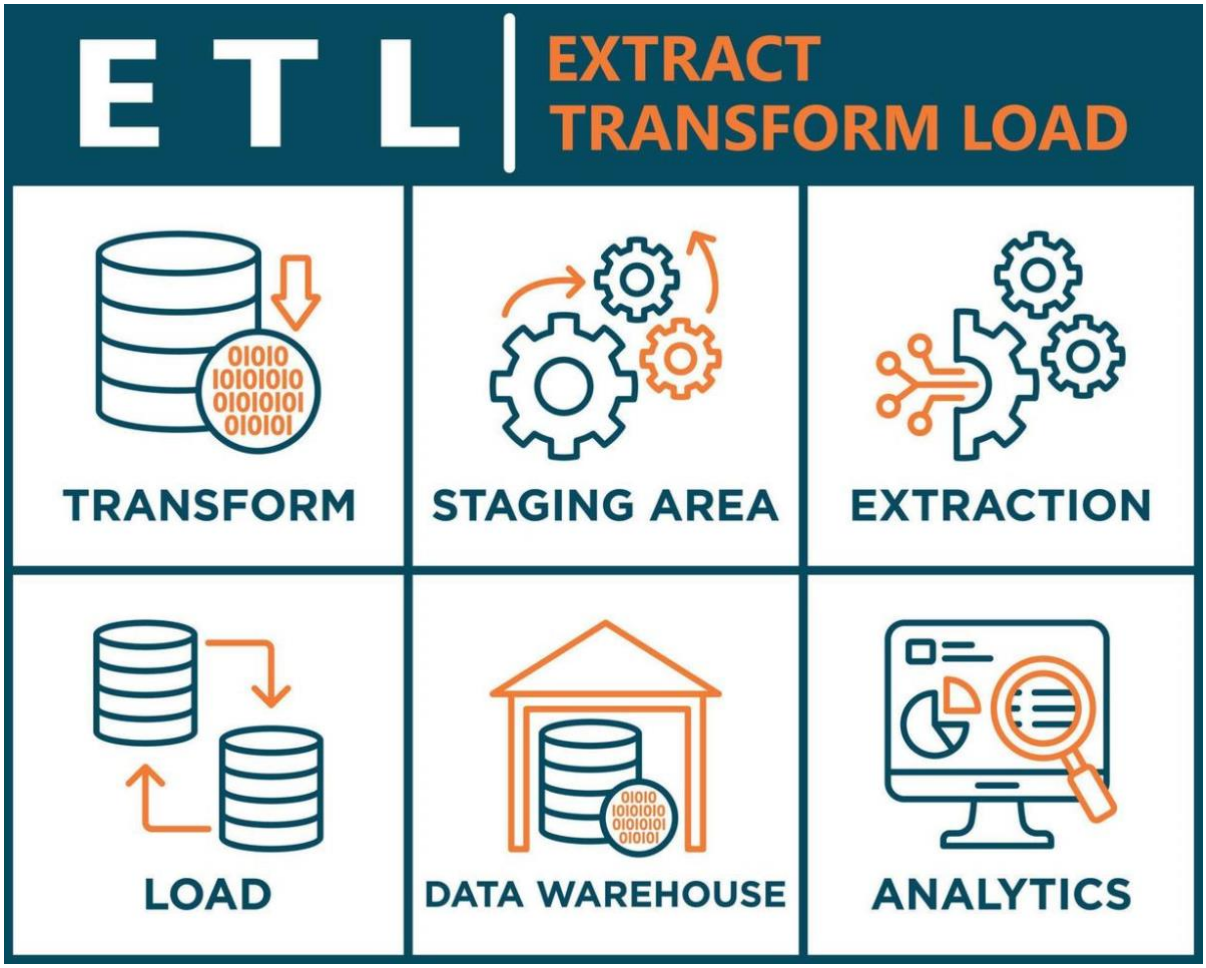
DATAWAREHOUSE AND ETL

Group #8 : Amazon Sales



Data ScienceTech Institute

Anthony Baudchon
Marie-Caroline Bertheau
Nans Long
Giti Shekari
Damien Rondet

Table des matières

Table des illustrations

Introduction

The primary objective of this project is to design and implement a comprehensive data warehouse solution that supports advanced business intelligence by integrating Amazon retail data through automated extract, transform, and load (ETL) workflows.

To achieve this, we selected a retail dataset from Amazon that includes 10 interrelated tables: customers, employee_territories, employees, order_details, orders, products, regions, shippers, suppliers, and territories. These tables range from 3 to 2,156 rows respectively for the smallest table, shippers table, and the largest one which is order_details. The dataset spans a historical range from July 4th, 1996, to May 6th, 1998, based on the first recorded order and last shipping date.

Initial data exploration revealed the dataset's structure, relationships, and potential insights. Based on this analysis, we designed a star schema designed for analytical queries and business intelligence reporting. The star schema consists of a central fact table, FACT_OrderDetails, and 6 surrounding dimension tables: DimCustomers, DimOrders, DimEmployees, DimDate, DimProduct, and DimSuppliers. This structure allows for efficient searching and simple navigation of business data.

The solution was built using the following Microsoft tools:

- SQL Server Management Studio (SSMS) 2019 v20.2:
  - Used for Database creation and query management
- Visual Studio 2022 v17.13.5:
  - Served as the development environment for building and deploying SSIS packages.
- SQL Server Integration Services (SSIS) 2022 v1.5:
  - Handled ETL workflows by extracting data from flat files and relational sources, transforming it, and loading it into the warehouse.
- Power BI v2.141.1253.0 (March 2025):
  - Used for data visualization, enabling the creation of interactive dashboards and reports.

These tools form an integrated pipeline, transforming raw transaction records into clean, structured warehouse data that is optimized for reporting and trend analysis.

To maintain clarity and consistency, we followed standard SSIS naming conventions based on the guidance from https://sqlkover.com/ssis-naming-conventions-2-0/ ,which helped organize package names and components throughout the ETL process.

In terms of database configuration, we chose the Latin1_General_100_CI_AS_SC_UTF8 set in SSMS to support special characters and multilingual text, especially those used in various European languages. During schema design, NVARCHAR was chosen for text fields where Unicode support was required, despite the fact that VARCHAR was more storage-efficient for non-Unicode data. This decision ensures that the warehouse can store and process multilingual content without data loss.

Additionally, to align with European reporting standards, the data warehouse uses region-specific formatting, such as a comma as a decimal separator (e.g. 1000,00), which increases compatibility with local tools and readability for regional users.

I. Staging (STA)

Staging is the first step of an ETL process, which consists in extracting raw data from the database fand stocking it as VARiable CHARactere (VARCHAR) in an OLE_DB Destinations table, to make data available for the operational data storage part (ODS). VARCHAR is a data type which allocates dynamic storage for variable length text optimizing the memory and performance of database. The staging is important because it ensures that we have all the data needed before starting to transform it.

To start this process, we established a connection SSIS tool - Visual studio, to ensure the information transfer between them. Then, in the Control Flow layout of Visual studio, we added an "Execute SQL Task" to truncate the tables in SSIS. This step is important to avoid duplicates when new data is added to our source. It can be referred to SQL – TRUNCATE AMAZON STA_Orders (Figures 1 and 2). Notice, we used Order as for example in this section to describe our staging process.

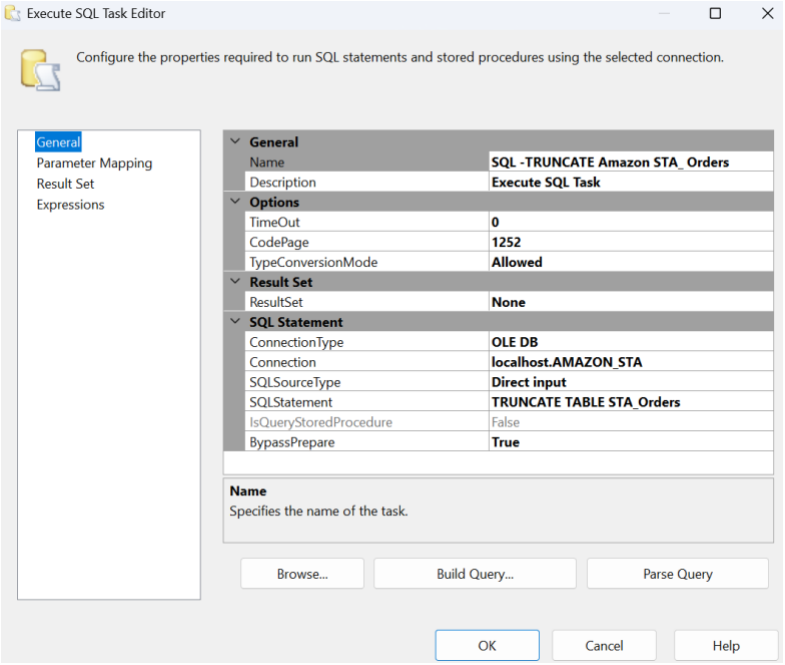
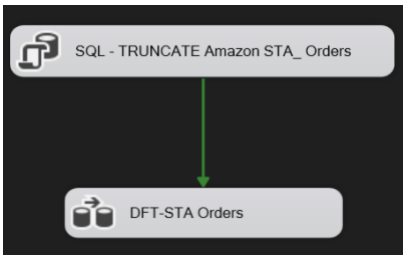


Figure 1: Truncate STA Tables



Once we secured the duplicates, we created the Data Flow Task block to start extracting and loading our raw data (Figure 2).

Figure 2: Control Flow for STA Packages

The goal of this step was to retrieve data from the source file (CSV format) and store it in an OLE_Destination after creating a Flat File Connection manager (Figure 3).

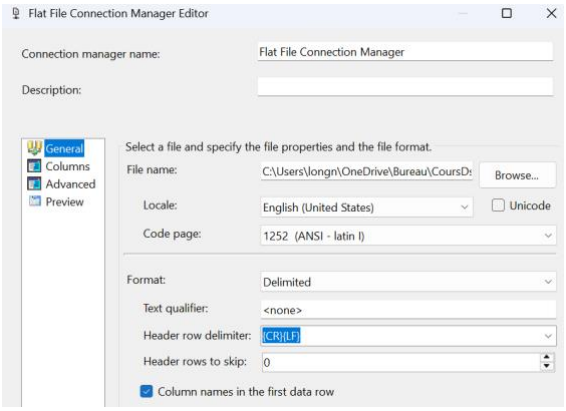


Figure 3:Extraction of the raw data from the source file (CSV format)

The previously collected data in the STA table (OLE_DST-STA Orders (Figure 4), after creating the receipt SQL table (STA Database). For this transfer, we have mapped the columns we extracted to their targets. Note that all the queries used to create the different tables are available in the ZIP file attached to our project report.

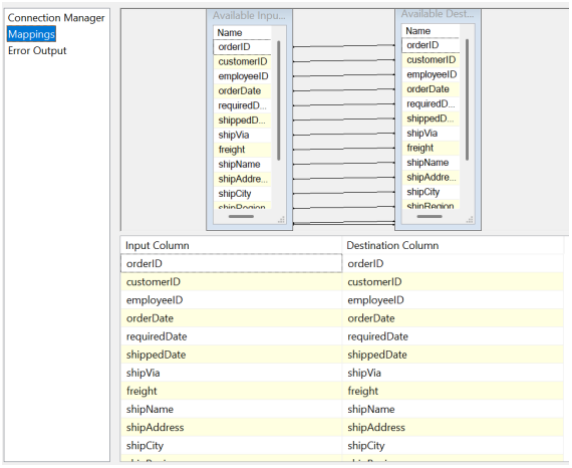


Figure 4:Mapping of the columns in OLE_DST – Orders

At the end of the staging, our Data Flow looks like the one presented in figure 5.

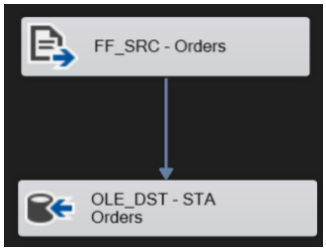


Figure 5:Data Flow Layout for the Orders Package

II. Transformation Part (ODS)

In the ETL processus, transformation phase is the process that involves converting, cleaning and enriching data before storage. This step ensures data quality, integrity and consistency while respecting business rules such as standardization (date format), validation (checking of numerical value) and integration/consolidation (table merge) of data. Each step will be presented in this section, using all the data stocked in tables from the previous part of staging.

A. ODS Customers & ODS Suppliers

To begin our transformation procedure, we began by resizing some tables as Customers (OLE_SRC-STA Customers) and Suppliers (OLE_SRC-STA Suppliers) by reducing the size of all informative strings from these two tables from 255 characters to 110 (or less) characters in function of features. These adequate dimensions restrictions constrain the engineer/DBA/developer to respect this rule to allow a best database management and avoid mistakes. Then, we performed these data conversions (DCNV) to produce new tables with appropriate string size (OLE_DST - ODS Customers or Suppliers) (Figure 6 and Cf. SQL formula table creation).

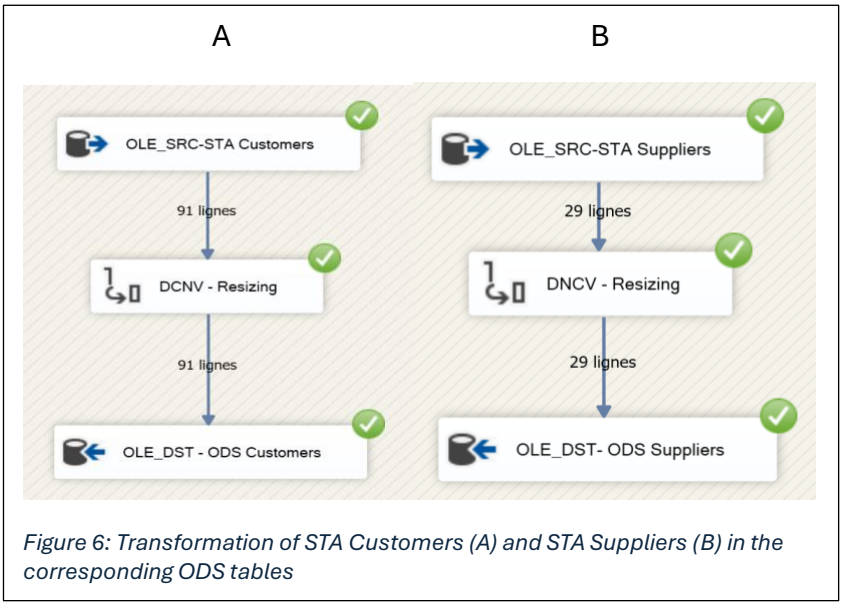


Figure 6: Transformation of STA Customers (A) and STA Suppliers (B) in the corresponding ODS tables

B. ADM TechnicalRejects

We converted (DCNV) the date from varchar to date time (DT_DATE) format. Then the errors outputs processed are transferred to new DER column rejects, confirming the code presented in Figure 7. All rejects were joined and stocked in OLE_DST – ADM TechnicalRejects, a table registering all the errors. We have set up the error output by redirecting the lines from DNCV - date process to DER - rejects step. After the validation of operations, the data are pushed in OLE_DST – ODS Employees. This document will have all administration information about amazon employees. Another essential data for CEO or team managers was to determine precisely the locations of intervention for each human resource to improve workflow and operations quality.

In fact, we implemented the information from the ODS_EmployeeTerritories into the ODS_Employees table to be able to have only one table grouping all the details about employees. To do this, we used a SQL query within a lookup function (LKP) matching employeeID from ODS_Employees to ODS_EmployeeTerritories as we see in the next chapters c and d. We noticed no errors from DCNV were detected.

Derived Column Name	Derived Column	Expression	Data Type	Length
RejectsDate	<add as new column>	GETDATE()	database timestamp ...	
RejectsPkgAndTask	<add as new column>	(DT_WSTR,50)@[System::PackageName] + (DT_WSTR,50)@[System::TaskName]	Unicode string [DT_...	100
RejectColumn	<add as new column>	"hireDate"	Unicode string [DT_...	8
RejectsDescription	<add as new column>	"The value " + (DT_WSTR,50)hireDate + " is not a valid date"	Unicode string [DT_...	80

Figure 7: Errors messages configuration for ADM TechnicalRejects table in the DER Transformation

C. ODS Employees territories

We merged the STA Territories to EmployeeTerritories by a LKP which selected TerritoryID and returned territoryDescription and regionID (Figure 8). This procedure is named LKP – Territory ID (Figure 9). We did the same execution to link STA Regions table on regionID by LKP which restituted regionDescription (LKP – Region Description). Then all the data were resized and sent to OLE_DST – ODS EmployeeTerritories (Figure 13).

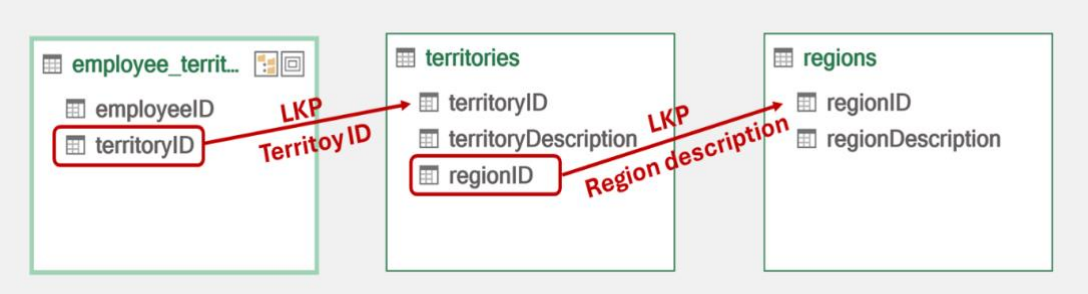


Figure 8: Schematic representation to merge territories and regions table to EmployeeTerritories

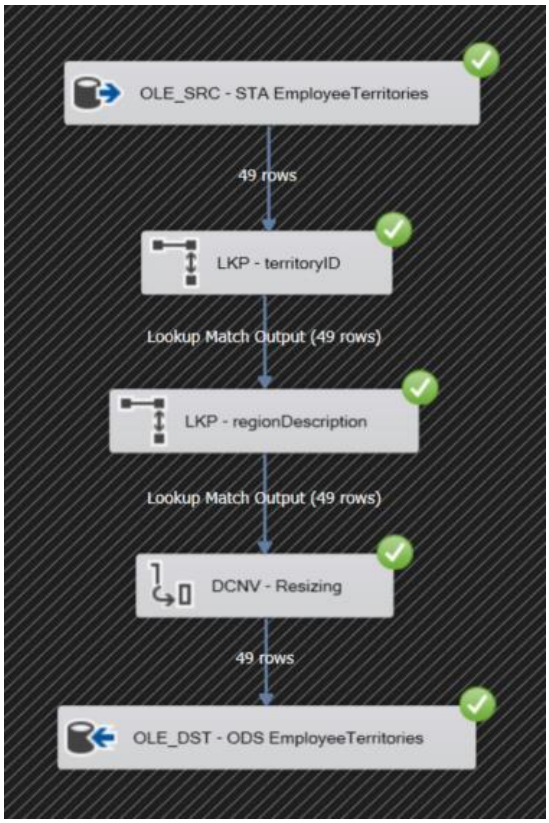


Figure 9: Adjustment of STA table EmployeeTerritories in ODS EmployeeTerritories

To get only one row in the dimension table, we aggregate the territory data with the corresponding employee row.

There is the procedure to get this final result in SSIS :

→ Use a Data Flow Taski in your SSIS package:
- Create a new Data Flow Task in the Control Flow.

→ Source Components :

- Add an OLE DB Source for ODS Employees.
- Add another OLE DB Source for ODS EmployeeTerritories.

→ Aggregate Territories (into string) per Employee:
- Do a Lookup Transformation to match each employee with their territory list.
- Pre-aggregate using SQL in the OLE DB connection manager of the lookup transformation (LKP – territoryList & regionDescription).

To visualise this new table, an SQL query was issued (Figure 10) and the output, named ODS_EmployeeTerritories, showed 11 rows (Figure 11), corresponding to the work area of each AMAZON employee from the database presented in this report.

Figure 10: SQL Code1

```
SELECT
    employeeID,
    STRING_AGG(territoryDescription, ', ') AS territoryList,
    regionDescription
FROM dbo.ODS_EmployeeTerritories
GROUP BY employeeID, regionDescription
ORDER BY employeeID
```

Then we get the figure 11 as result.

employeeID	territoryList	regionDescription
1	Wilton, Neward	Eastern
2	Westboro, Bedford, Georgetow, Boston, Cambridge, Braintree, Louisville	Eastern
3	Atlanta, Savannah, Orlando, Tampa	Southern
4	Rockville, Greensboro, Cary	Eastern
5	Providence, Morristown, Edison, NewYork, NewYork, Mellvile, Fairport	Eastern
6	Phoenix, Scottsdale, Bellevue, Redmond, Seattle	Western
7	HoffmanEstates, Chicago, Denver, ColoradoSprings, SantaMonica, MenloPark, SanFrancisco, Campbell, SantaClara, SantaCruz	Western
8	Philadelphia, Beachwood, Findlay, Racine	Northern
9	Hollis, Portsmouth, Southfield, Troy, BloomfieldHills, Roseville, Minneapolis	Northern

Figure 11: Result in SSIS

→ After, we select the 2 new columns that will be added to the new employees table.

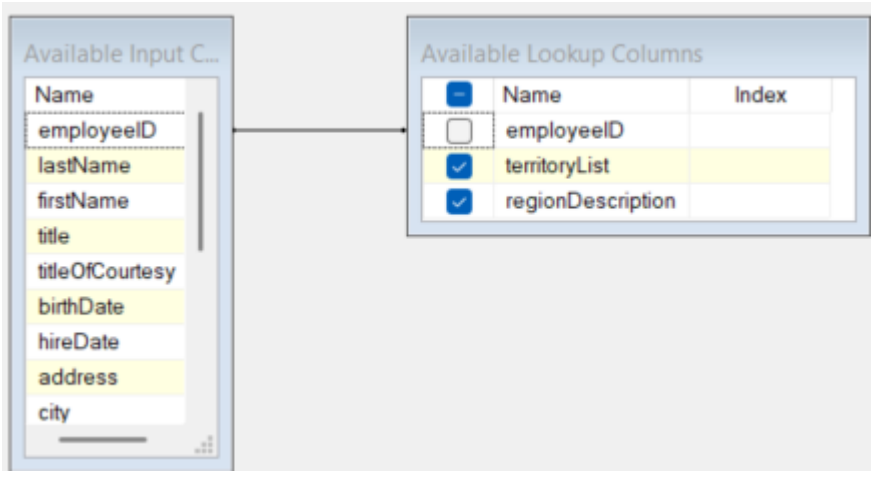


Figure 12 : LookUp manipulation

Once this is executed, the new ODS_Employees table dimension will still have one row per employee, so the DimEmployees relationship to FACT_OrderDetails can remain one-to-many, avoiding duplicates and incorrect aggregations.

→ Run the pipeline ODS_Employees :

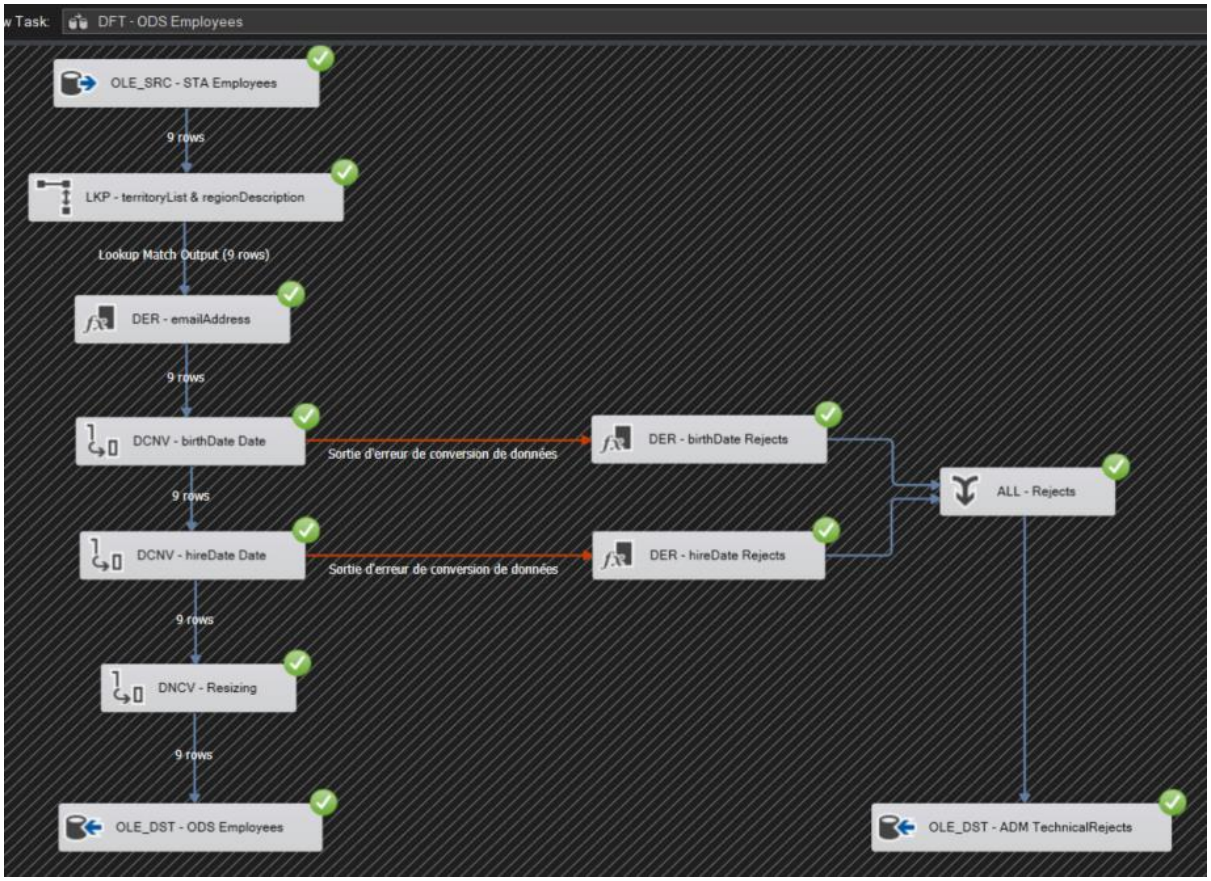


Figure 13: ODS-Employees Pipeline

→ Result in SSMS:

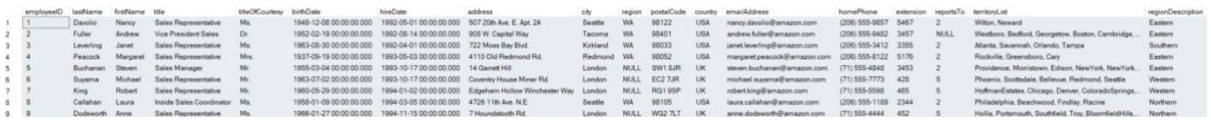


Figure 14: ODS-Employees in SSMS

All this procedure, let us to edit a DimEmployees table in the Datawarehouse and remove the DimEmployeeTerritories to obtain a star schema at the end.

D. ODS Employees

From STA employees table, some attributes were adujusted (Birthdate and Hire Date) and new mail adress column was created. This latter is the result from First name, dote and last name were merged and the domain ("@amazon.com") were added. This operation were did in one step in a derived column (DER), thanks to this following code :

= LOWER(firstName) + "." + LOWER(lastName) + "@amazon.com"

As a result, we can see the email addresses on SSMS in the ODS_Employees table below:

emailAddress
nancy.davolio@amazon.com
andrew.fuller@amazon.com
janet.leverling@amazon.com

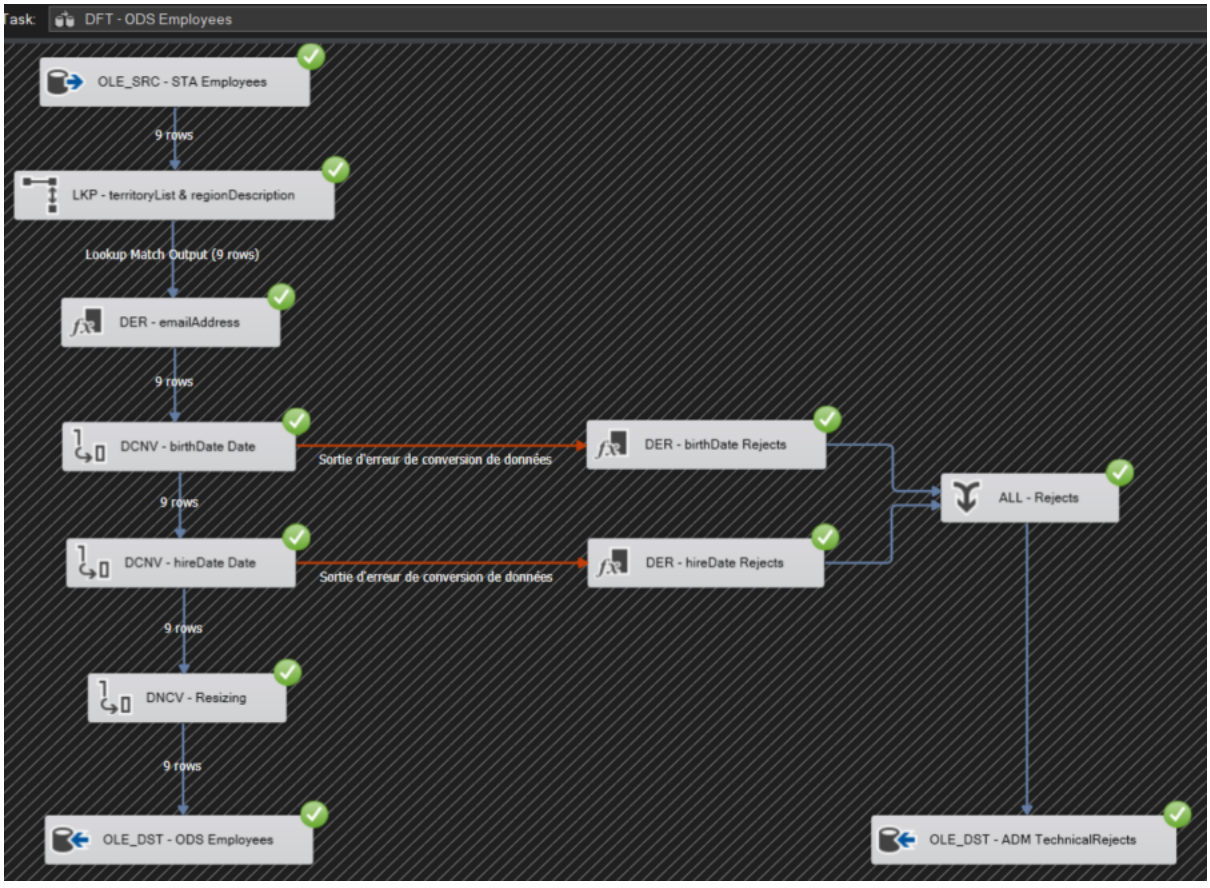


Figure 15 : Modification of STA Employees in an ODS table

E. ODS ORDERS

As previously, all date formats (Order Date, Required Date and ShipperDate) are modified in DCNV procedure from STA Orders. We noted 21 conversions (ShipperDate) were incompatibles and the corresponding error messages were loaded to ADM technical rejects (Figure 18) from code presented in Figure 7.

Then, the LKP function allowed to retrieve:

- Employee Name from STA Employee by a match on EmployeeID,
- Company Shipper Name (and phone information) from STA Shippers using as link the ShipperID,
- Company Contact Name from customerID thanks to STA Customers.

Then, we converted freight number varchar in numerical format:

- Step1: DER Freight Replace decimal symbol, with the formula : REPLACE(freight,",",".")

- Step 2: Converting varchar to DT_NUMERIC with a precision of 18 and a scale of 2 alias DCNV – Freight num, to have a float number of maximal length of 18 digits with 2 decimals)

As previously (Figure 7), we created a converting error part, and all the outputs are grouped in the in OLE_DST – ADM TechnicalRejects(Figure 17). When we activated this SSIS package, 21 rows of ShipperDates were declined(Figure 18).

Finally, we merge the STA Shippers with STA Orders in the purpose to get a star schema with the ODS Orders. To get more clearer, we delete the ShipID from ShipperTable, because the ID will not be used anymore. We also modify the shipper's value by "Transfer....." to avoid any confusion (Figure 16).

Figure 16: Renaming values in DimOrders

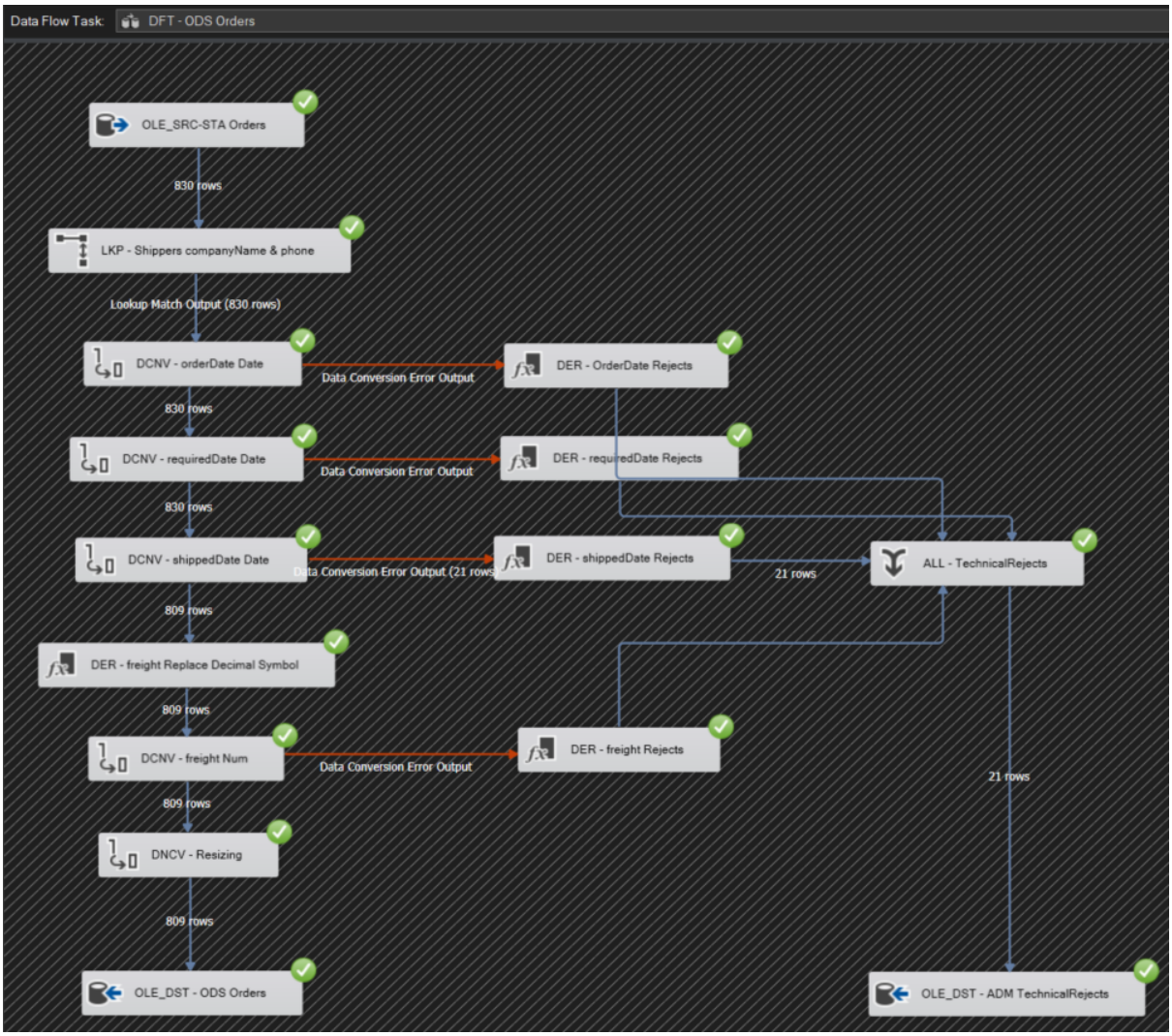


Figure 17: Transformation of STA Orders into ODS Orders

```
SELECT TOP (1000) [RejectsDate]
      ,[RejectsPkgAndTask]
      ,[RejectColumn]
      ,[RejectsDescription]
  FROM [AMAZON_ADM].[dbo].[TechnicalRejects]
```

100 %

Résultats Messages

	RejectsDate	RejectsPkgAndTask	RejectColumn	RejectsDescription
1	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
2	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
3	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
4	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
5	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
6	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
7	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
8	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
9	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
10	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
11	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
12	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
13	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
14	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
15	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
16	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
17	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
18	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
19	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
20	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date
21	2025-03-20 13:09:48.297	Package1DFT - ODS Orders	[shippedDate]	The value NULL is not a valid date

Figure 18: Technical rejects table with errors messages from the transformation of STA Orders in ODS

F. ODS Orders Details

To continue our exploration of the Order topic, we focused on Orderdetails (Figure 19), *i.e.* we've added the product name based on the product ID, and then we've adjusted the correct format of each number (UnitPrice and Discount) as explained below. We made an exception for quantity which is an integer noted int in the SSIS package by using DT_NUMERIC format with a precision of 18 and a scale of 0 alias DCNV – Quantity int (Figure 13). No rows were put in the reject technical table in our conditions.

In the last part of the transformation of our Amazon database, we modified the UnitPrice varchar in decimal format from STA Products. For the orders number, UnitsOnOrder, UnitsInStock, ReorderLevel and Discontined are converted in integers to have the ODS Products with no operation errors during the data process (Figure 19). Based on all the transformations carried out and presented in this section, these new tables were ready to be used in loading, the final step of our ETL project.

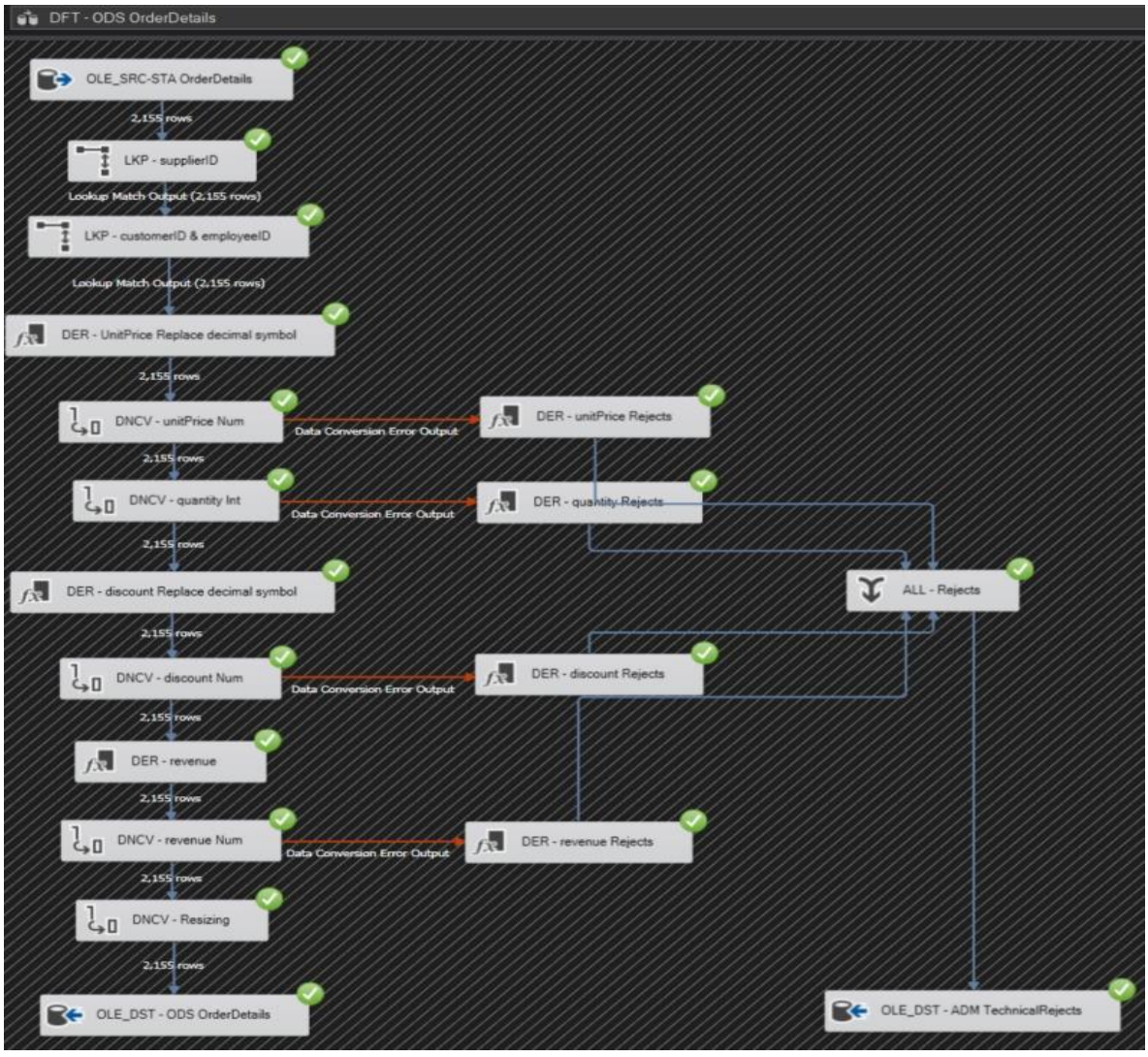


Figure 19:Modification process of STA OrderDetails in a ODS table

The aim of this approach was to store the previous data (ODS tables) in a target database while guaranteeing their integrity and performance. To be sure to get the right value of revenue in ODS-OrdersDetails, we use the formula below:

(DT_R8)unitPrice * (DT_I4)quantity - ((DT_R8)discount * (DT_R8)unitPrice * (DT_I4)quantity)

We use different data types in the formula :

- (DT_I4) for integer
- (DT_R8) for double

In our Dataset, we got two families of values called “unitPrice”, in two differents tables. We decided two keep them both because:

- In the product tables, it corresponding to the actual price of the product.
- In OrderDetails, it corresponded to another value at other moment (negociation with customer or ajustement of raw material market).

example: For producID = 2 in the OrderDetails table, we have the orderID 10258 with a unitPrice = 15.2. But the actual unitPrice in the products table is 19 and we can see that for the recent order 11030 in the OrderDetails table.

So, we decided to make it clearer in our Datawarehouse, to change the name of the unitPrice in ODS_OrderDetails and we name it “orderUnitPrice”.

G. ODS Products

For the ODS product, several transformations have been made to ensure data consistency and compliance with European standards. Firstly, the decimal symbol was replaced to conform to the European standard, a change previously detailed in the report. In addition, the UnitPrice field, originally a varchar in the Amazon database, was converted to a decimal format based on the structure of STA products. Several fields - OrdersNumber, UnitsOnOrder, UnitsInStock, ReorderLevel and Discontinued - were also converted to integer format in order to improve data processing and avoid operational errors. A general data conversion was applied to all numeric fields in the Products ODS table.

In addition, the Category ID column was removed as it only appeared in one table and was completely empty and irrelevant.

After all these transformations, the resulting tables were ready for the loading phase, the final step of our ETL project. The aim of this process was to store the transformed ODS tables in the target database while ensuring data integrity and optimal performance. (Figure 20)

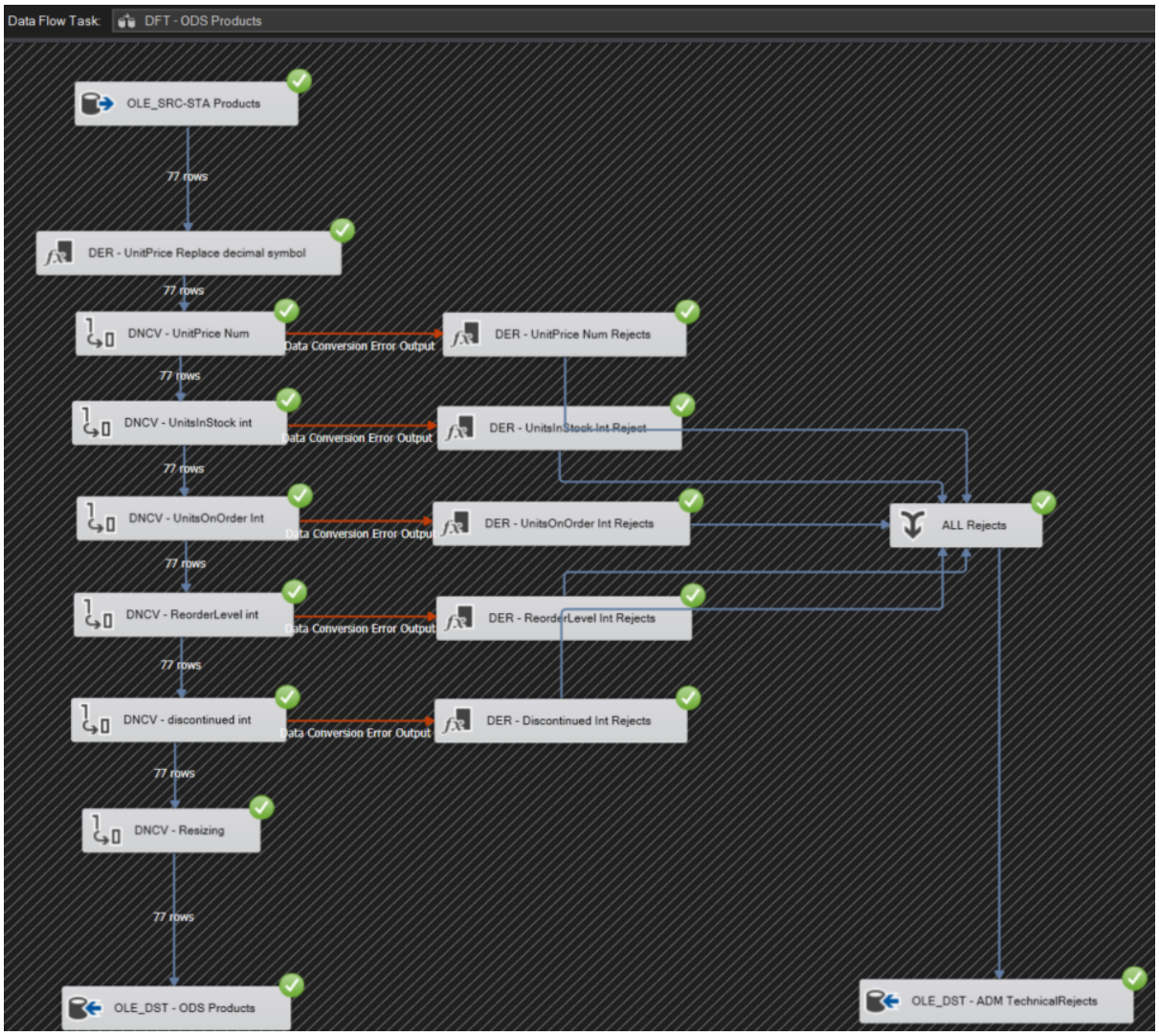


Figure 20:Transformation of STA Products in ODS tables corresponding

III. Load Part (DWH)

A. Organization

OrderDetails became the FACT table of the Datawarehouse.

- **FACT_orderDetails** aligned with the granularity of the data, which was at the **order line level** (each row represented a specific product within an order).

This table appears to be from the orderDetails table. The columns orderID, productID, unitPrice, quantity, and discount were characteristics of the orderDetails table, which stored information about individual products within an order.

Each row represented a product purchased in a specific order, detailing:

- orderID: The order identifier.
- productID: The product associated with that order.
- unitPrice: The price per unit of the product at the date of this order.
- quantity: The number of units ordered.
- discount: Any discount applied to that product in the order.

This orderDetails table was typically a junction table that linked orders (in orders table) and products (in products table) in our relational database. We also kept employee's territories informations without duplication (as we explain in the ODS part – Figure 21 and 22).

To prevent any duplication, we use this SQL code to update of DimEmployees for creating the SQL command in the CMD – update DimEmployees:

```
MySQL
UPDATE [dbo].[DimEmployees]
   SET [employeeID] = <employeeID, varchar(10),>
      ,[lastName] = <lastName, varchar(50),>
      ,[firstName] = <firstName, varchar(50),>
      ,[title] = <title, varchar(50),>
      ,[titleOfCourtesy] = <titleOfCourtesy, varchar(10),>
      ,[birthDate] = <birthDate, datetime,>
      ,[hireDate] = <hireDate, datetime,>
      ,[address] = <address, varchar(110),>
      ,[city] = <city, varchar(50),>
      ,[region] = <region, varchar(50),>
      ,[postalCode] = <postalCode, varchar(20),>
      ,[country] = <country, varchar(50),>
      ,[emailAddress] = <emailAddress, nvarchar(110),>
      ,[homePhone] = <homePhone, varchar(20),>
      ,[extension] = <extension, varchar(10),>
      ,[reportsTo] = <reportsTo, varchar(10),>
 WHERE <Search Conditions,,>
```

Figure 21: SQL Code 2

And we edited it to use this command:

```
UPDATE [dbo].[DimEmployees]
   SET [lastName] = ?
      ,[firstName] = ?
      ,[title] = ?
      ,[titleOfCourtesy] = ?
      ,[birthDate] = ?
      ,[hireDate] = ?
      ,[address] = ?
      ,[city] = ?
      ,[region] = ?
      ,[postalCode] = ?
      ,[country] = ?
      ,[emailAddress] = ?
      ,[homePhone] = ?
      ,[extension] = ?
      ,[reportsTo] = ?
 WHERE [employeeID] = ?
```

Figure 22: SQL Code 3

B. KEYS

About our Datawarehouse, we integrated surrogate keys into our 7 tables. This avoids repetition, stays intuitive, and aligns with common best practices. Naming surrogate keys consistently and clearly is important for maintaining a clean, understandable, and scalable data warehouse schema.

So, we chose to name our keys depending on the table names:

- orderDetailsKEY
- OrderKEY
- employeeKEY
- employeeTerritoriesKEY
- customerKEY
- productKEY
- SupplierKEY
- OrderDateKEY
- ShipperDateKEY
- requireddateKEY

About the DateKEY, we implemented them in DimOrders (DWH_orders), but we encountered a big issue. When we ran the pipeline, everything was running but the date return NULL values. The root of the problem was from the date type. It didn't match in DWH_Orders pipeline between the tables DimOrders and DimDate. We fixed the problem by editing all the date type from ODS to DWH and into the DimDate table too as "datetime".

C. Slowly Changing Dimension

1) Type 1

The SCD Type 1 has the specifity to delete the old value and remplace them by the newer values. It is very usefull when you don't need to have a historical of the values. The SCD takes place in three steps presented in the schema below:

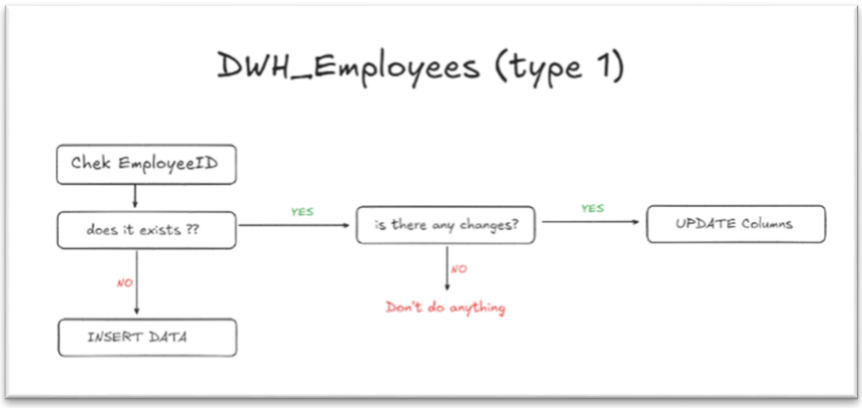


Figure 23: SCD type 1 operation

To do this, Two LKP were successively undertaken to update the columns (OLE DB Command noted CMD) (Figure 24).

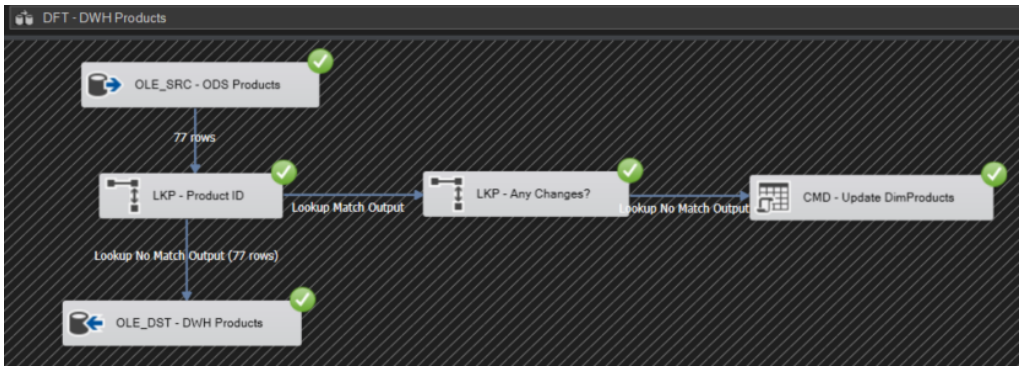


Figure 24: DWH Products

In our datawarehouse, we choose to put all the table in Type 1:

- *DimEmployees*
- *DimProducts*
- *DimSuppliers*
- *Fact OrderDetails*
- *DimOrders*

2) Type 2

For implementing SCD Type 2 (Slowly Changing Dimension Type 2) in our pipeline in Visual Studio using SSIS, the best table to apply it to depends on the business context and which entities we need to track historical changes for. But based on standard data warehousing practices, we choose to use a SCD type 2 to the DimCustomers table because of the following reasons (Figure 25):

- Customer data tends to change over time (address, contact info, etc.)
- It helps to keep history of these changes (e.g., track where a customer lived when they placed an order)
- This kind of historical accuracy matters in the business

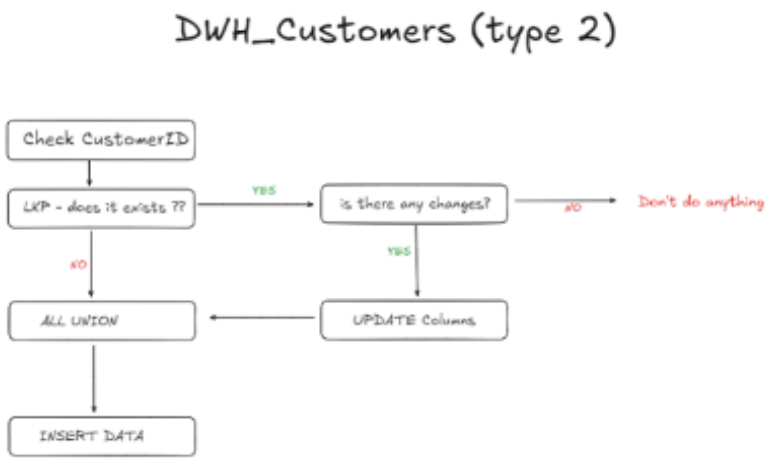


Figure 25: SCD Type 2 operation

With the type 2, we will be able to see the changes if a customer changes his details (name, phone, address...) and we will be able to keep track of the date in the DimCustomers table directly, when it changes. (Figure 26)

We decided to transform and update with records the following features:

- to keep track of the previous contact costumer Name: contactName
- to get the business costumer behaviour: address, city, region, postalCode, country, phone

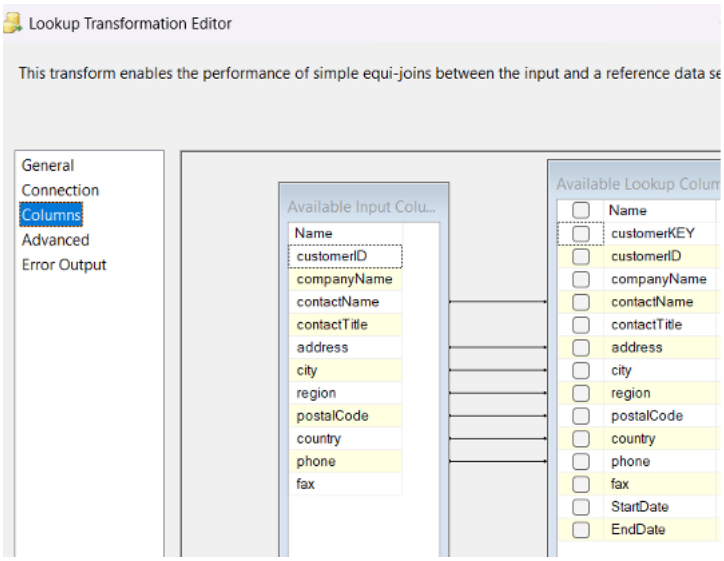


Figure 26: Lookup settings

We used an SQL query (Figure 27) for updating the date system in our SCD type 2 DimCustomers table. To do so, we wrote it in the CMD – Update DimCustomers to do the transformation. So the query matched the customerID on the parameter 0 as it is the "?" in the query.

```
UPDATE [dbo].[DimCustomers]
SET [EndDate] = GETDATE()
WHERE [customerID] = ? AND [EndDate] IS NULL
```

Figure 27: SQL Code 4

Then, we edit the first contactName (customerID = ALFKI) in the ODS_Customers to analyse the update in DimCustomers:

	customerKEY	customerID	companyName	contactName	contactTitle	address	city	region	postalCode	country	phone	fax	StartDate	EndDate
1	1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NULL	12209	Germany	030-0074321	030-0076545	2025-04-10 17:01:10.430	NULL
2	2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	NULL	05021	Mexico	(5) 555-4729	(5) 555-3745	2025-04-10 17:01:10.430	NULL
3	3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	NULL	05023	Mexico	(5) 555-3932	NULL	2025-04-10 17:01:10.430	NULL
4	4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	NULL	WA1 1DP	UK	(171) 555-7788	(171) 555-6750	2025-04-10 17:01:10.430	NULL

Figure 28: DimCustomers before updating

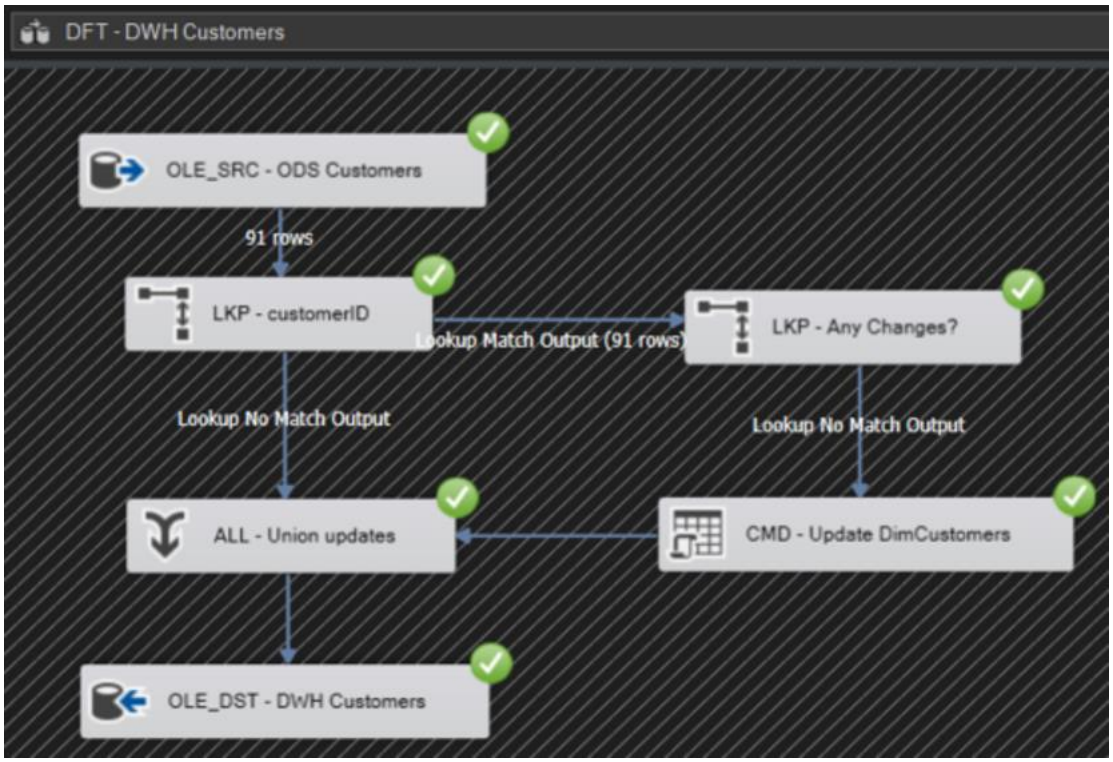


Figure 29:DWH_Customers pipeline after updating ODS_customers

The newly created DimCustomers table represented the new customerKEY by incrementing.(Figures 29, 30 & 31)

	customerKEY	customerID	companyName	contactName	StartDate	EndDate
1	1	ALFKI	Alfreds Futterkiste	Maria Anders	2025-04-10 17:01:10.430	2025-04-11 17:23:49.027
2	92	ALFKI	Alfreds Futterkiste	Sandra Anders	2025-04-10 17:23:49.507	NULL

Figure 30: DimCustomer indicating the changing time of contactName

	customerKEY	customerID	companyName	contactName	contactTitle	address	city	region	postalCode	country	phone	fax
1	1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NULL	12209	Germany	030-0074321	030-0076545
2	2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	NULL	05021	Mexico	(5) 555-4729	(5) 555-3745
3	3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	NULL	05023	Mexico	(5) 555-3932	NULL
4	4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	NULL	WA1 1DP	UK	(171) 555-7788	(171) 555-6750
5	5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	NULL	S-958 22	Sweden	0921-12 34 65	0921-12 34 67
6	6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	NULL	68306	Germany	0621-08460	0621-08924
7	7	BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager	24 place Kléber	Strasbourg	NULL	67000	France	88.60.15.31	88.60.15.32

Figure 31: DimCustomers with customerKEY

Finally, it could be interesting to use SCD type 2 for the DimEmployees, DimProducts and DimSuppliers table in case some of the business needs evolve in the future but in the case of our project it don't seems more useful than the SCD type 1.

D. Results

1) DWH_Orders

After the extraction of the data from ODS Orders, we did a lookup (LKP) on the OrderID. The goal is to determine if every orders already exist in DimOrders. If the command, already exists the SCD type 1 is applied. If the command does not exist yet, the Order ID follows the way to be added to DimOrders.

Then, we checked the date values (OrderDateKey, shippedDate, requiredDate). Using the multicast transformation (MLT), we can update in the same two flows, the principal and the rejects. With the derived column (DER) is used for generating or replacing the date KEY. Everytime, the values are not corresponding of the criteria difined before, they are redirected in the technical rejects.

The technical rejects from every steps are grouped (ALL) and aggregated (AGG). Finally, they are redirected to “FunctionalRejects” table.

All the corrects values are stored in the DWH Orders. To be able to reach the customer details for each order, we can use the relationships between DimOrders and the FACT_OrderDetails table, and then between the FACT table to the DimCustomers one. (Figure 32)

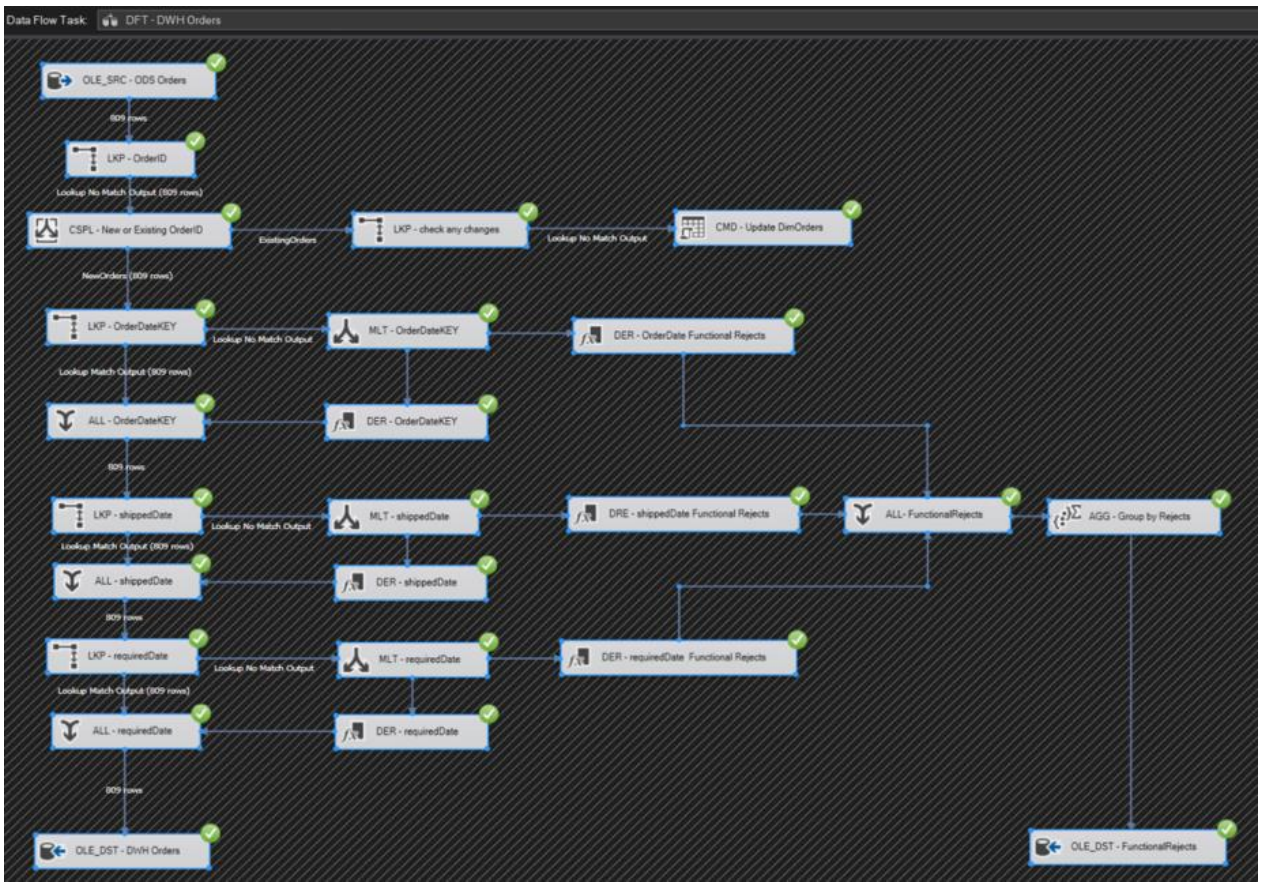


Figure 32: DWH Orders pipeline

To avoid redundancies, we removed the details of the customers in DimOrders because we can also find them in DimCustomers. Nevertheless, we should also remove them in ODS_Orders but we choose to keep them.

	orderKEY	orderID	orderDateKEY	requiredDateKEY	shippedDateKEY	freight	shipTransferName	shipTransferPhone
1	1	10248	19960704	19960716	19960716	32.38	Federal Shipping	(503) 555-9931
2	2	10249	19960705	19960710	19960710	11.61	Speedy Express	(503) 555-9831
3	3	10250	19960708	19960712	19960712	65.83	United Package	(503) 555-3199
4	4	10251	19960708	19960715	19960715	41.34	Speedy Express	(503) 555-9831
5	5	10252	19960709	19960711	19960711	51.30	United Package	(503) 555-3199
6	6	10253	19960710	19960716	19960716	58.17	United Package	(503) 555-3199
7	7	10254	19960711	19960723	19960723	22.98	United Package	(503) 555-3199
8	8	10255	19960712	19960715	19960715	148.33	Federal Shipping	(503) 555-9931
9	9	10256	19960715	19960717	19960717	13.97	United Package	(503) 555-3199
10	10	10257	19960716	19960722	19960722	81.91	Federal Shipping	(503) 555-9931

Figure 33: DimCustomers in SSMS

2) DWH_OrderDetails to Fact_OrderDetails

The purpose of the pipeline was to integrate all the data in the Fact Table OrdersDetails. As usual, we began with an extraction of the data from the ODS OrdersDetails.

Then we focused on the surrogates keys. For each key, we have done a lookup (LKP) to associate a dimension value with a key (OrderID, ProductID, SupplierID, CustomerID, EmployeeID)

If there is no match, the value goes in the technical rejects, to be analyzed later. The derived columns let us add or modify the key to the dataflow, to finally will be added to the Fact Table. (Figures 34 & 35)

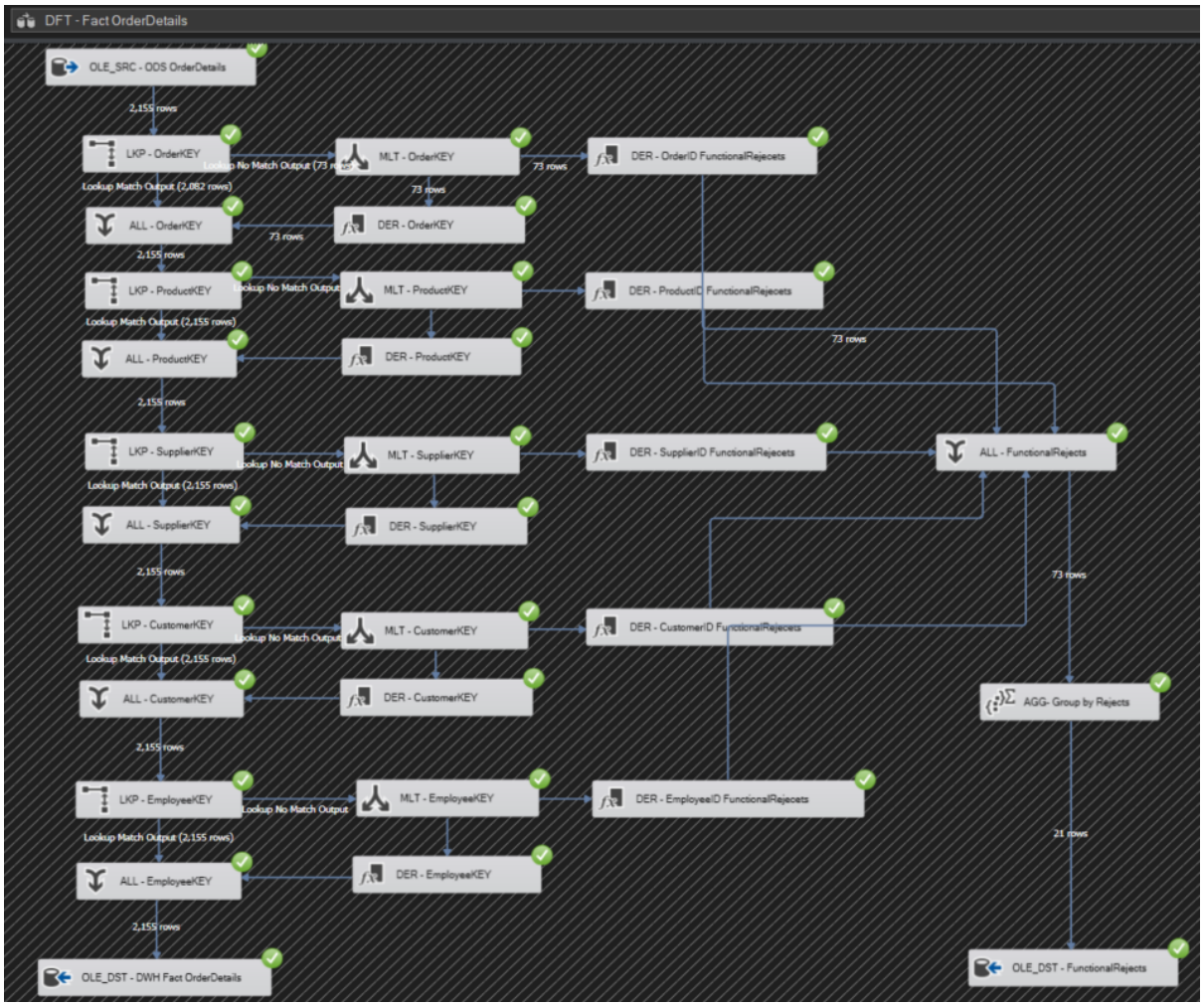


Figure 34: Fact OrderDetails

	orderDetailsKEY	orderKEY	productKEY	supplierKEY	customerKEY	employeeKEY	orderUnitPrice	quantity	discount	revenue
1	1	1	11	5	85	5	14.00	12	0.00	168.00
2	2	1	42	20	85	5	9.80	10	0.00	98.00
3	3	1	72	14	85	5	34.80	5	0.00	174.00
4	4	2	14	6	79	6	18.60	9	0.00	167.40
5	5	2	51	24	79	6	42.40	40	0.00	1696.00
6	6	3	41	19	34	4	7.70	10	0.00	77.00
7	7	3	51	24	34	4	42.40	35	0.15	1261.40
8	8	3	65	2	34	4	16.80	15	0.15	214.20
9	9	4	22	9	84	3	16.80	6	0.05	95.76
10	10	4	57	26	84	3	15.60	15	0.05	222.30
11	11	4	65	2	84	3	16.80	20	0.00	336.00

Figure 35: Fact OrderDetails in SSMS

IV. Issues

A. No Unicode type

While the project, we encountered a problem concerning SQL Server's support for special characters. We therefore configured our environment in UTF-8 to enable management of Unicode characters. To be able to face every king of characters, we used this collation :

Latin1_General_100_CI_AS_SC_UTF8

This configuration enabled us to maintain data integrity without the risk of data loss or additional transformations.

For example, the orderKEY = 5, the sender named is **“Suprêmes délices”**. With our modification of the encoding, we are able to to make it human readable. (Figure 36)

You can find below, the website’s link that help us to managed our issue:

[How to Change Your SQL Collation Settings]

	orderKEY	orderID	orderDateKEY	requiredDateKEY	shippedDateKEY	freight	shipSenderName	shipSenderAddress	shipSenderCity	shipSenderRegion	shipSenderPostalCode	shipSenderCountry	shipTransferName	shipTransferPhone
1	1	10248	19960704	19960716	19960716	32.38	Vins et alcools Chevalier	59 rue de l'Abbaye	Reims	NULL	51100	France	Federal Shipping	(503) 555-9931
2	2	10249	19960705	19960710	19960710	11.61	Toms Spezialitäten	Luisenstr. 48	Münster	NULL	44087	Germany	Speedy Express	(503) 555-9831
3	3	10250	19960708	19960712	19960712	65.83	Hanari Carnes	Rua do Paço 67	Rio de Janeiro	RJ	05454-876	Brazil	United Package	(503) 555-3199
4	4	10251	19960708	19960715	19960715	41.34	Victuailles en stock	2 rue du Commerce	Lyon	NULL	69004	France	Speedy Express	(503) 555-9831
5	5	10252	19960709	19960711	19960711	51.30	Suprêmes délices	Boulevard Tirou 255	Charleroi	NULL	B-6000	Belgium	United Package	(503) 555-3199
6	6	10253	19960710	19960716	19960716	58.17	Hanari Carnes	Rua do Paço 67	Rio de Janeiro	RJ	05454-876	Brazil	United Package	(503) 555-3199
7	7	10254	19960711	19960723	19960723	22.98	Chop-suey Chinese	Hauptstr. 31	Bern	NULL	3012	Switzerland	United Package	(503) 555-3199
8	8	10255	19960712	19960715	19960715	148.33	Richter Supermarkt	Starenweg 5	Genève	NULL	1204	Switzerland	Federal Shipping	(503) 555-9931
9	9	10256	19960715	19960717	19960717	13.97	Wellington Importadora	Rua do Mercado 12	Resende	SP	08737-363	Brazil	United Package	(503) 555-3199
10	10	10257	19960716	19960722	19960722	81.91	HILARION-Abastos	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	Venezuela	Federal Shipping	(503) 555-9931
11	11	10258	19960717	19960723	19960723	140.51	Ernst Handel	Kirchgasse 6	Graz	NULL	8010	Austria	Speedy Express	(503) 555-9831

Figure 36: ODS_Orders table (with UTF-8 collation)

B. DateTime

During final data integration in the DWH_Orders table, we encountered a major problem: several records contained NULL values in DateKEY columns (such as orderDateKEY, shippedDateKEY, requiredDateKEY). (Figure 37)

The first step in solving the problem was to standardize the date format throughout the process. To achieve this, we forced an explicit DATETIME cast on all date columns, including in the DimDate dimension table.

An analysis of the DWH_Orders pipeline revealed that records were being inserted with DateKEY set to NULL, indicating a conditional logic flaw in the handling of new versus existing orders.

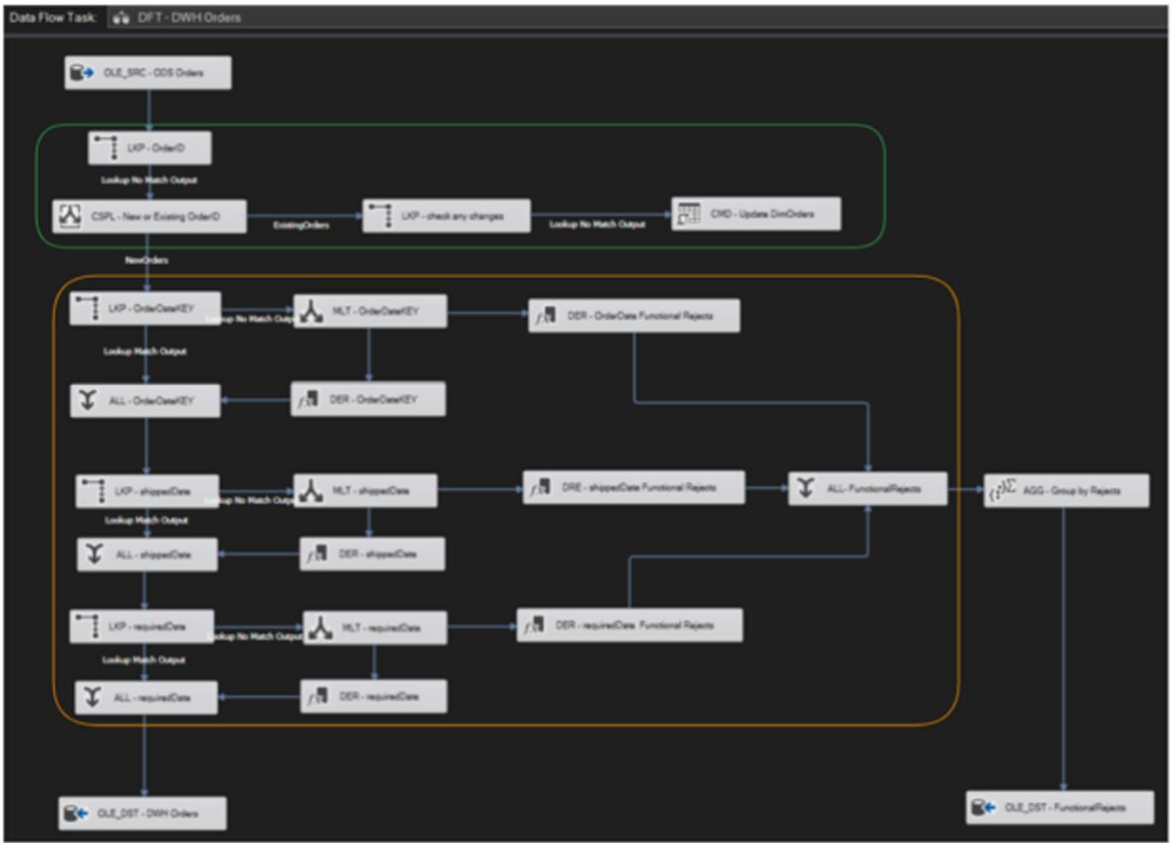


Figure 37: Schema of DWH Orders issue

To confirm this anomaly, we ran the following query in SQL Server :

```
SELECT COUNT(*)
FROM [AMAZON_DWH][dbo][DimOrders]
```

Figure 38: SQL Code 5

This query revealed a duplication of certain rows in the DimOrders table. To correct the behavior, we have integrated a “Conditional Split” transformation into the data stream.

This transformation differentiates between the following cases:

- If the OrderID already exists in the dimension: the row is updated with the new dates (orderDateKEY, shippedDateKEY, requiredDateKEY).
- If the OrderID is new: the record is inserted into the DimOrders table, by linking the corresponding date keys from the DimDate table.

Thanks to this conditional fork, we've eliminated redundant records and avoided inserting incomplete data.

Before reloading, we removed old, incorrect data containing NULL values in the date keys. To do this, we used the following SQL command:

```
TRUNCATE TABLE [AMAZON_DWH].[dbo].[DimOrders]
```

Figure 39: SQL Code 6

This enabled us to start from a clean base and guarantee data integrity after pipeline correction. (Figure 40)

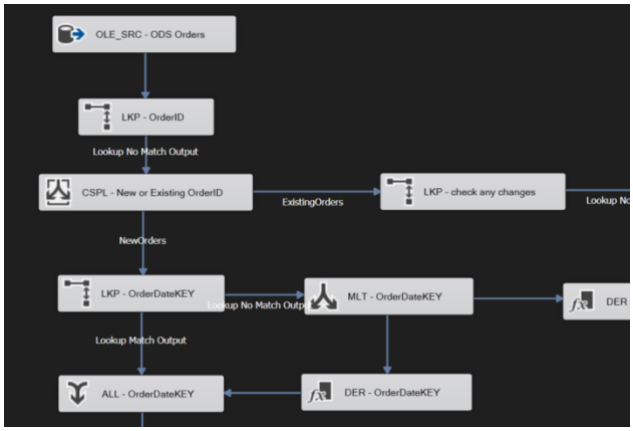


Figure 40: Conditional Split

C. Missing Values

While exploring the data in SSMS, we identified a data quality issue: 8 OrderID values (11070 to 11077) exist in the order_details table but are missing in the orders table. Acknowledging this inconsistency, we chose to proceed with the dataset while being mindful of potential impacts on certain analyses. These records can be excluded where necessary for accuracy.

We can see this gap with this SQL code:

```
USE  AMAZON_DWH

SELECT MAX([orderID])
FROM [AMAZON_ODS].[dbo].[ODS_Orders]

SELECT [orderID]
FROM [AMAZON_ODS].[dbo].[ODS_OrderDetails]
WHERE orderID > 11069
```

Figure 41: SQL Code 7

V. ETL

The ETL (Extract - Transform - Load) flow architecture was implemented for the progressive loading of data, from operational sources to the final data warehouse. This process follows a three-stage structure, divided into three zones: STA, ODS and DWH.

Before to run it, we truncate the tables in SSMS and then we start the execution

```
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Territories]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Suppliers]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Shippers]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Regions]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Products]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Employees]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Orders]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_OrderDetails]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_EmployeeTerritories]
TRUNCATE TABLE [AMAZON_STA].[dbo].[STA_Customers]

TRUNCATE TABLE [AMAZON_ODS].[dbo].[ODS_EmployeeTerritories]
TRUNCATE TABLE [AMAZON_ODS].[dbo].[ODS_Employees]
TRUNCATE TABLE [AMAZON_ODS].[dbo].[ODS_Products]
TRUNCATE TABLE [AMAZON_ODS].[dbo].[ODS_Customers]
TRUNCATE TABLE [AMAZON_ODS].[dbo].[ODS_Suppliers]
TRUNCATE TABLE [AMAZON_ODS].[dbo].[ODS_Orders]
TRUNCATE TABLE [AMAZON_ODS].[dbo].[ODS_OrderDetails]

TRUNCATE TABLE [AMAZON_DWH].[dbo].[DimEmployees]
TRUNCATE TABLE [AMAZON_DWH].[dbo].[DimProducts]
TRUNCATE TABLE [AMAZON_DWH].[dbo].[DimCustomers]
TRUNCATE TABLE [AMAZON_DWH].[dbo].[DimSuppliers]
TRUNCATE TABLE [AMAZON_DWH].[dbo].[DimOrders]
TRUNCATE TABLE [AMAZON_DWH].[dbo].[FACT_OrderDetails]
```

Figure 42: SQL Code 8

A. STA (Staging Area)

The STA zone represents the temporary staging area where raw data is initially deposited. It enables :

- Temporary storage of source data without modification.
- A clear separation between extraction and transformation, facilitating monitoring and the management of any anomalies.

The following tables are loaded into this zone:

Figure 43: Global STA

B. ODS (Operational Data Store)

The ODS zone is designed for operational data integration.

This is where data is :

- Cleansed
- Conformed to target models
- Potentially enriched or lightly transformed (standardization, typing, renaming, etc.).

ODS objects are named according to a logic parallel to the staging zone:

Figure 44: Global ODS

C. DWH (Data Warehouse)

The DWH zone is the final layer of the data warehouse, where data is ready for analysis. Here, data is loaded in a structured form, optimized for business intelligence queries.

DWH tables are designed according to a star schema and include :

Figure 45: Global DWH

The Data Flow Task at the center of the diagram orchestrates the passage of data from one level to another, ensuring sequential and logical processing:

From STA to ODS Then from ODS to DWH Each component with a green tick (✓) indicates that the steps have been successfully completed.

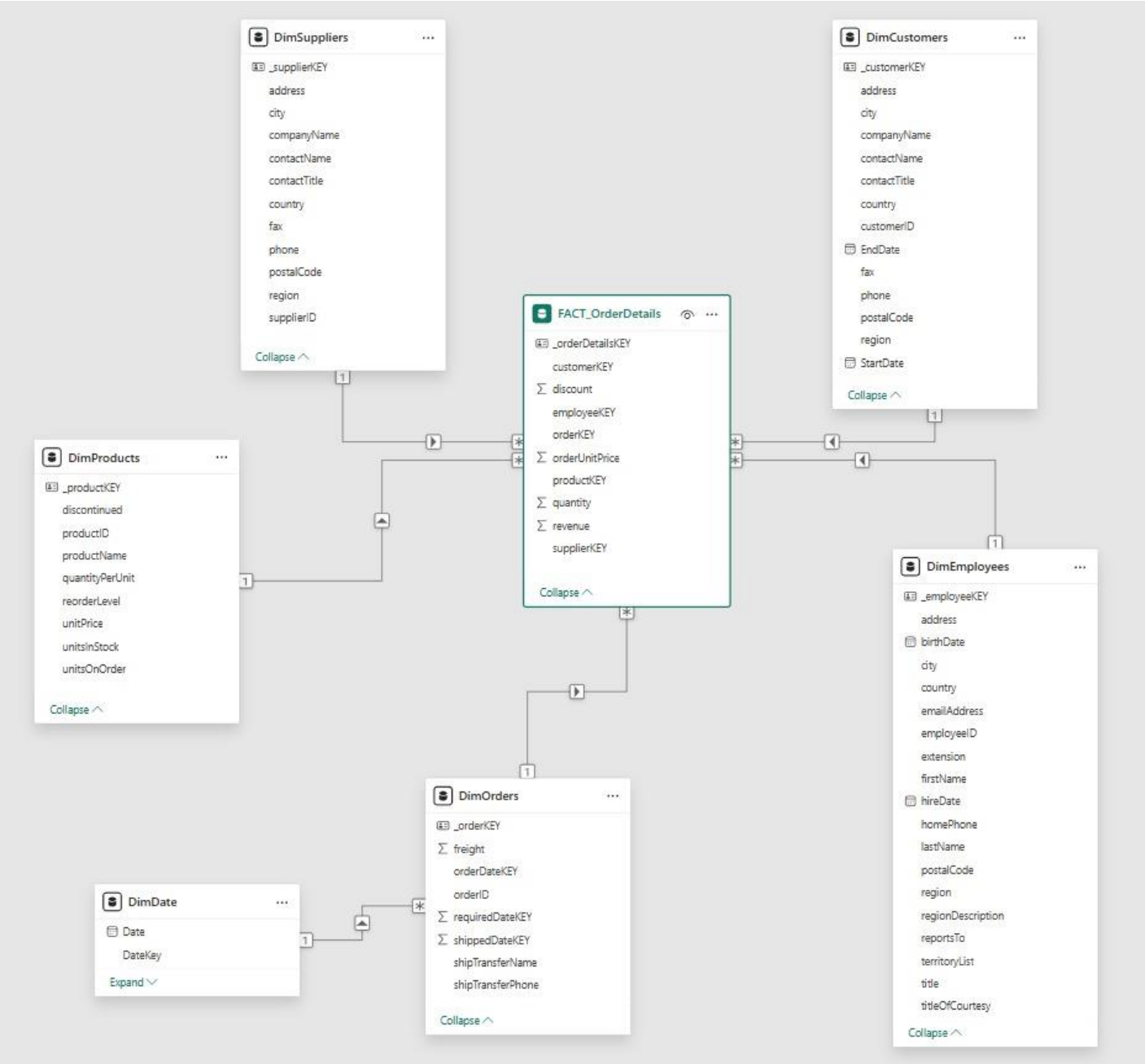
Figure 46: Complete ETL Process

VI. Star Schema

We finally get our star schema composed by:

- FACT_OrderDetails
- DimOrders
- DimDate
- DimEmployees
- DimCustomers
- DimProducts
- DimSuppliers

Figure 47: Star Schema



It was essential to design a star schema to ensure that our data warehouse is well structured, efficient and aligned with best practices in dimensional modelling. This approach enables clearer data organization by separating fact tables from dimensional tables, establishing proper relationships through primary and foreign keys, and ensuring data integrity and consistency. It also simplifies analytical queries and improves performance, making the data model more scalable and maintainable. (Figure 47)

VII. Data Analysis using PowerBI

A. Settings to connect to Power BI and summary of the Dashboard

First, we connect our SQL Server Database to PowerBI. To connect Power BI to SQL Server:

1. Open Power BI Desktop

2. Go to Home > Get Data > SQL Server

3. Enter:

   - Server name (e.g., localhost or myserver.database.windows.net)
   - Optionally, a Database name

4. Choose:

   - Import (data is loaded into Power BI model)
   - or DirectQuery (queries run live on your SQL Server)

5. Authenticate with your credentials

6. Load tables or write a custom SQL query if needed

Then, we can create the model of DWH as we saw in the previous chapter VI. Once we have our model in Power Bi, we can start with the analysis and create our dashboard.

The Sales report that we've created contains 8 Pages that gives an efficient overview of the employee performances, the top customers and their information, the inventory flow and products performances.

The objectives that were given for this analysis were as follows:

- What factors influence key performance metrics?
- How do different attributes (e.g., payment methods, customer location, product categories) impact business outcomes?
- What trends or patterns can be identified from the data?

The Dashboard presented below will try to answer those questions.

B. Dashboard Presentation

1) Employees

This page shows the Total revenues per employees and how much they made per quarter. You can click on an employee's name to see the turnover he made for each quarter.

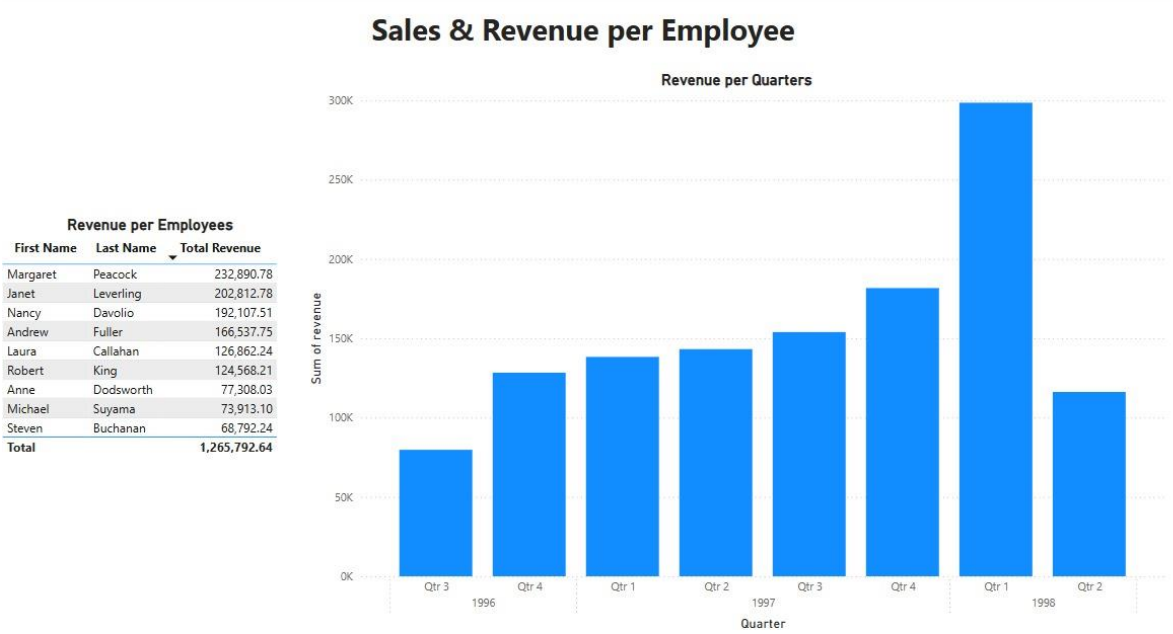
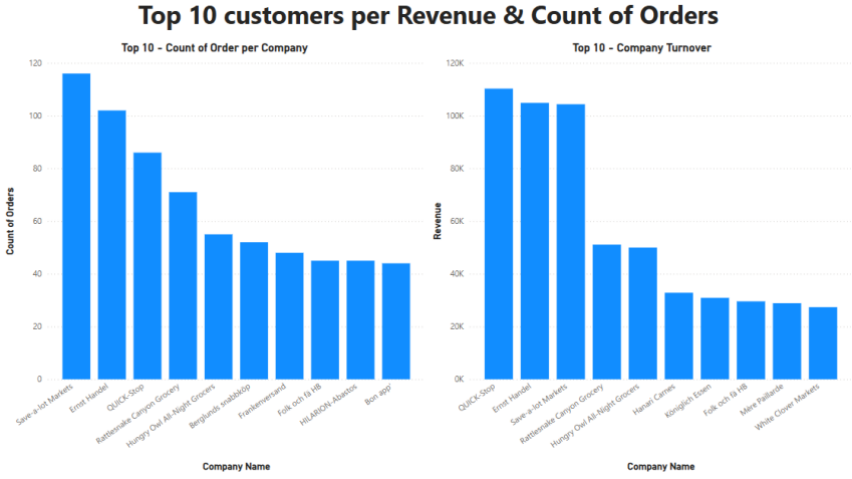


Figure 48: Sales and Revenue per Employee

2) Customers



This graph shows the Top 10 Customers' breakdown by the Count of Orders and the total revenue they made buying our products.

Figure 49: Top 10 customers per Revenue & Number of Orders

The purpose of this part was to know the customers who not ordered between July 1996 and May 1998. Then, we can decided to create a marketing operation to trigger orders from those customers.

Customers with no orders between July 1996 and May 1998

orderID	companyName	contactName	contactTitle	phone	address	city	postalCode	country
	Blauer See Delikatessen	Hanna Moos	Sales Representative	0621-08460	Forsterstr. 57	Mannheim	68306	Germany
	Bon app'	Laurence Lebihan	Owner	91.24.45.40	12 rue des Bouchers	Marseille	13008	France
	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	(604) 555-4729	23 Tsawassen Blvd.	Tsawassen	T2F 8M4	Canada
	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	(1) 135-5555	Cerrito 333	Buenos Aires	1010	Argentina
	Ernst Handel	Roland Mendel	Sales Manager	7675-3425	Kirchgasse 6	Graz	8010	Austria
	Great Lakes Food Market	Howard Snyder	Marketing Manager	(503) 555-7555	2732 Baker Blvd.	Eugene	97403	USA
	La maison d'Asie	Annette Roulet	Sales Manager	61.77.61.10	1 rue Alsace-Lorraine	Toulouse	31000	France
	Lehmanns Marktstand	Renate Messner	Sales Representative	069-0245984	Magazinweg 7	Frankfurt a.M.	60528	Germany
	LILA-Supermercado	Carlos González	Accounting Manager	(9) 331-6954	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto	3508	Venezuela
	LINO-Delicateses	Felipe Izquierdo	Owner	(8) 34-56-12	Ave. 5 de Mayo Porlamar	I. de Margarita	4980	Venezuela
	Pericles Comidas clásicas	Guillermo Fernández	Sales Representative	(5) 552-3745	Calle Dr. Jorge Cash 321	México D.F.	05033	Mexico
	Queen Cozinha	Lúcia Carvalho	Marketing Assistant	(11) 555-1189	Alameda dos Canàrios 891	Sao Paulo	05487-020	Brazil
	Rancho grande	Sergio Gutiérrez	Sales Representative	(1) 123-5555	Av. del Libertador 900	Buenos Aires	1010	Argentina
	Rattlesnake Canyon Grocery	Paula Wilson	Assistant Sales Representative	(505) 555-5939	2817 Milton Dr.	Albuquerque	87110	USA
	Reggiani Caseifici	Maurizio Moroni	Sales Associate	0522-556721	Strada Provinciale 124	Reggio Emilia	42100	Italy
	Ricardo Adocicados	Janete Limeira	Assistant Sales Agent	(21) 555-3412	Av. Copacabana 267	Rio de Janeiro	02389-890	Brazil
	Richter Supermarkt	Michael Holz	Sales Manager	0897-034214	Grenzacherweg 237	Genève	1203	Switzerland

country and country

country Argentina Austria Belgium Brazil Canada Denmark Finland France Germany Ireland Italy Mexico Norway Poland Portugal Spain Sweden Switzerland UK USA

Figure 50: Customers without order between July 1996 and May 1998

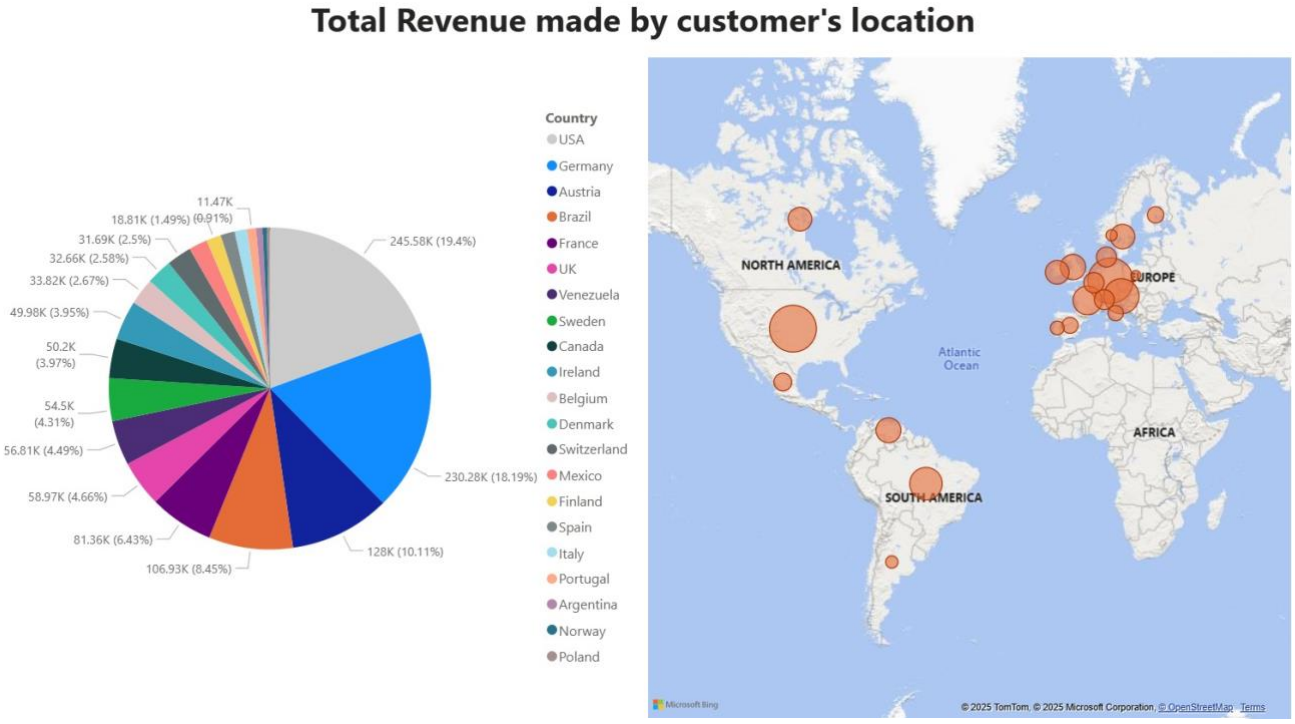


Figure 51: Revenue by customer's location

In the Figure 51, we can see the ranking of the best countries in terms of revenue. USA, Germany, Austria, Brazil and France are the largest in terms of sales. You can click on the dots to see the revenue coming from each country.

3) Products

The top product in terms of volume is the “Raclette Courdavault” and in terms of revenue, we have the “Côte de Blaye”. The point that you see in the right graph is the “Côte de Blaye”. Indeed, this is not the most sold in terms of volume, but as you will see in Figure 52, it’s the most expensive in our Inventory.

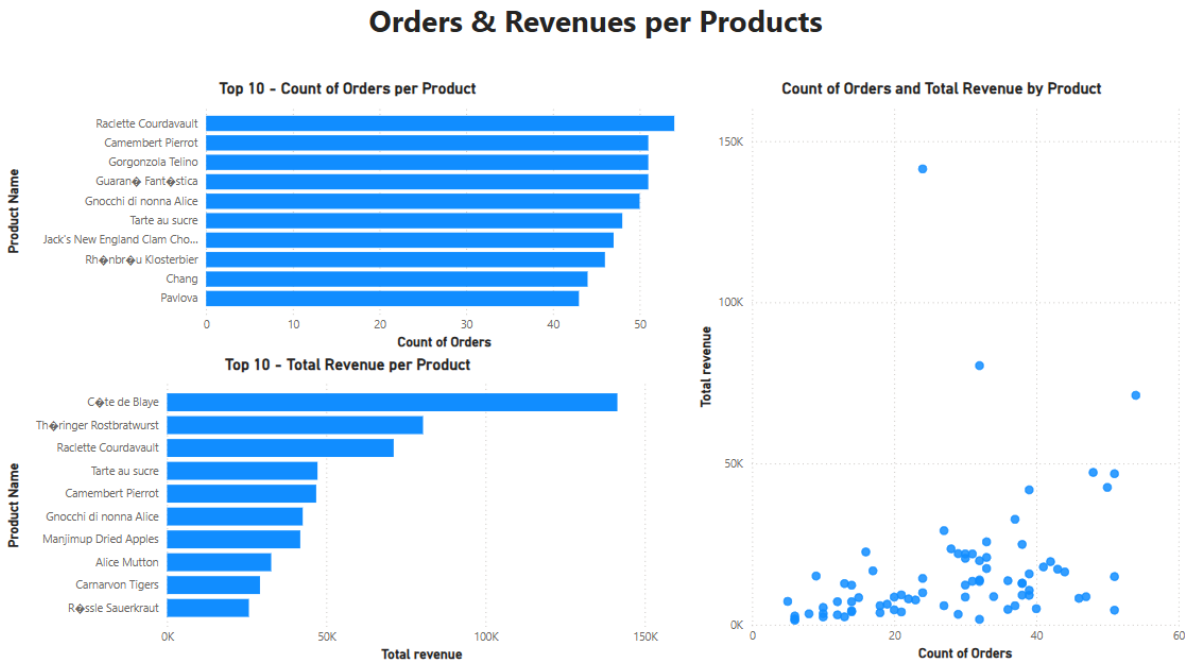
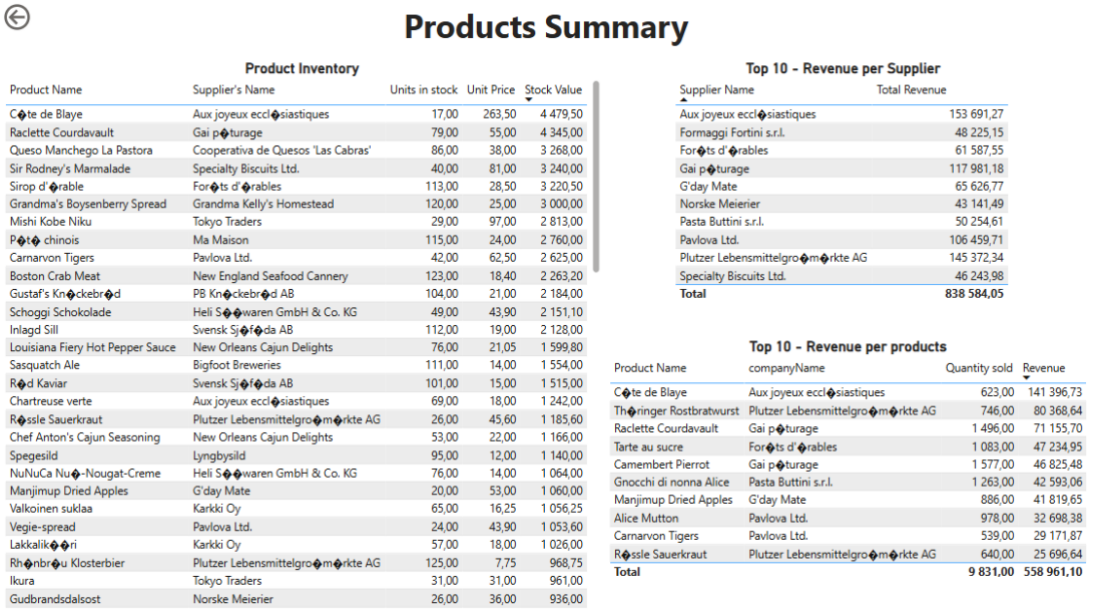


Figure 52: Orders & Revenues per Products

In the figure 53, you can see that the supplier which sell the most in terms of revenue is “Aux Joyeux ecclésiastiques”, and it’s without surprise that we observe that our top product is sold by our top suppliers.



Products Summary

Product Inventory

Product Name	Supplier's Name	Units in stock	Unit Price	Stock Value
C�te de Blaye	Aux joyeux eccl�siastiques	17,00	263,50	4 479,50
Raclette Courdavault	Gai p�turage	79,00	55,00	4 345,00
Queso Manchego La Pastora	Cooperativa de Quesos 'Las Cabras'	86,00	38,00	3 268,00
Sir Rodney's Marmalade	Specialty Biscuits Ltd.	40,00	81,00	3 240,00
Sirop d'�rable	For�ts d'�rables	113,00	28,50	3 220,50
Grandma's Boysenberry Spread	Grandma Kelly's Homestead	120,00	25,00	3 000,00
Mishi Kobe Niku	Tokyo Traders	29,00	97,00	2 813,00
P�t chinois	Ma Maison	115,00	24,00	2 760,00
Carnarvon Tigers	Pavlova Ltd.	42,00	62,50	2 625,00
Boston Crab Meat	New England Seafood Cannery	123,00	18,40	2 263,20
Gustaf's Kn�ckebr�d	PB Kn�ckebr�d AB	104,00	21,00	2 184,00
Schoggi Schokolade	Heli S��waren GmbH & Co. KG	49,00	43,90	2 151,10
Inlagd Sill	Svensk Sj�f�da AB	112,00	19,00	2 128,00
Louisiana Fiery Hot Pepper Sauce	New Orleans Cajun Delights	76,00	21,05	1 599,80
Sasquatch Ale	Bigfoot Breweries	111,00	14,00	1 554,00
R�d Kaviar	Svensk Sj�f�da AB	101,00	15,00	1 515,00
Chartreuse verte	Aux joyeux eccl�siastiques	69,00	18,00	1 242,00
R�ssle Sauerkraut	Plutzer Lebensmittelgro�m�rkte AG	26,00	45,60	1 185,60
Chef Anton's Cajun Seasoning	New Orleans Cajun Delights	53,00	22,00	1 166,00
Spegesild	Lyngbysild	95,00	12,00	1 140,00
NuNuCa Nu�-Nougat-Creme	Heli S��waren GmbH & Co. KG	76,00	14,00	1 064,00
Manjimup Dried Apples	G'day Mate	20,00	53,00	1 060,00
Valkoinen suklaa	Karkki Oy	65,00	16,25	1 056,25
Vegie-spread	Pavlova Ltd.	24,00	43,90	1 053,60
Lakkalik��ri	Karkki Oy	57,00	18,00	1 026,00
Rh�nbr�u Klosterbier	Plutzer Lebensmittelgro�m�rkte AG	125,00	7,75	968,75
Ikura	Tokyo Traders	31,00	31,00	961,00
Gudbrandsdalsost	Norske Meierier	26,00	36,00	936,00

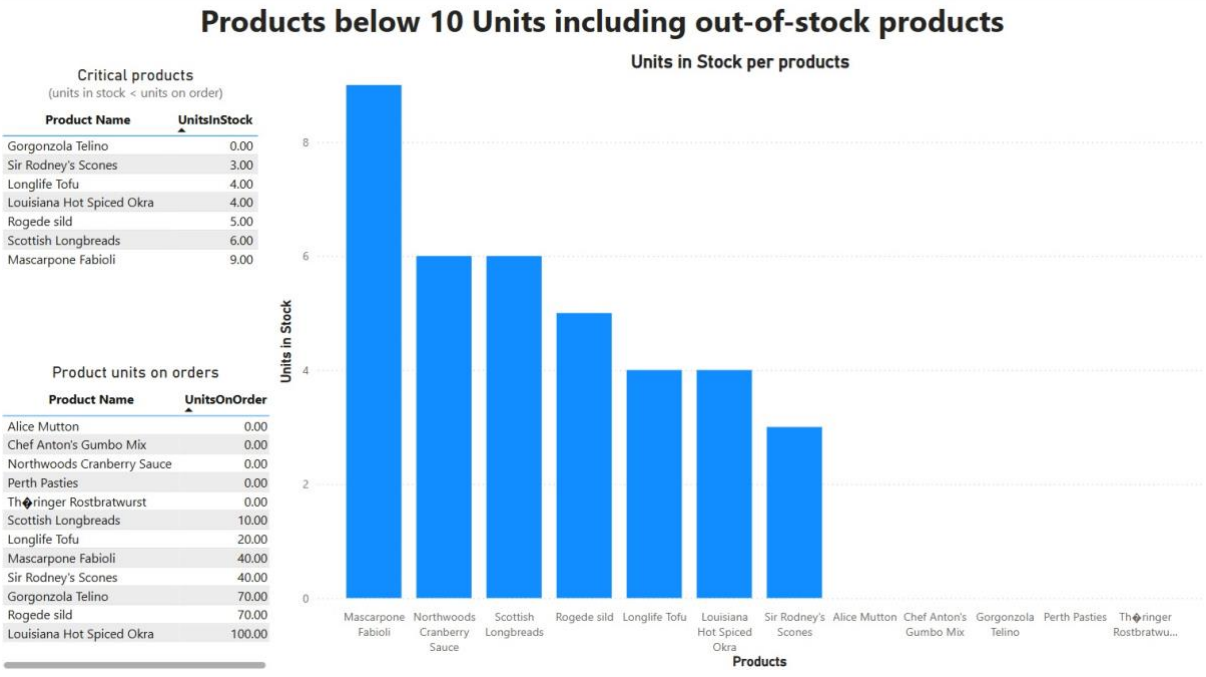
Top 10 - Revenue per Supplier

Supplier Name	Total Revenue
Aux joyeux eccl�siastiques	153 691,27
Formaggi Fortini s.r.l.	48 225,15
For�ts d'�rables	61 587,55
Gai p�turage	117 981,18
G'day Mate	65 626,77
Norske Meierier	43 141,49
Pasta Buttini s.r.l.	50 254,61
Pavlova Ltd.	106 459,71
Plutzer Lebensmittelgro�m�rkte AG	145 372,34
Specialty Biscuits Ltd.	46 243,98
Total	**838 584,05**

Top 10 - Revenue per products

Product Name	companyName	Quantity sold	Revenue
C�te de Blaye	Aux joyeux eccl�siastiques	623,00	141 396,73
Th�ringer Rostbratwurst	Plutzer Lebensmittelgro�m�rkte AG	746,00	80 368,64
Raclette Courdavault	Gai p�turage	1 496,00	71 155,70
Tarte au sucre	For�ts d'�rables	1 083,00	47 234,95
Camembert Pierrot	Gai p�turage	1 577,00	46 825,48
Gnocchi di nonna Alice	Pasta Buttini s.r.l.	1 263,00	42 593,06
Manjimup Dried Apples	G'day Mate	886,00	41 819,65
Alice Mutton	Pavlova Ltd.	978,00	32 698,38
Carnarvon Tigers	Pavlova Ltd.	539,00	29 171,87
R�ssle Sauerkraut	Plutzer Lebensmittelgro�m�rkte AG	640,00	25 696,64
Total		**9 831,00**	**558 961,10**

Figure 53: Product Summary



Products below 10 Units including out-of-stock products

Critical products
(units in stock < units on order)

Product Name	UnitsInStock
Gorgonzola Telino	0.00
Sir Rodney's Scones	3.00
Longlife Tofu	4.00
Louisiana Hot Spiced Okra	4.00
Rogede sild	5.00
Scottish Longbreads	6.00
Mascarpone Fabioli	9.00

Product units on orders

Product Name	UnitsOnOrder
Alice Mutton	0.00
Chef Anton's Gumbo Mix	0.00
Northwoods Cranberry Sauce	0.00
Perth Pasties	0.00
Th�ringer Rostbratwurst	0.00
Scottish Longbreads	10.00
Longlife Tofu	20.00
Mascarpone Fabioli	40.00
Sir Rodney's Scones	40.00
Gorgonzola Telino	70.00
Rogede sild	70.00
Louisiana Hot Spiced Okra	100.00

Units in Stock per products

Units in Stock: 0, 2, 4, 6, 8

Mascarpone Fabioli; Northwoods Cranberry Sauce; Scottish Longbreads; Rogede sild; Longlife Tofu; Louisiana Hot Spiced Okra; Sir Rodney's Scones; Alice Mutton; Chef Anton's Gumbo Mix; Gorgonzola Telino; Perth Pasties; Th�ringer Rostbratwu...

Products

Figure 54: Products below 10 units

Figure 54 shows the products that have less than 10 units in stock and those that are out of stock. This allows us to monitor the stocks and order products if we see that some are in a critical position.

4) Suppliers

We can also observe a correlation between the number of suppliers and shippers and the overall revenue. These actors were mainly concentrated in the USA, France, the Germany, and Australia (Figure 55). This suggests that in these countries, the supply chain and sales operations are more efficient, likely because businesses can more easily produce, distribute, and sell their products locally or internationally

Figure 55: Interactive map to see the suppliers' location

To answer the questions asked at the beginning of the analysis, we can say that the factors that influence the key performance indicators are the location of the customers and the price of the product they buy. In fact, we can see that the Côte de Blaye is our top product, even if it's not the one we sell the most. We also don't see any particular pattern in our data, except that Germany and the USA are the most dynamic countries in terms of sales, shipments and suppliers.

Conclusion

From this project, we learned how to tackle a robust, real-world scenario to build a modern data warehouse, from raw data ingestion to insightful reporting.

Throughout this project, we gained experience in:

- Dimensional modeling by conceiving a star-schema for organizing data,
- Creating a strong ETL design solution in SSIS,
- Data Type Handling & Lookup Challenges as we learned how to manage data type mismatches (e.g., datetime vs date), and how to use Derived Columns, Lookups, and Conditional Splits to ensure accurate joins and transformations,
- Error Handling & Debugging: We became more comfortable identifying and resolving common ETL issues, such as failed lookups, NULL mismatches, and incremental load configuration,
- Data Visualization by creating interactive dashboards to analyze global shipping coverage, top-selling products, and sales performance. These insights revealed high-revenue items, geographic shipping trends, and seasonal ordering patterns,
- Attention to Detail in Design by recognizing the importance of naming conventions, collation settings, and regional formatting standards in building a clean, professional, and user-friendly data solution.

Overall, this project helped us connect the dots between raw data, data architecture, and decision-making insights. It demonstrated how thoughtful data modeling and efficient ETL design are critical to unlocking the full potential of business intelligence tools.

To further improve the project, we can:

- Enhance the DimEmployees table by replacing the reportsTo column (varchar) with a integer parentEmployeeKEY to better support hierarchical queries.
- Clean the ODS_Orders pipeline by removing the columns including the name “Sender” in it because we edited the DWH_Orders to avoid duplicated columns and this was due to a minor issue identify at the last minute.