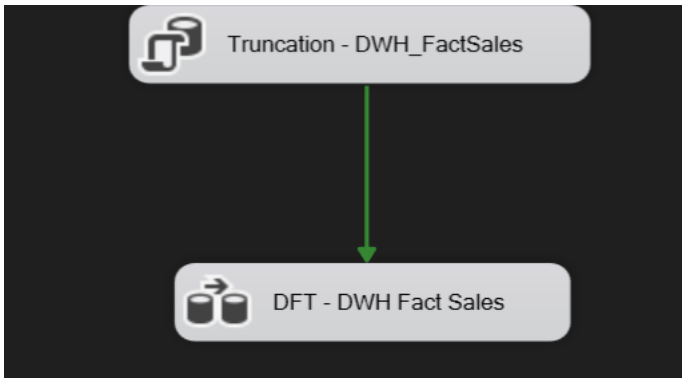
- Sequence Containers:
  - Group related tasks (STA Load, ODS Transformation, DWH Population) into logical units.
  - Useful for controlling rollback or retries at the group level if one step fails.
- Execute Package Tasks:
  - Each major sub-process is modularized into its own SSIS package.
  - Tasks are executed in order: STA → ODS → DWH using precedence constraints (green arrows for success).
- Error Flow Handling:
  - In case of task failure, failure paths direct to Log Error tasks for notification and alerting.



Pre-Execution Reset: Truncate Tables

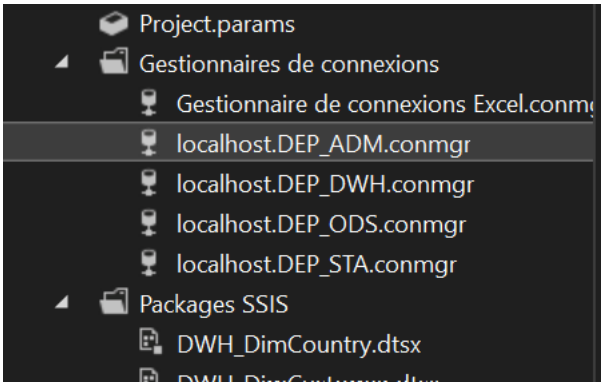
To ensure clean data loads and avoid duplicate entries:

- Before inserting data into ODS and DWH tables, Execute SQL Tasks are used to truncate destination tables.
- These scripts are embedded in the control flow at the start of each package.



Parameterization and Configuration

- Connection Managers are reused across packages and configured for flexibility.
- Project-level parameters hold common values such as file paths and database names.



Reusability and Scheduling

- The modular structure of the SSIS solution allows for reuse of packages with different datasets or business rules.

- This design is also compatible with SQL Server Agent Jobs for scheduled execution in production environments.

Data Transformation Summary: From 541,909 to 397,884 Rows

During the ETL (Extract, Transform, Load) phase, we applied several data cleansing and filtering steps to ensure that only valid, meaningful transactions were loaded into the data warehouse. The reduction from 541,909 raw rows to 397,884 clean rows was the result of the key transformations mentioned above.

---

6. Data Analysis & Insights

Once the ETL process was completed and the star schema successfully populated, we used SQL Server Management Studio (SSMS) to perform analytical queries on the data warehouse. These queries were designed to explore business questions related to sales trends, customer behavior, and product performance.

a. Business Questions and Analytical Goals

We structured the analysis around the following key questions:

- **Q1. What are the overall sales trends over time?**
- **Q2. Which countries contribute the most to revenue?**
- **Q3. Who are the top customers by total purchase amount?**
- **Q4. Which products are sold the most?**
- **Q5. What is the average order value (AOV) per customer, and how does it vary across countries?**

These queries provide critical insights for any retail business, offering clarity on customer value, geographic distribution of sales, inventory planning, and time-based sales strategies.

b. Analytical Queries and Results

Insight 1: Monthly Sales Trend

SQLQuery1.sql - GA...NG-WS\gallas (90))*

```
SELECT
    d.Year, d.Month, SUM(f.TotalAmount) AS MonthlySales
FROM
    dbo.FactSales f
JOIN
    dbo.DimDate d ON f.DateKey = d.DateKey
GROUP BY
    d.Year, d.Month
ORDER BY
    d.Year, d.Month;
```

100 %

Results | Messages

	Year	Month	MonthlySales
1	2010	12	572698.890000001
2	2011	1	569430.040000001
3	2011	2	447137.350000001
4	2011	3	595500.760000001
5	2011	4	469185.361000001
6	2011	5	678579.560000001
7	2011	6	661213.690000001
8	2011	7	600091.011000001
9	2011	8	645343.900000001
10	2011	9	952808.381000001
11	2011	10	1039288.79
12	2011	11	1161787.38
13	2011	12	518177.79

⊕ ***Key Observations:***

- ***Peak Sales:*** *November and October 2011 recorded the* ***highest revenue****, reaching over £1M each—likely driven by pre-holiday purchases and promotions.*
- ***Gradual Growth:*** *Sales steadily increased from early 2011, with a dramatic jump starting in* ***September****, suggesting strategic seasonal demand or marketing campaigns.*
- ***Business Cyclicality:*** *The trend confirms that the business has* ***strong Q4 performance****, which could be optimized further with targeted campaigns and inventory planning.*

Insight 2: Top 5 Countries by Revenue

SQLQuery1.sql - GA...NG-WS\gallas (90))*

```
SELECT
    c.Country, SUM(f.TotalAmount) AS TotalRevenue
FROM
    dbo.FactSales f
JOIN
    dbo.DimCountry c ON f.CountryKey = c.CountryKey
GROUP BY
    c.Country
ORDER BY
    TotalRevenue DESC
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;
```

100 %

Results | Messages

	Country	TotalRevenue
1	United Kingdom	7308226.55300348
2	Netherlands	285446.34
3	EIRE	265545.9
4	Germany	228867.14
5	France	209024.050000001

⊕ ***Key Insights:***

- ***UK Dominance:*** *The United Kingdom is the primary revenue generator, contributing more than* ***85%*** *of total sales. This is expected, as the business is based in the UK.*
- ***European Market Focus:*** *The remaining top countries—****Netherlands, EIRE, Germany, and France****—highlight a strong presence in Western Europe.*

Insight 3: Top 10 Customers

SQLQuery2.sql - GA...NG-WS\gallas (85)) SQLQuery1.sql - GA...NG-WS\gallas (90))*

```
SELECT
    cu.CustomerID, SUM(f.TotalAmount) AS CustomerTotal
FROM
    dbo.FactSales f
JOIN
    dbo.DimCustomer cu ON f.CustomerKey = cu.CustomerKey
GROUP BY
    cu.CustomerID
ORDER BY
    CustomerTotal DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;
```

100 %

Results Messages

	CustomerID	CustomerTotal
1	14646	280206.02
2	18102	259657.3
3	17450	194550.79
4	16446	168472.5
5	14911	143825.06
6	12415	124914.53
7	14156	117379.63
8	17511	91062.38
9	16029	81024.84
10	12346	77183.6

⊕ ***Key Insights:***

- ***Customer 14646*** *leads the chart with over* ***£280,000*** *in purchases, accounting for a significant share of total sales.*
- *The* ***top 3 customers*** *alone contribute more than* ***£730,000****, showcasing a heavy reliance on high-value buyers.*

Insight 4: Best-Selling Products

SQLQuery2.sql - GA...NG-WS\gallas (85)) SQLQuery1.sql - GA...NG-WS\gallas (90))*

```sql
SELECT
    p.Description, SUM(f.Quantity) AS TotalUnitsSold
FROM
    dbo.FactSales f
JOIN
    dbo.DimProduct p ON f.ProductKey = p.ProductKey
GROUP BY
    p.Description
ORDER BY
    TotalUnitsSold DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;
```

100 %

Results Messages

	Description	TotalUnitsSold
1	PAPER CRAFT , LITTLE BIRDIE	80995
2	MEDIUM CERAMIC TOP STORAGE JAR	77916
3	WORLD WAR 2 GLIDERS ASSTD DESIGNS	54415
4	JUMBO BAG RED RETROSPOT	46181
5	WHITE HANGING HEART T-LIGHT HOLDER	36725
6	ASSORTED COLOUR BIRD ORNAMENT	35362
7	PACK OF 72 RETROSPOT CAKE CASES	33693
8	POPCORN HOLDER	30931
9	RABBIT NIGHT LIGHT	27202
10	MINI PAINT SET VINTAGE	26076

⊕ ***Key Insights:***

- ***Paper craft items and storage jars*** *are leading in volume, suggesting high demand for decorative and functional household products.*
- *Seasonal or novelty items like* ***World War II gliders*** *and* ***Rabbit Night Lights*** *also perform strongly, hinting at gift-focused purchases.*
- ***Cake cases and retro-themed items*** *are consistently popular, supporting trends in home baking and vintage aesthetics.*

Insight 5: Seasonality and Sales Peaks

SQLQuery2.sql - GA...NG-WS\gallas (85)) SQLQuery1.sql - GA...NG-WS\gallas (90))*

```
SELECT
    c.Country,
    cu.CustomerID,
    SUM(f.TotalAmount) AS TotalSpent,
    COUNT(DISTINCT f.SalesKey) AS TotalOrders,
    ROUND(SUM(f.TotalAmount) * 1.0 / COUNT(DISTINCT f.SalesKey), 2) AS AverageOrderValue
FROM
    dbo.FactSales f
JOIN
    dbo.DimCustomer cu ON f.CustomerKey = cu.CustomerKey
JOIN
    dbo.DimCountry c ON f.CountryKey = c.CountryKey
GROUP BY
    c.Country, cu.CustomerID
ORDER BY
    AverageOrderValue DESC;
```

100 %

Results Messages

	Country	CustomerID	TotalSpent	TotalOrders	AverageOrderValue
1	United Kingdom	12346	77183.6	1	77183.6
2	United Kingdom	16446	168472.5	3	56157.5
3	United Kingdom	15098	39916.5	3	13305.5
4	United Kingdom	15749	44534.3	10	4453.43
5	United Kingdom	15195	3861	1	3861
6	United Kingdom	13135	3096	1	3096
7	United Kingdom	17846	2033.1	1	2033.1
8	United Kingdom	18087	4055.72	2	2027.86
9	United Kingdom	16532	6748.8	4	1687.2
10	United Kingdom	16000	12393.7	9	1377.08
11	United Kingdom	16754	2002.4	2	1001.2
12	Japan	12755	3811.95	4	952.99
13	United Kingdom	18133	931.5	1	931.5

Query executed successfully. GALLAS-NG-WS (15.0 RTM) | GALLAS-NG-WS\gallas (90) | DEP_DWH | 00:00:00 | 4,345 rows

⊕ ***Key Observations:***

- ***High AOV Outliers****:*
  - *Customers like 12346 and 16446 in the **United Kingdom** show extremely high AOVs due to making **fewer orders with large total values**.*
  - *For instance, Customer 12346 had only **one order worth £77,183.60**—an unusual case likely skewing the average.*
- ***Typical AOV Range****:*
  - *Many UK customers cluster around **£300–£600 AOV**, suggesting a general average order behavior.*
  - *Other countries like **Japan**, **Germany**, and **Canada** appear in the data but with **fewer customers and orders** so far.*

c. Key Conclusions

- The UK market is dominant; marketing efforts could be tailored for top-performing regions.
- High-value customers significantly impact overall revenue. Loyalty programs and segmentation would be beneficial.
- Product seasonality is evident, which supports calendar-based sales strategies.
- The dataset, though historical, provides actionable retail insights typical for medium-sized e-commerce operations.

Conclusion

This project aimed to build a scalable and insightful data warehouse solution for the **Online Retail dataset**, enabling deeper analysis of sales performance, customer behavior, and product trends. Using SQL Server Integration Services (SSIS), we successfully designed, transformed, and loaded data into a star schema model to support meaningful business intelligence reporting.

⊕ **Key Achievements:**

- **Data Modeling**:
  A star schema was created with a central **FactSales** table and connected **Dimension tables** for Customers, Products, Date, and Geography. This schema optimized query performance and analytical flexibility.
- **ETL Process**:
  Leveraging SSIS, we implemented a robust ETL pipeline that:
  - Cleaned and filtered raw transactional data.
  - Removed invalid or incomplete records.
  - Transformed columns (e.g., derived year/month from dates).
  - Handled surrogate keys and slowly changing dimensions (SCDs).
  - Populated the data warehouse in a structured, consistent manner.
- **Analytical Queries & Insights**: We performed multiple business-focused analyses, including:
  - **Monthly Sales Trend**: Identified seasonal peaks, with **November 2011** as the strongest month.
  - **Top Countries by Revenue**: The **UK** dominated sales, followed by **Netherlands** and **EIRE**.
  - **Top Customers**: High-value customers were identified for potential loyalty or CRM initiatives.
  - **Top-Selling Products**: Popular SKUs like *"Paper Craft, Little Birdie"* highlighted strong demand.

- **Average Order Value**: Calculated per customer, revealing purchasing power and segmentation insights.

⊕ **Business Value Delivered:**

- Enabled **data-driven decision making** through reliable sales KPIs.
- Improved **customer segmentation** potential for targeted marketing.
- Highlighted **product performance** to inform inventory or promotional strategies.
- Established a foundation for future integration with **reporting tools (Power BI)**.

This project demonstrates the value of **modern data warehousing techniques** in transforming raw data into actionable insights. It lays a strong groundwork for continuous analytics and strategic business improvements.