

Business Intelligence Project

Practical Business Intelligence Project: MACROBUS

Subject : **MacroBus** is a company that sells vehicles across various markets and aims to measure the performance of its sales representatives over time. Following a discussion with the CEO of **MacroBus**, he expresses the need to enhance the company's performance and effectively manage its operations. Given the current context, **MacroBus**'s priority is to have a decision support system in place to efficiently manage the thousands of orders it receives.

Presented by :

ADAMA FALL

INTRODUCTION.....	2
A. Design and Implementation.....	3
1) Propose Model : Snowflake Multidimensional Model.....	3
2) Implementation of this model using SSIS (Considering DimDate and the question).....	4
a. The MAIN package.....	4
b. The ODS package.....	5
i. Sequence Container	5
c. The DIM Package of the DIM Task	7
i. Sequence container 1	8
ii. Sequence container 2	11
d. Task FAIT and package FactCommande.....	13
3) Proposed script for loading the fact table through a stored procedure	13
4) Among other dimension tables, create the DimCommercial, DimSupplier, DimContact, DimGeography, and DimManager dimensions.	14
5) Proposal for a default record in each dimension table.	14
B. Functional Validation.....	Erreur ! Signet non défini.
1) Number and total amount of orders per territory. (Territory).....	17
2) Top 2 sales representatives who receive the most orders.....	18
3) The top three most ordered products during the second quarter of the year 2003.	19
4) Total amount of orders per subsidiary and per product category during the second half of the year 2003.	Erreur ! Signet non défini.
C. Creating visualizations in response to the previous questions (SAP Predictive Analysis).....	21
1. Number and total amount of orders per territory (Territory)	21
2. Top 2 sales representatives who receive the most orders.....	21
3. The top three most ordered products during the second quarter of the year 2003.	22
4. Total amount of orders per subsidiary and per product category during the second half of 2003.....	22
D. Summary dashboard for all four previous questions.....	23
E. Some execution screenshots of our SSIS packages.....	24
1. ODS	24
2. DIM	25
3. FactCommande.....	28
4. MAIN	28
F. Final Dashboard (Power BI)	29
CONCLUSION.....	30

INTRODUCTION

As part of the Business Intelligence final training project, we are required to implement a dashboard to assist the business leaders of the company **MACROBUS** in their decision-making process.

Our work will be focused on four (04) main areas, namely: designing and implementing a snowflake model tailored to our subject, ensuring the functional validation of our Data Warehouse, creating visualizations to address requirements and reporting goals, finally developing a dashboard using these visualizations.

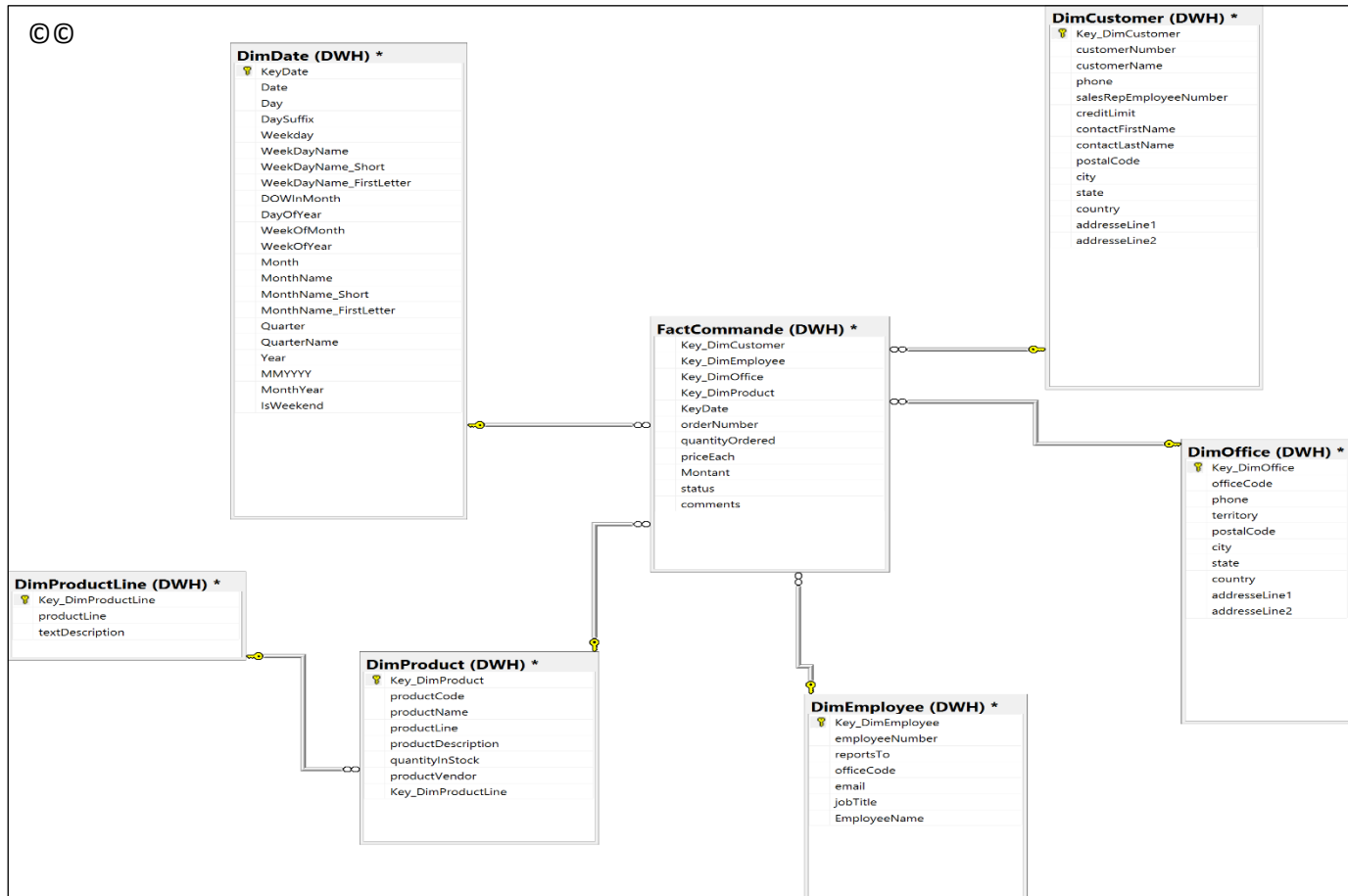
A. Design and Implementation

In this section, we will design our multidimensional model and proceed with its implementation.

Snowflake Multidimensional Model.

Initially, for the snowflake model, the dimensions DimDate, DimProduct, DimEmployee, DimCustomer, and DimOffice are the only dimensions linked to the FactCommande fact table. The ProductLine dimension is connected to the DimProduct dimension.

Hence, the following schema:



➤ Adding new dimensions

- Dimensions DimManager and DimCommercial

Managers represent a category of employees who oversee other employees.

Sales Representatives are employees who do not supervise any other employees.

Therefore, the Managers and Sales Representatives allow us to fully cover the DimEmployee dimension. As a result, the dimensions Dim Commercial and Dim Manager will be linked to the DimEmployee dimension.

- Dimension DimContact

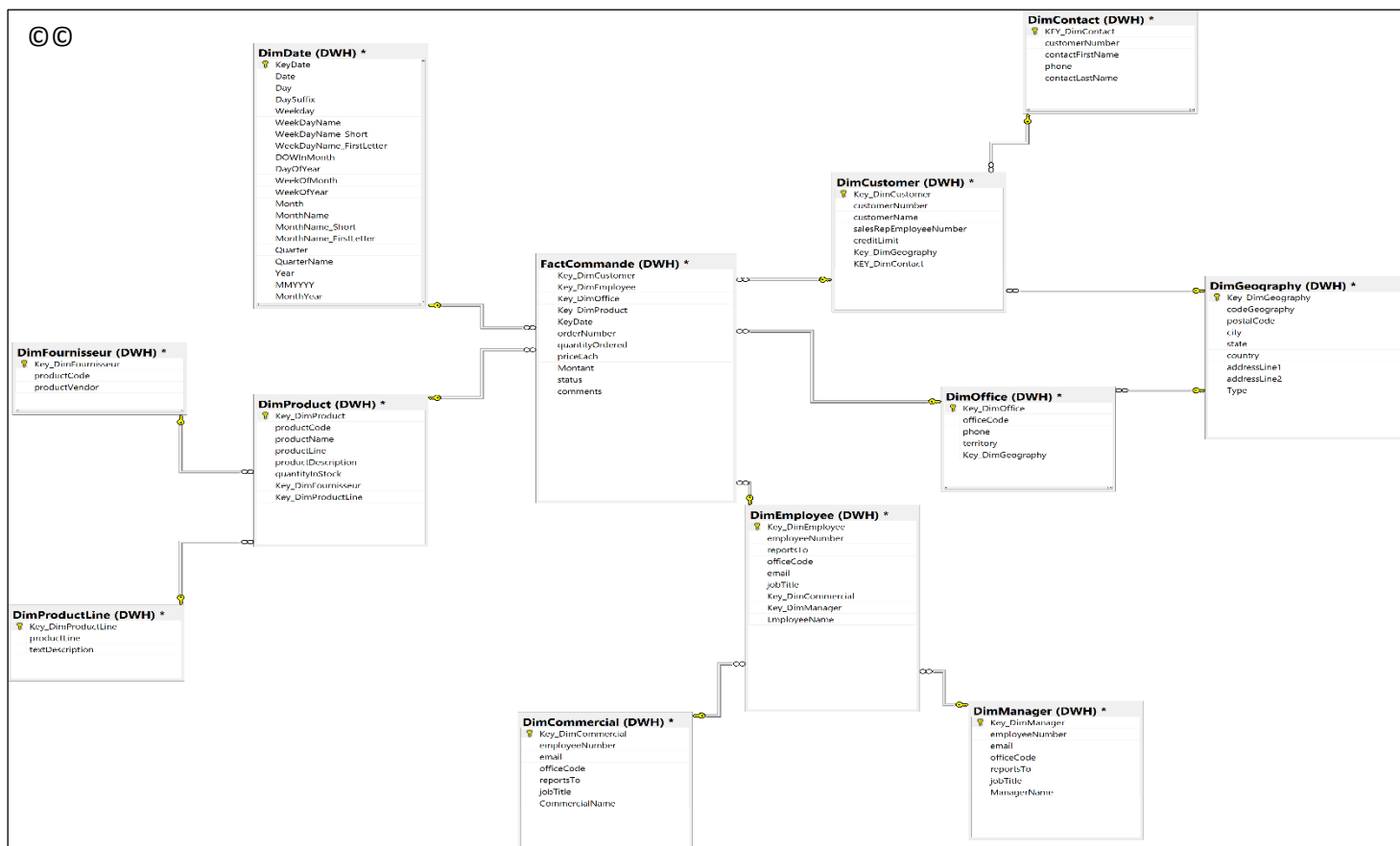
The DimCustomer dimension contains attributes: Contact FirstName and Contact LastName. These pieces of information can be grouped into a new dimension, resulting in the DimContact dimension.

- The dimension DimSupplier

The DimProduct dimension contains information about the Product Vendor. Hence, the new dimension DimSupplier will be created, containing information about the suppliers for each product. This dimension will be connected to the DimProduct dimension.

- The dimension DimGeographie

The DimCustomer and DimOffice dimensions both contain location information. Therefore, a new dimension called DimGeography will be created to hold location information for both DimCustomer and DimOffice. Technically, a new field "Type" will be inserted into DimGeography to facilitate the loading of DimCustomer and DimOffice dimensions:

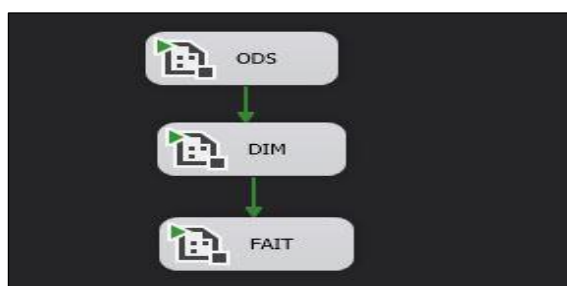


1) Implementation of this model using SSIS (Considering DimDate and the question)

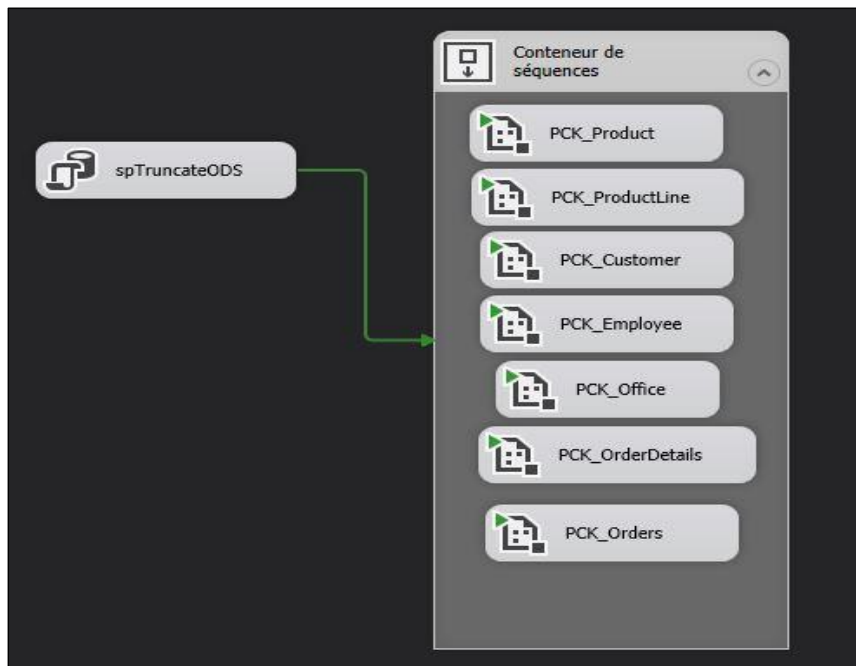
The main packages created for the project are:

- a. The MAIN package

It allows for sequencing the execution of our main packages, namely the ODS package, the DIM package, and the FACT package.



b. The ODS package



This package ensures the sequencing of tasks involved in loading the ODS (Operational Data Store) area of our data warehouse from the production MACROBUS_PROD database's base tables. spTruncateODS

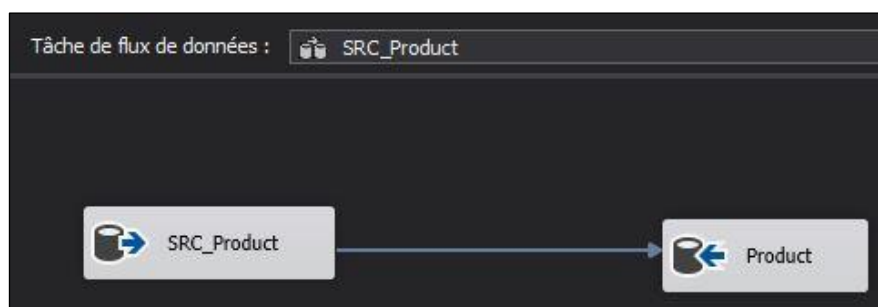
This represents a task for executing an SQL query, which in this case runs the stored procedure that clears all the contents of the tables in our ODS schema. This stored procedure is named dbo.spTruncateODS.

Sequence Container

It enables the parallel execution of tasks that facilitate loading our ODS schema from the MACROBUS_PROD database tables. These tasks are contained within the following package execution tasks: PCK_Product, PCK_ProductLine, PCK_Customer, PCK_Employee, PCK_Office, PCK_OrderDetails, PCK_Orders.

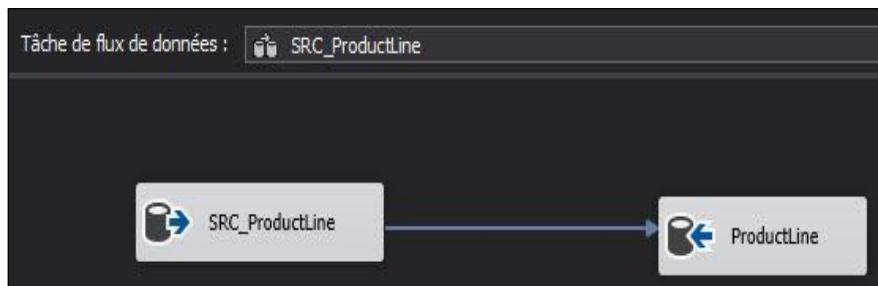
- **PCK_Product**

Loads the ods.Product table from the dbo.Product table in MACROBUS_PROD.



- **PCK_ProductLine**

Loads the ods.ProductLine table from the dbo.Product table in MACROBUS_PROD.



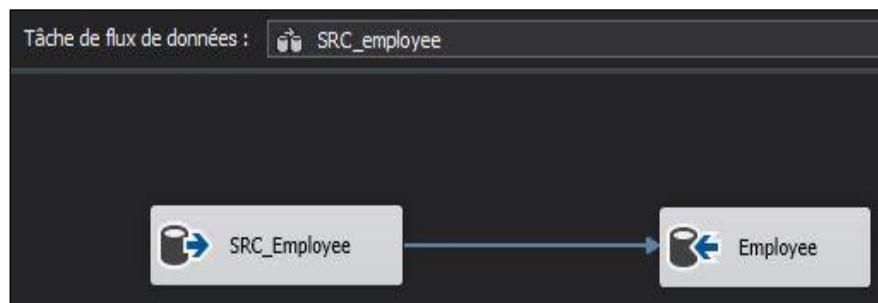
- **PCK_Customer**

Loads the ods.Customer table from the dbo.Customer table in MACROBUS_PROD.



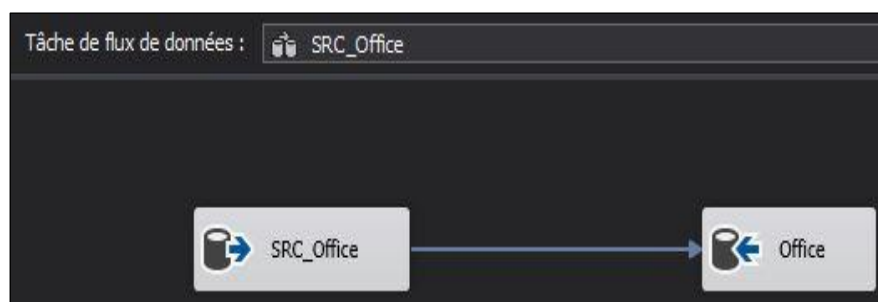
- **PCK_Employee**

Loads the ods.Employee table from the dbo.Employee table in MACROBUS_PROD.



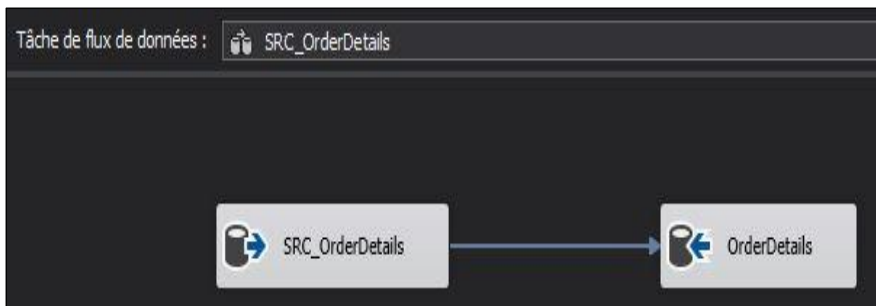
- **PCK_Office**

Load the ods.Office table from the dbo.Office table in MACROBUS_PROD.



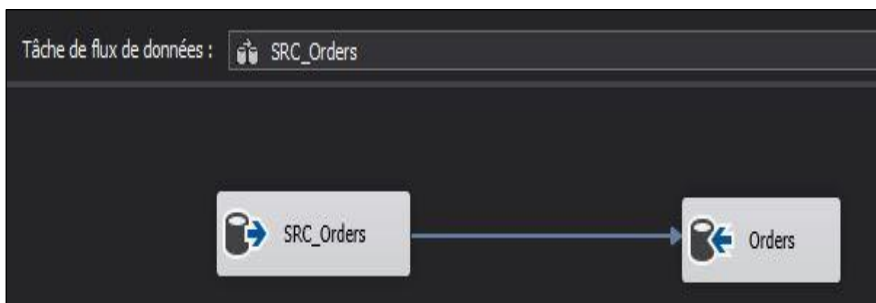
- **PCK_OrderDetails**

Loads the ods.OrderDetails table from the dbo.OrderDetails table in MACROBUS_PROD.

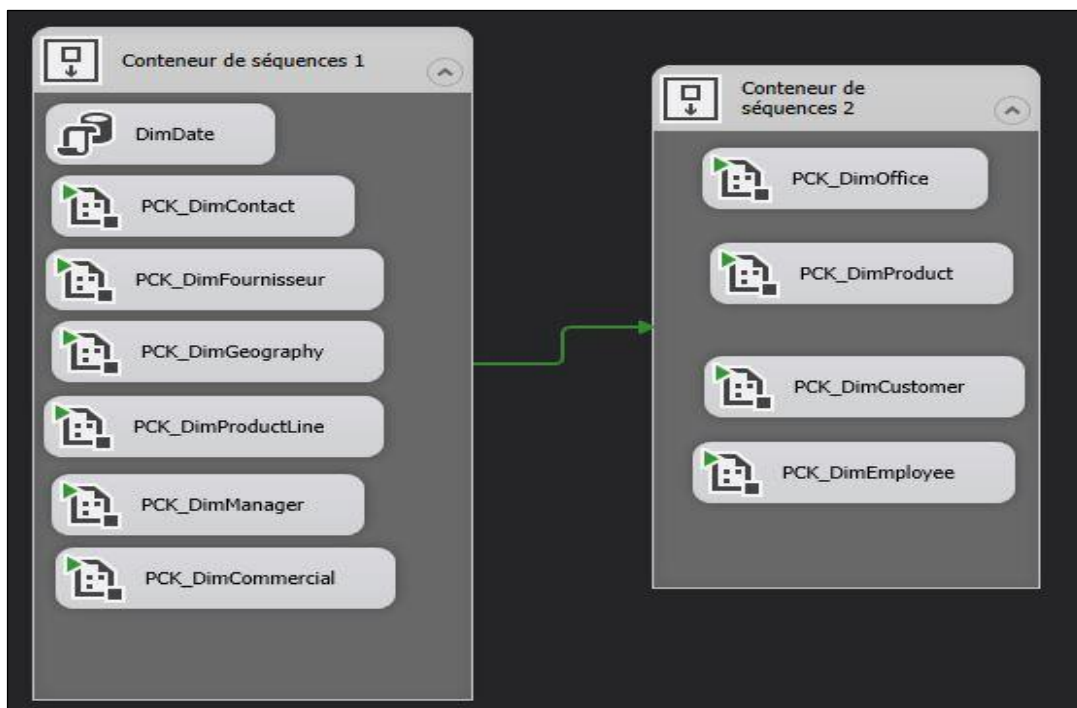


- **PCK_Orders**

Loads the ods.Orders table from the dbo.Orders table in MACROBUS_PROD.



c. The DIM Package of the DIM Task



This package orchestrates the execution of package tasks that contribute to loading all dimensions of the DWH schema in our Data Warehouse.

To execute the package execution task PCK_DimEmployee, the Sequence Container 2 needs to be executed beforehand. And to execute the tasks within Sequence Container 1...

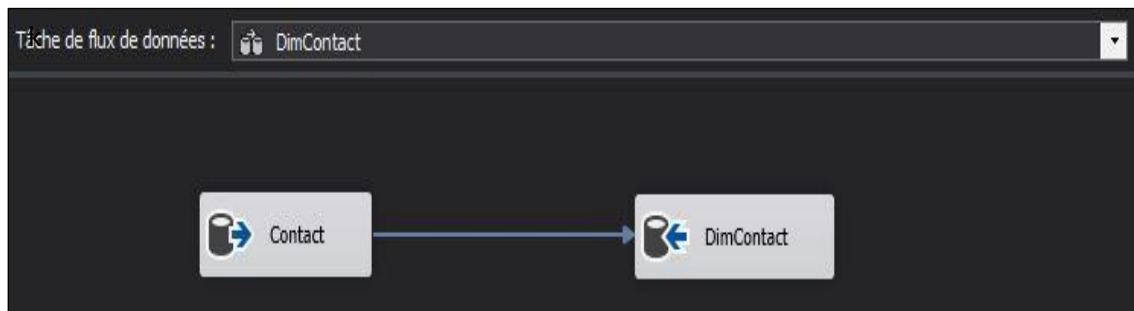
i. Sequence container 1

- **Dimension DimDate**

It allows for the execution of the stored procedure that loads the DimDate dimension.

- **Task PCK_DimContact**

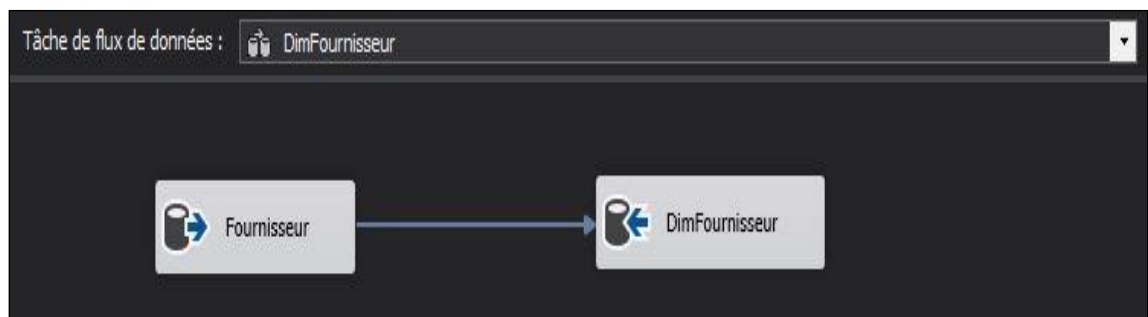
Enables the execution of the package that loads the DimContact dimension into our Data Warehouse.



The Contact source points to the ods.Customer table, and we select only the columns of interest: Contact FirstName, Contact LastName, phone, and CustomerNumber.

- **Task PCK_DimFournisseur**

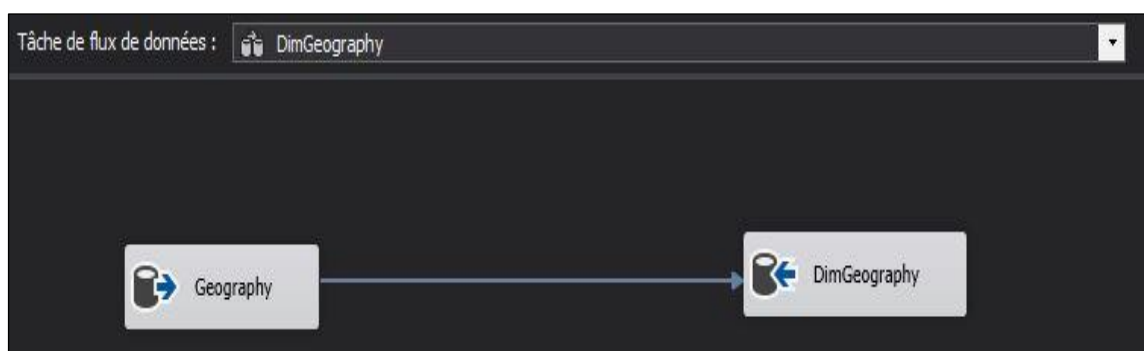
Allows the execution of the package that loads the DimSupplier dimension into our Data Warehouse.



The **Supplier** source is pointed to the ods.Product table, and we select the columns: ProductCode and ProductVendor.

- **Task PCK_DimGeography**

Allows the execution of the package that loads the DimGeography dimension into our Data Warehouse.

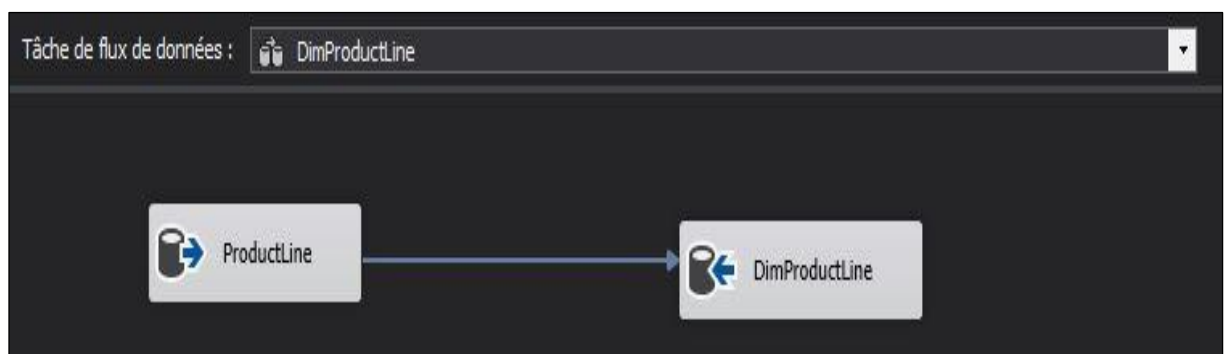


As mentioned in question 1, we are using the ods.Customer and ods.Office tables here to populate the Geography source. Hence, the new column "Type" will have a value of 'office' for a record originating from the ods.office table, and a value of 'customer' for a record originating from the ods.Customer table.

```
Texte de la commande SQL :  
  
select [officecode] as codeGeography, [postalCode], [city], [state],  
[country], [addressLine1], [addressLine2], 'office' as Type  
from [ODS].[Office]  
union  
select [CustomerNumber] as codeGeography, [postalCode], [city],  
[state], [country], [addressLine1], [addressLine2], 'customer' as Type  
from [ODS].[Customer]
```

Task PCK DimProductLine

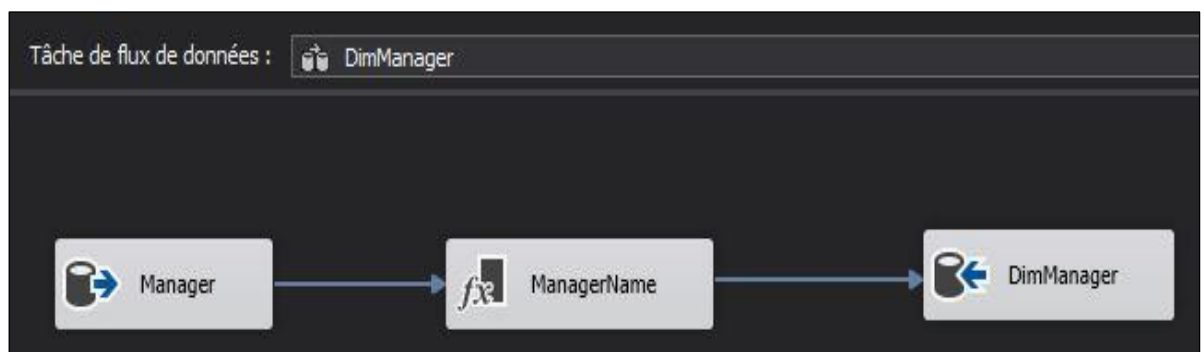
Allows the execution of the package that loads the DimProductLine dimension into our Data Warehouse.



The ProductLine source is pointed to the ods.Product table, and we select the columns: ProductLine and ProductDescription.

Task PCK DimManager

Allows the execution of the package that loads the DimManager dimension into our Data Warehouse.



The Manager source contains all employees from ods. Employee who supervise other employees. We obtain them through the following query:

```

Commande SQL

Texte de la commande SQL :

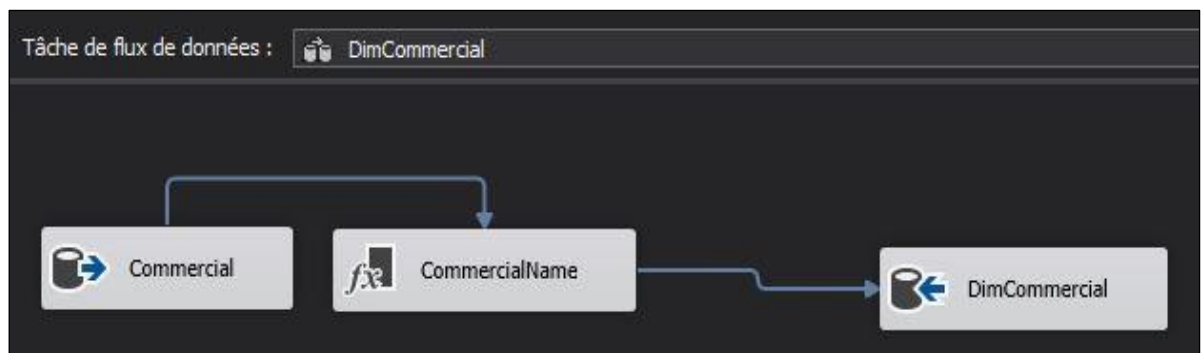
SELECT [employeeNumber]
,[lastName]
,[firstName]
,[extension]
,[email]
,[officeCode]
,[reportsTo]
,[jobTitle]
FROM [MACROBUS_DWH_MEDJEU FOPA JUNIOR].[ODS].[employee]
WHERE [employeeNumber] IN (SELECT DISTINCT [reportsTo] FROM
[MACROBUS_DWH_MEDJEU FOPA JUNIOR].[ODS].[employee])

```

Then we create a calculated column named **ManagerName** that will contain the manager's last name and first name.

Task PCK DimCommercial

Allows the execution of the package that loads the DimCommercial dimension into our Data Warehouse.



The Commercial source contains all employees from ods.Employee who do not supervise any employees. We obtain them through the following query:

```

Texte de la commande SQL :

SELECT [employeeNumber]
,[lastName]
,[firstName]
,[extension]
,[email]
,[officeCode]
,[reportsTo]
,[jobTitle]
FROM [ODS].[employee]
except
SELECT [employeeNumber]
,[lastName]
,[firstName]
,[extension]
,[email]
,[officeCode]
,[reportsTo]
,[jobTitle]
FROM [ODS].[employee]
WHERE [employeeNumber] IN (SELECT DISTINCT [reportsTo] FROM [ODS].[employee])

```

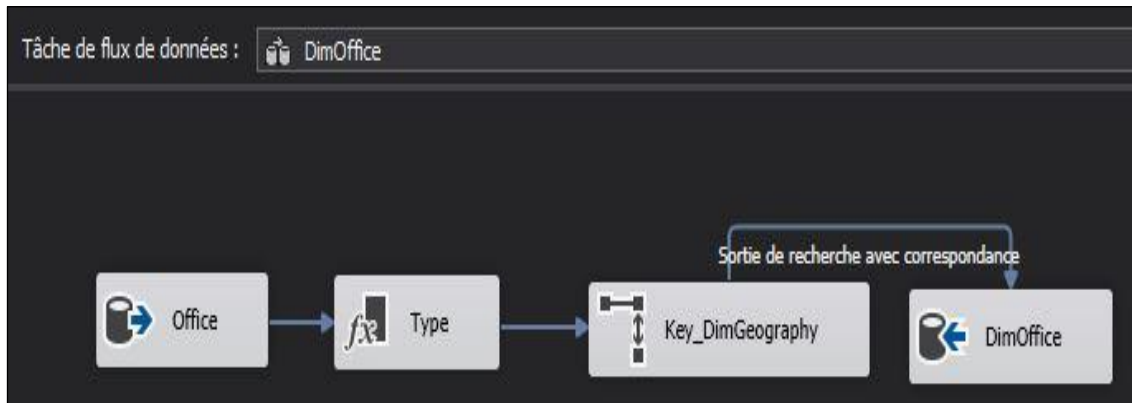
Then we create a calculated column named **CommercialName** that will contain the commercial's last name and first name.

Sequence container 2

After the successful execution of all tasks within Sequence Container 1, we can now proceed to the tasks within Sequence Container 2.

Task PCK DimOffice

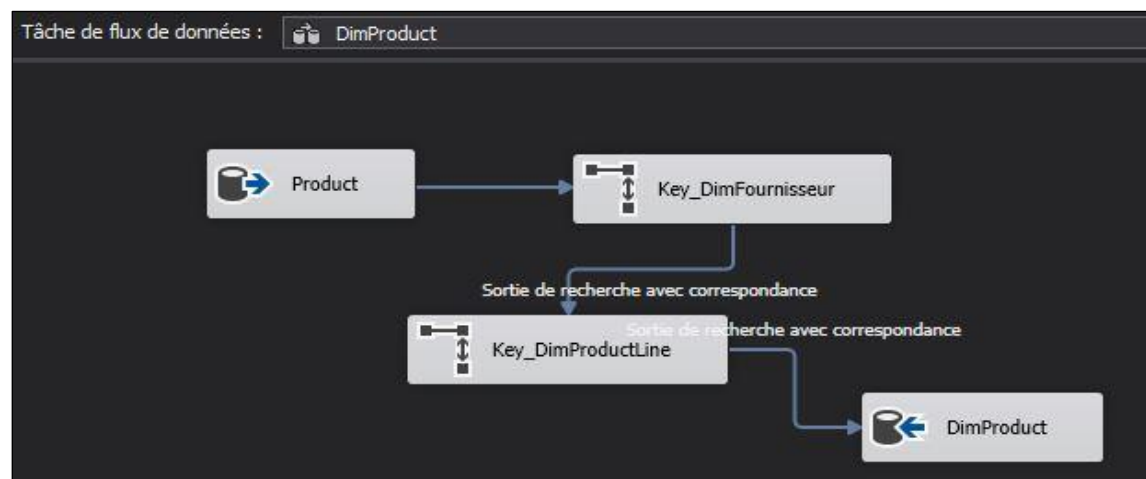
Allows the execution of the package that loads the DimOffice dimension into our Data Warehouse.



The **Office** source is pointed to the ods.Office table. Then, we create a calculated column named **Type** and set it to 'office'. Afterwards (using a lookup object), we join the OfficeCode and Type columns with the codeGeography and Type columns of the DimGeography dimension to retrieve its **Key_DimGeography** key.

Task PCK DimProduct

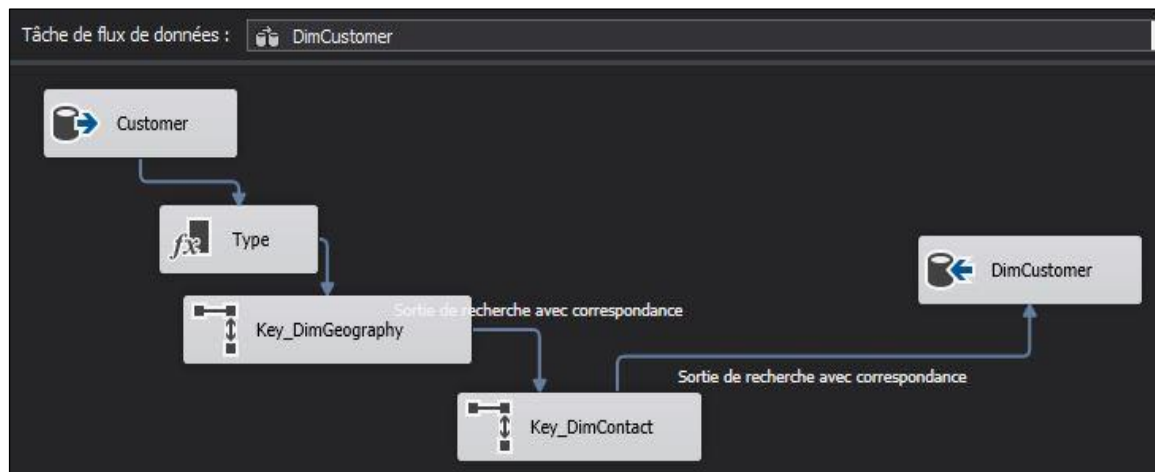
Allows the execution of the package that loads the DimProduct dimension into our Data Warehouse.



The Product source is pointed to the ods.Product table, and we retrieve the information that will be useful to us: ProductCode, ProductName, ProductLine, and Quantity in Stock. Then, using a lookup object, we retrieve the attribute **Key_DimFournisseur** of the dimension DimSupplier using the ProductCode attribute, which is shared with our source. Then, we retrieve the **Key_DimProductLine** attribute from the DimProductLine dimension using the ProductLine attribute, which is also shared with our source.

Task PCK DimCustomer

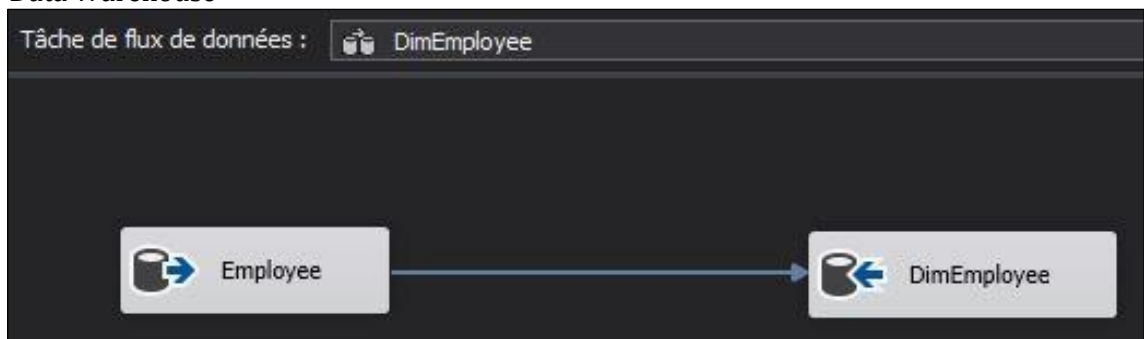
Allows the execution of the package that loads the DimCustomer dimension into our Data Warehouse.



The Customer source is pointed to the **ods.Customer** table. Then, we create a calculated column named Type and set it to 'customer'. After that, we create a lookup object where we join the CustomerNumber and Type columns with the codeGeography and Type columns of the DimGeography dimension to retrieve its Key_DimGeography key. Subsequently, we create another lookup object where we join the CustomerNumber column from our source with the CustomerNumber column of the DimContact dimension to retrieve the Key_DimContact key from the latter.

Task PCK DimEmployee

Allows the execution of the package that loads the DimEmployee dimension into our Data Warehouse



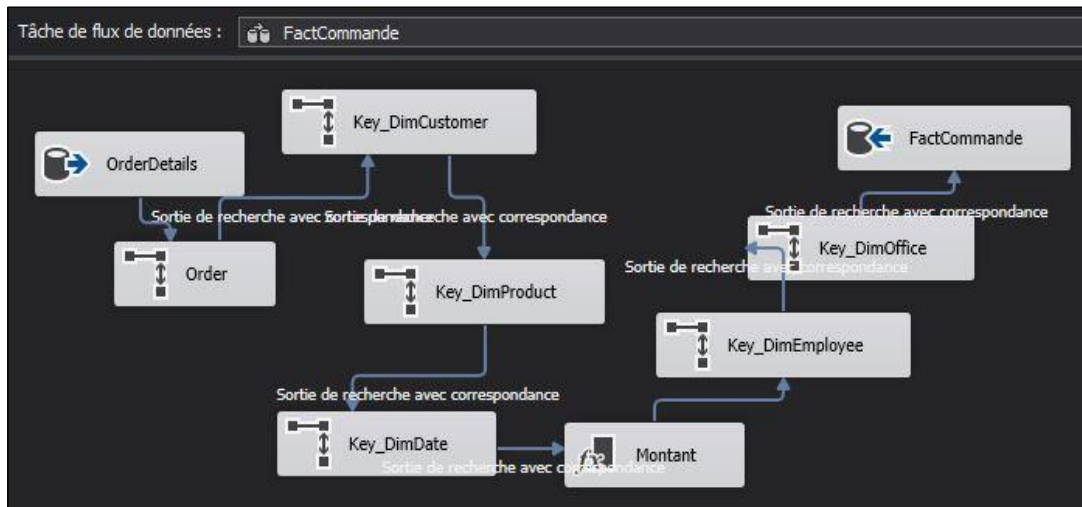
As mentioned in question 1), our employee source is populated from the DimCommercial and DimManager dimensions, as indicated by the following query:

```

Texte de la commande SQL :
SELECT [Key_DimCommercial]
, -1 as [Key_DimManager]
,[employeeNumber]
,[email]
,[officeCode]
,[reportsTo]
,[jobTitle]
,[CommercialName] as [EmployeeName]
FROM [DWH].[DimCommercial]
union
SELECT -1 as [Key_DimCommercial]
,[Key_DimManager]
,[employeeNumber]
,[email]
,[officeCode]
,[reportsTo]
,[jobTitle]
,[ManagerName] as [EmployeeName]
FROM [DWH].[DimManager]
  
```

d. Task and package FactCommande

This task enables the execution of the FactCommande package, which is responsible for loading the fact table in our Data Warehouse.



Our fact table is constructed from 5 regular dimensions (DimCustomer, DimProduct, DimDate, DimEmployee, and DimOffice) and a degenerate dimension (OrderDetails) since all attributes of the latter are found in the fact table.

We begin by retrieving the content of the ods.OrderDetails table. Then, using a lookup object, we join it with the ods.Orders table based on the common attribute OrderNumber to obtain the CustomerNumber. This CustomerNumber is subsequently used (through another lookup object) to retrieve the Key_DimCustomer key through a join on their common attribute CustomerNumber.

Next, through a lookup object, we retrieve the Key_DimProduct key from the DimProduct dimension using a join on the ProductCode attribute, which is shared with our degenerate dimension (ods.OrderDetails).

Further, using a lookup object, we retrieve the Key_DimDate key from the DimDate dimension through a join between ods.Orders.OrderDate and dwh.DimDate.Date.

Following this, a calculated column is created to generate a new column called Amount, which results from the product of quantityOrdered and Unitprice.

Subsequently, through a lookup object and a join between dwh.DimCustomer.SalesRepEmployeeNumber and dwh.DimEmployee.EmployeeNumber, we retrieve the Key_DimEmployee key.

Finally, using another lookup object and a join between dwh.DimEmployee.OfficeCode and dwh.DimOffice.OfficeCode, we retrieve the Key_DimOffice key from the DimOffice dimension.

2) Proposed script for loading the fact table through a stored procedure


```

IF OBJECT_ID (N'DWH.FactCommande',N'U') IS NOT NULL
    DROP TABLE [DWH].[FactCommande]
SELECT [Key_DimCustomer]
      ,[Key_DimEmployee]
      ,[Key_DimOffice]
      ,[Key_DimProduct]
      ,da.[KeyDate]
      ,od.[orderNumber]
      ,[quantityOrdered]
      ,[priceEach]
      ,[quantityOrdered] * od.[priceEach] as [Montant]
      ,[status]
      ,[comments] into [DWH].[FactCommande]
FROM [ODS].[OrderDetails] od
left join [ODS].[Orders] o on od.[orderNumber] = o.[orderNumber]
left join [DWH].[DimCustomer] cu on o.[customerNumber] = cu.customerNumber
left join [DWH].[DimProduct] pro on od.[productCode] = pro.[productCode]
left join [DWH].[DimDate] da on o.[orderDate] = da.[Date]
left join [DWH].[DimEmployee] em on cu.[salesRepEmployeeNumber] = em.[employeeNumber]
left join [DWH].[DimOffice] offi on em.[officeCode] = offi.[officeCode]

```

3) Among other dimension tables, create the DimCommercial, DimSupplier, DimContact, DimGeography, and DimManager dimensions.

Please refer to questions 1) and 2).

4) Proposal for a default record in each dimension table.

```

USE [MACROBUS_DWH_MEDJEU_FOPA_JUNIOR]
GO

set identity_insert [DWH].[DimCommercial] ON
INSERT INTO [DWH].[DimCommercial]
      ([Key_DimCommercial]
      ,[employeeNumber]
      ,[email]
      ,[officeCode]
      ,[reportsTo]
      ,[jobTitle]
      ,[CommercialName])
VALUES (0, -1, 'no email', -1, -1, 'no Title', 'no commercial Name')
set identity_insert [DWH].[DimCommercial] OFF
GO

set identity_insert [DWH].[DimContact] ON
INSERT INTO [DWH].[DimContact]
      ([KEY_DimContact]
      ,[customerNumber]
      ,[contactFirstName]
      ,[phone]
      ,[contactLastName])
VALUES (0, -1, 'no first name', 'no phone Number', 'no last name')
set identity_insert [DWH].[DimContact] OFF
GO

set identity_insert [DWH].[DimCustomer] ON
INSERT INTO [DWH].[DimCustomer]
      ([Key_DimCustomer]
      ,[customerNumber]
      ,[customerName]
      ,[phone]
      ,[salesRepEmployeeNumber]
      ,[creditLimit]
      ,[Key_DimGeography]
      ,[KEY_DimContact])

```

```

VALUES(0, -1, 'no customer name', 'no phone number', -1, -1, 0, 0)
set identity_insert [DWH].[DimCustomer] OFF
GO
set identity_insert [DWH].[DimEmployee] ON
INSERT INTO [DWH].[DimEmployee]
    ([Key_DimEmployee]
    ,[employeeNumber]
    ,[reportsTo]
    ,[officeCode]
    ,[email]
    ,[jobTitle]
    ,[Key_DimCommercial]
    ,[Key_DimManager]
    ,[EmployeeName])
VALUES (0, -1, -1, -1, 'no email', 'no job Title', 0, 0, 'no EmployeeName')
set identity_insert [DWH].[DimEmployee] OFF
go
set identity_insert [DWH].[DimFournisseur] ON
INSERT INTO [DWH].[DimFournisseur]
    ([Key_DimFournisseur]
    ,[productCode]
    ,[productVendor])
VALUES (0, 'no product code', 'no productVendor')
set identity_insert [DWH].[DimFournisseur] OFF
go
set identity_insert [DWH].[DimGeography] ON
INSERT INTO [DWH].[DimGeography]
    ([Key_DimGeography]
    ,[codeGeography]
    ,[postalCode]
    ,[city]
    ,[state]
    ,[country]
    ,[addressLine1]
    ,[addressLine2]
    ,[Type])

```

```

VALUES (0, -1, 'no postal code', 'no city', 'no state', 'no country', 'no addressLine1', 'no addressLine2', 'no Type')
set identity_insert [DWH].[DimGeography] OFF
GO
set identity_insert [DWH].[DimManager] ON
INSERT INTO [DWH].[DimManager]
    ([Key_DimManager]
    ,[employeeNumber]
    ,[email]
    ,[officeCode]
    ,[reportsTo]
    ,[jobTitle]
    ,[ManagerName])
VALUES (0, -1, 'no email', -1, -1, 'no Job title', 'no Manager name')
set identity_insert [DWH].[DimManager] OFF
GO
set identity_insert [DWH].[DimOffice] ON
INSERT INTO [DWH].[DimOffice]
    ([Key_DimOffice]
    ,[officeCode]
    ,[phone]
    ,[territory]
    ,[Key_DimGeography])
VALUES (0, -1, 'no phone', 'no territory', 0)
set identity_insert [DWH].[DimOffice] OFF
go
set identity_insert [DWH].[DimProduct] ON
INSERT INTO [DWH].[DimProduct]
    ([Key_DimProduct]
    ,[productCode]
    ,[productName]
    ,[productLine]
    ,[productDescription]
    ,[quantityInStock]
    ,[Key_DimFournisseur]

```



```
        ,[Key_DimProductLine])
    VALUES (0, 'no product code', 'no product name', 'no product Line', 'no product Description', 0, 0, 0)
set identity_insert [DWH].[DimProduct] OFF
go
set identity_insert [DWH].[DimProductLine] ON
INSERT INTO [DWH].[DimProductLine]
    ([Key_DimProductLine]
    ,[productLine]
    ,[textDescription])
    VALUES (0, 'no productLine', 'no textDescription')
set identity_insert [DWH].[DimProductLine] OFF
```

FUNCTIONAL VALIDATION

Answers to the questions asked in the production database and in the Data Warehouse (DWH).

1) Number and total amount of orders per territory. (Territory)

reponse aux quest...ER\CompaqPC (52))* X SQLQuery17.sql -...SER\CompaqPC (53))* ~vsF574.sql - USE...SER\CompaqPC (51))

```
--1. 1) Nombre et montant total des commandes par territoire (Territory)
-- Base de production
select offi.[territory], count(*) as [Nombre total], sum(od.[quantityOrdered]*od.[priceEach]) as [Montant Total]
from [ODS].[orders] o
join [ODS].[orderDetails] od on o.[orderNumber] = od.[orderNumber]
join [ODS].[Customer] cu on o.[customerNumber] = cu.[customerNumber]
join [ODS].[Employee] emp on cu.[salesRepEmployeeNumber] = emp.[employeeNumber]
join [ODS].[Office] offi on emp.[officeCode] = offi.[officeCode]

group by offi.[territory]
go

-- DatawareHouse
select offi.[territory], count(*) as [Nombre total], sum(f.[Montant]) as [Montant Total]
from [DWH].[FactCommande] f
join [DWH].[DimOffice] offi on f.[Key_DimOffice] = offi.[Key_DimOffice]

group by offi.[territory]
```

100 %

Résultats Messages

	territory	Nombre total	Montant Total
1	EMEA	1415	4520712,27999999
2	APAC	370	1147176,35
3	Japan	137	457110,07
4	NA	1074	3479191,91

	territory	Nombre total	Montant Total
1	APAC	370	1147176,35
2	EMEA	1415	4520712,28
3	Japan	137	457110,07
4	NA	1074	3479191,91

2) Top 2 sales representatives who receive the most orders.

```
--2. 2) Top 2 des commerciaux qui reçoivent plus de commandes.
-- Base de Production
with cte_commerciaux
as
(SELECT [employeeNumber], [lastName], [firstName], [extension], [email], [reportsTo], [jobTitle]
FROM [ODS].[employee]
except
SELECT [employeeNumber], [lastName], [firstName], [extension], [email], [reportsTo], [jobTitle]
FROM [ODS].[employee]
WHERE [employeeNumber] IN (SELECT DISTINCT [reportsTo] FROM [ODS].[employee])
)
select Top(2) cte.[employeeNumber] as [Id Commercial], cte.[lastName]+' '+cte.[firstName] as [Nom du Commercial], count(*) as [Total Commande]
from [ODS].[OrderDetails] od
join [ODS].[Orders] o on od.[orderNumber] = o.[orderNumber]
join [ODS].[Customer] cu on o.[customerNumber] = cu.[customerNumber]
join cte_commerciaux cte on cu.[salesRepEmployeeNumber] = cte.[employeeNumber]
group by cte.[lastName]+' '+cte.[firstName], cte.[employeeNumber]
order by [Total Commande] desc

-- DatawareHouse
select Top(2) com.[employeeNumber] as [Id commercial], com.[CommercialName] as [Nom du Commercial], count(*) as [Total Commande]
from [DWH].[FactCommande] f
join [DWH].[DimEmployee] emp on f.[Key_DimEmployee] = emp.[Key_DimEmployee]
join [DWH].[DimCommercial] com on emp.[Key_DimCommercial] = com.[Key_DimCommercial]
group by com.[employeeNumber], com.[CommercialName]
order by [Total Commande] desc
```

100 %

Résultats			Messages
	Id Commercial	Nom du Commercial	Total Commande
1	1370	Hernandez Gerard	396
2	1165	Jennings Leslie	331

	Id commercial	Nom du Commercial	Total Commande
1	1370	Hernandez Gerard	396
2	1165	Jennings Leslie	331

3) The top three most ordered products during the second quarter of the year 2003.

reponse aux quest...ER\CompaqPC (53))* X ~vsF574.sql - non connecté

```
--3) Les trois produits les plus commandé commandés durant le deuxième trimestre de l'année 2003.  
-- Base de production  
select Top(3) od.[productCode] , pro.[productName], sum(od.quantityOrdered) as [Total Commandé Trimestre 2 Année 2003]  
from [ODS].[OrderDetails] od  
join [ODS].[Orders] o on od.[orderNumber] = o.[orderNumber]  
join [ODS].[Product] pro on od.[productCode] = pro.[productCode]  
where (datepart(QUARTER, o.[orderDate])=2 and datepart(YEAR, o.[orderDate])=2003)  
  
group by od.[productCode] , pro.[productName]  
order by [Total Commandé Trimestre 2 Année 2003] desc  
  
-- DatawareHouse  
select Top(3) pro.[productCode] , pro.[productName], sum(f.quantityOrdered) as [Total Commandé Trimestre 2 Année 2003]  
from [DWH].[FactCommande] f  
join [DWH].[DimProduct] pro on f.[Key_DimProduct] = pro.[Key_DimProduct]  
join [DWH].[DimDate] da on f.[KeyDate] = da.[KeyDate]  
where (da.Quarter=2 and da.year=2003)  
  
group by pro.[productCode] , pro.[productName]  
order by [Total Commandé Trimestre 2 Année 2003] desc
```

100 %

Résultats Messages

	productCode	productName	Total Commandé Trimestre 2 Année 2003
1	S18_3232	1992 Ferrari 360 Spider red	95
2	S18_4600	1940s Ford truck	91
3	S18_2238	1998 Chrysler Plymouth Prowler	91

	productCode	productName	Total Commandé Trimestre 2 Année 2003
1	S18_3232	1992 Ferrari 360 Spider red	95
2	S18_4600	1940s Ford truck	91
3	S18_2238	1998 Chrysler Plymouth Prowler	91

4) Determine the combined order value for each subsidiary and product

```
reponse aux quest...ER\CompaqPC (52))* X SQLQuery17.sql -...SER\CompaqPC (53))* ~vsF574.sql - USE...SER\CompaqPC (51))*  
--4.4) Montant total des commandes par filiale et par catégorie de produit durant le second semestre de 2003.  
-- Base de Production  
select emp.[officeCode] as [Filiale], pro.[productLine], sum(od.[quantityOrdered]*od.[priceEach]) as [Montant Total des Commandes]  
from [ODS].[OrderDetails] od  
join [ODS].[Product] pro on od.[productCode] = pro.[productCode]  
join [ODS].[Orders] o on od.[orderNumber] = o.[orderNumber]  
join [ODS].[Customer] cu on o.[customerNumber] = cu.[customerNumber]  
join [ODS].[Employee] emp on cu.[salesRepEmployeeNumber] = emp.[employeeNumber]  
where (datepart(QUARTER, o.[orderDate])>=3 and datepart(YEAR, o.[orderDate])=2003)  
group by emp.[officeCode], pro.[productLine]  
order by emp.[officeCode], pro.[productLine]  
  
-- DatawareHouse  
select offi.[officeCode] as [Filiale], pro.[productLine], sum(f.Montant) as [Montant Total des Commandes]  
from [DWH].[FactCommande] f  
join [DWH].[DimProduct] pro on f.[Key_DimProduct] = pro.[Key_DimProduct]  
join [DWH].[DimOffice] offi on f.[Key_DimOffice] = offi.[Key_DimOffice]  
join [DWH].[DimDate] da on f.[KeyDate] = da.[KeyDate]  
where (da.Quarter>=3 and da.year=2003)  
group by offi.[officeCode], pro.[productLine]  
order by offi.[officeCode], pro.[productLine]
```

100 %

Résultats Messages

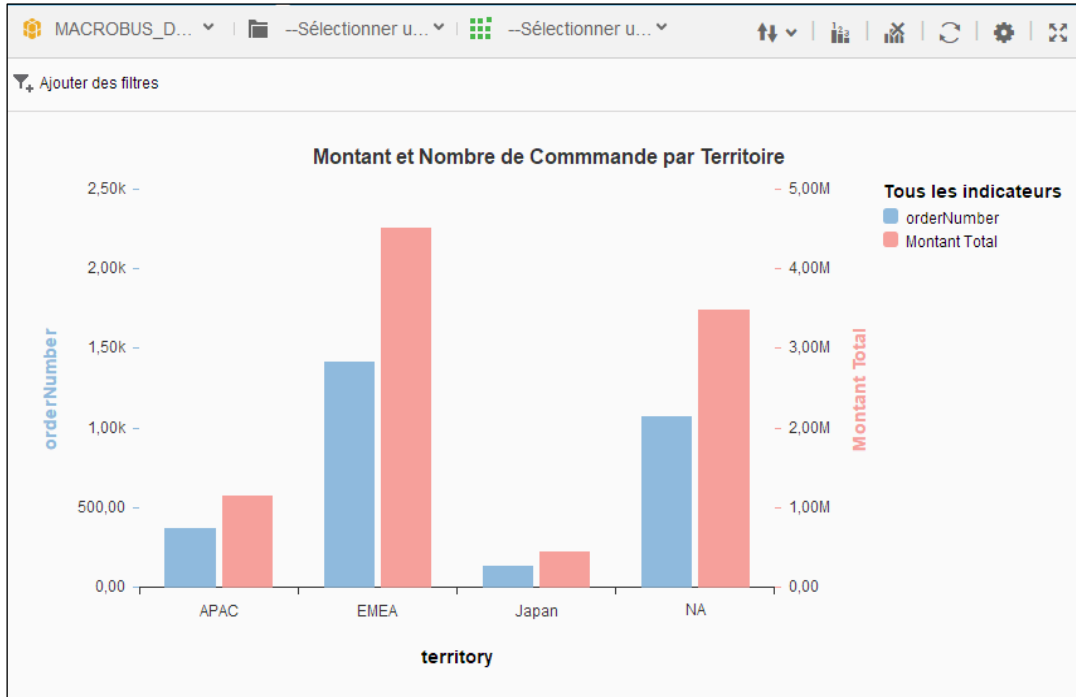
	Filiale	productLine	Montant Total des Commandes
1	1	Classic Cars	140221,14
2	1	Motorcycles	132639,97
3	1	Planes	42609,35
4	1	Ships	18727,53

	Filiale	productLine	Montant Total des Commandes
1	1	Classic Cars	140221,14
2	1	Motorcycles	132639,97
3	1	Planes	42609,35
4	1	Ships	18727,53

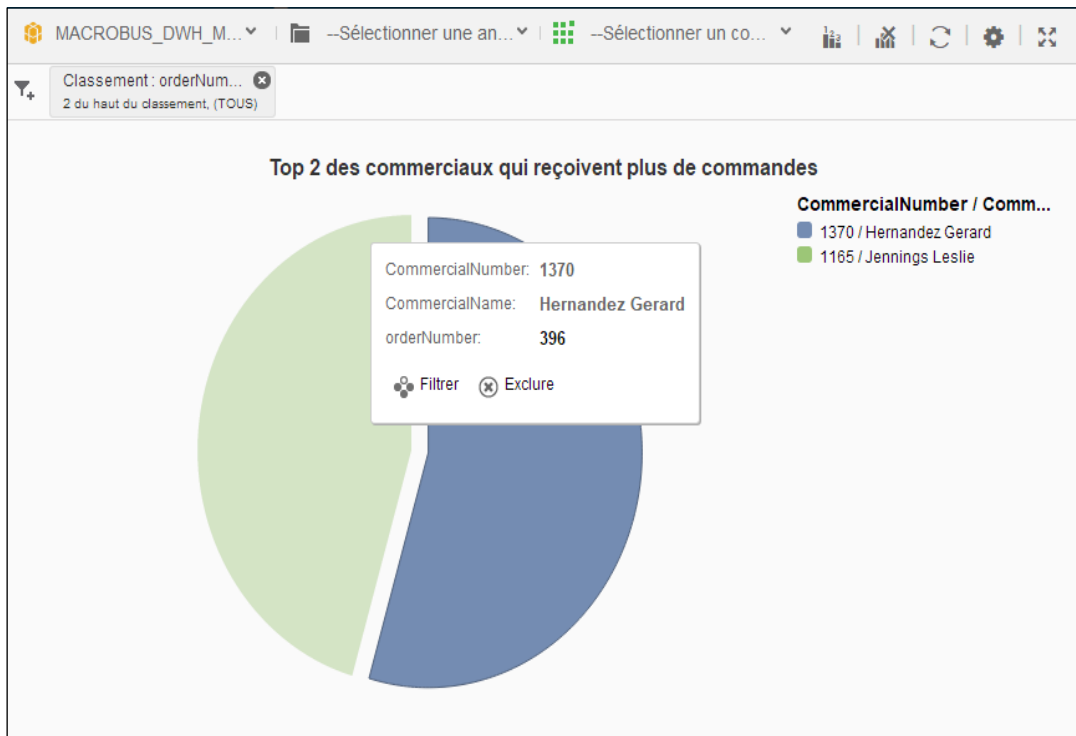
B. (SAP Predictive Analysis): Visualizations

using the tool *SAP Predictive Analysis*.

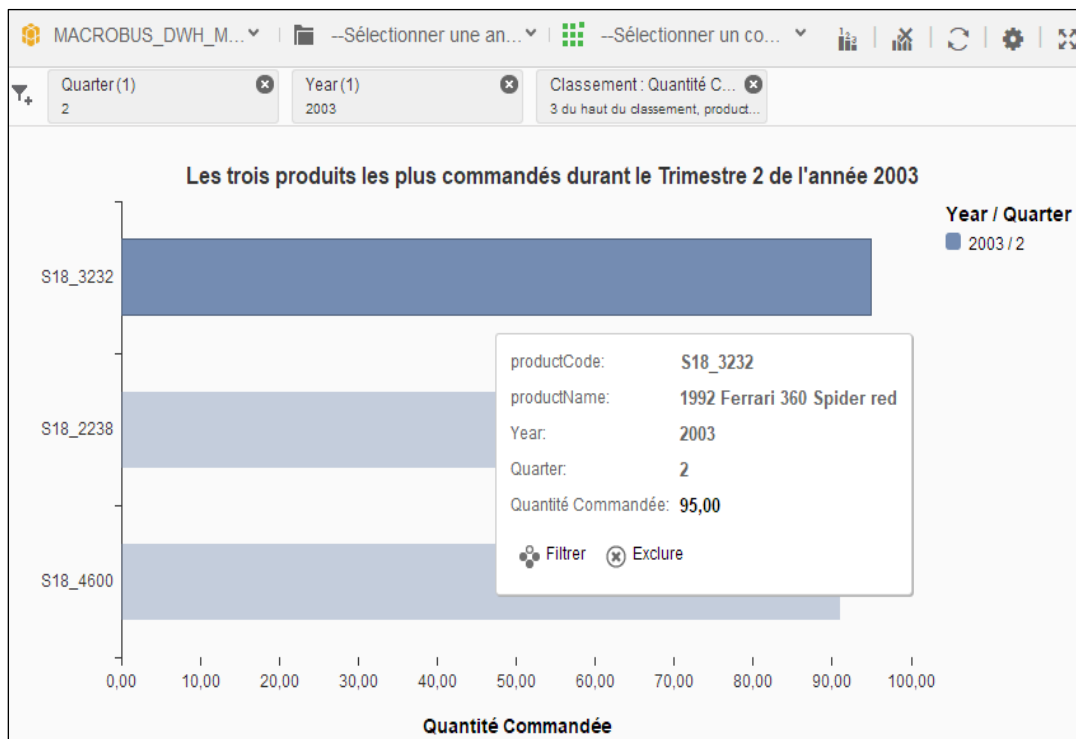
1. Number and total amount of orders per territory (Territory)



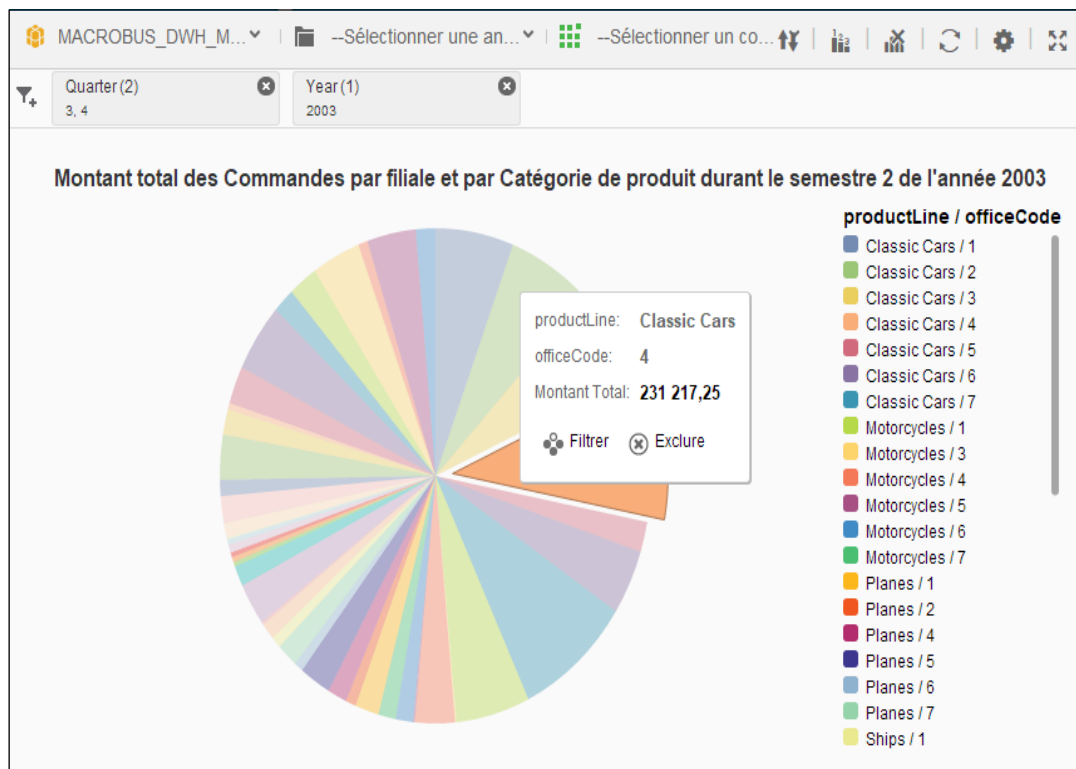
2. Top 2 sales representatives who receive the most orders.



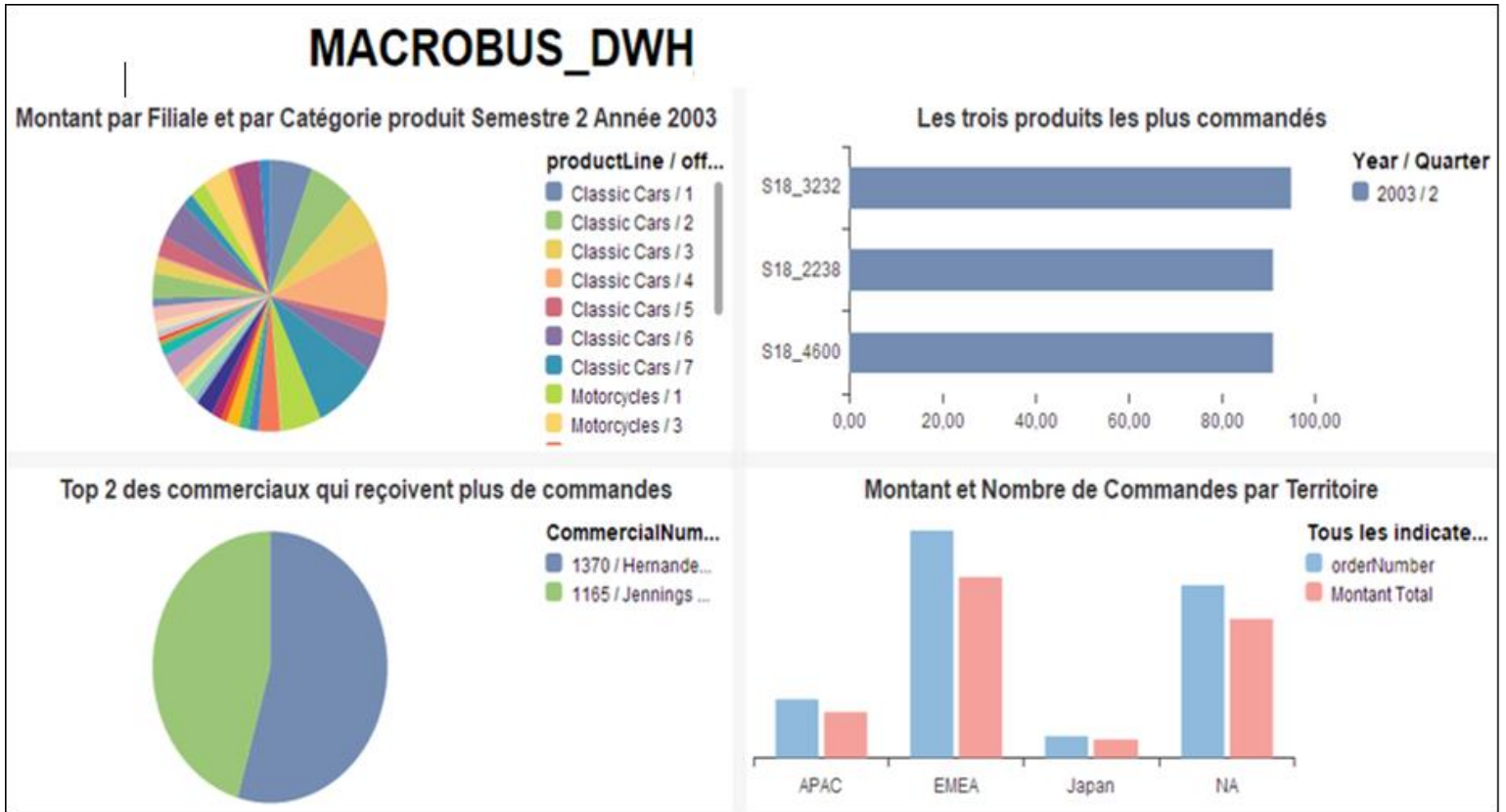
3. *The top three most ordered products during the second quarter of the year 2003.*



4. *Total amount of orders per subsidiary and per product category during the second half of 2003.*

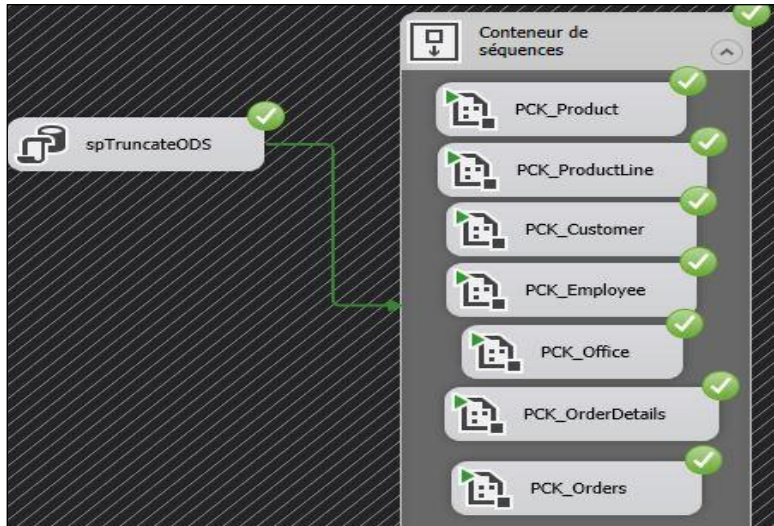


C. Summary dashboard for all four previous questions.



D. screenshots of our SSIS packages

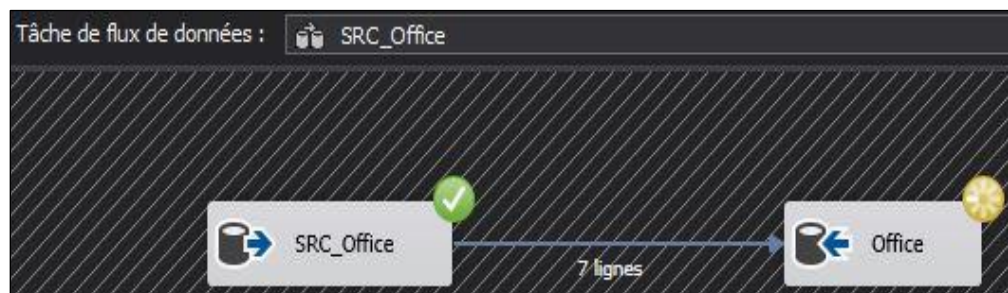
1. ODS



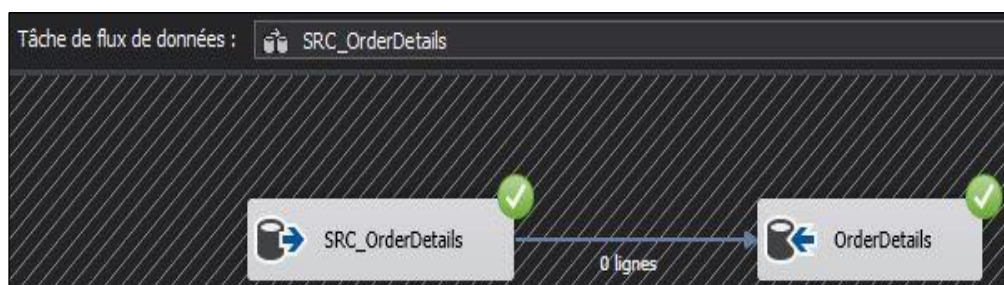
➤ SRC_Employee



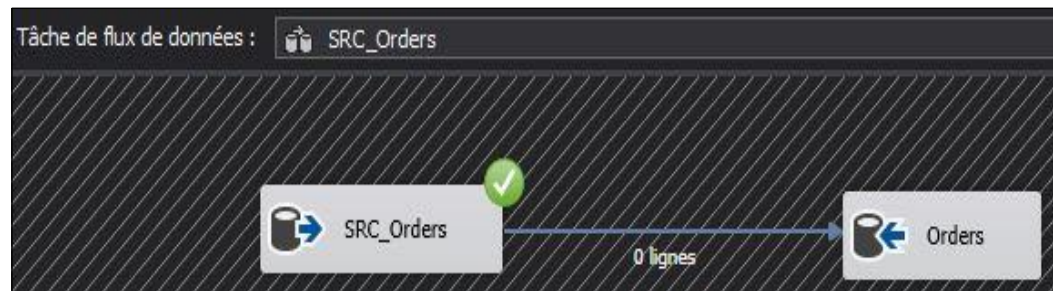
➤ SRC_Office



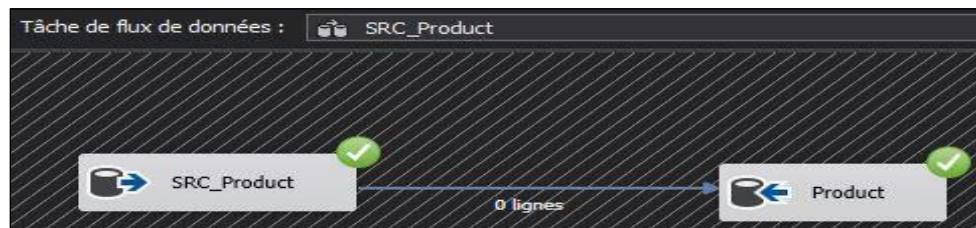
➤ SRC_OrderDetails



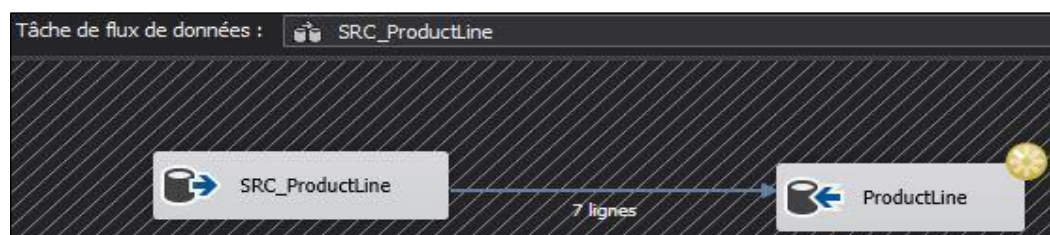
➤ SRC_Oders



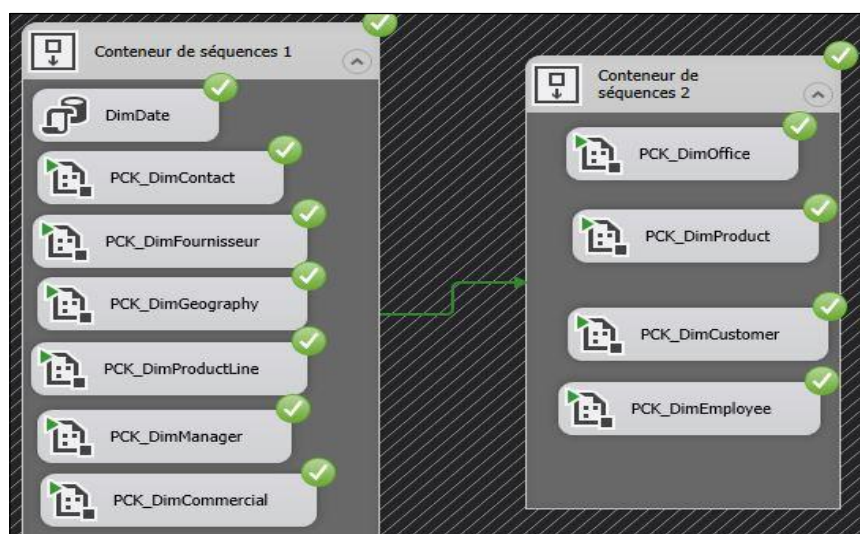
➤ SRC_Product



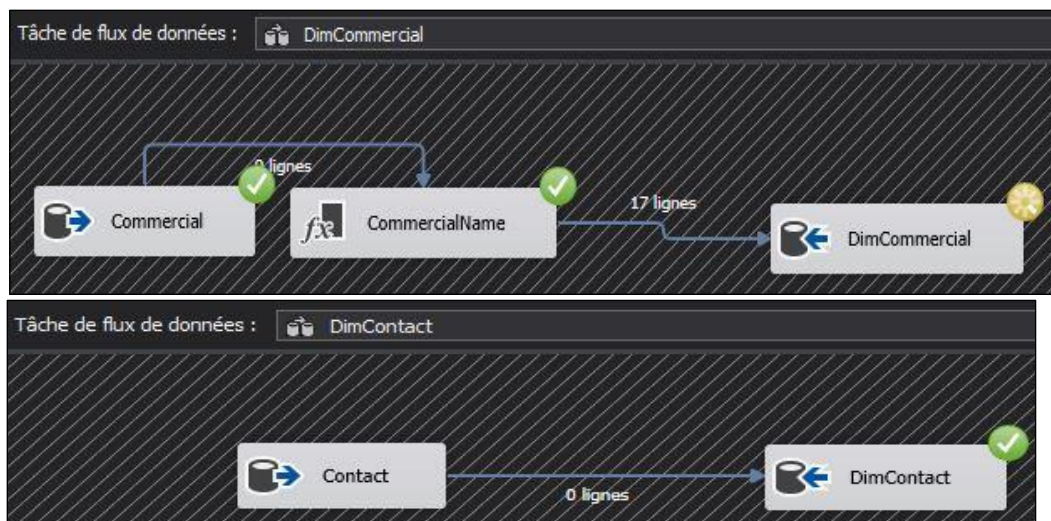
➤ SRC_ProductLine



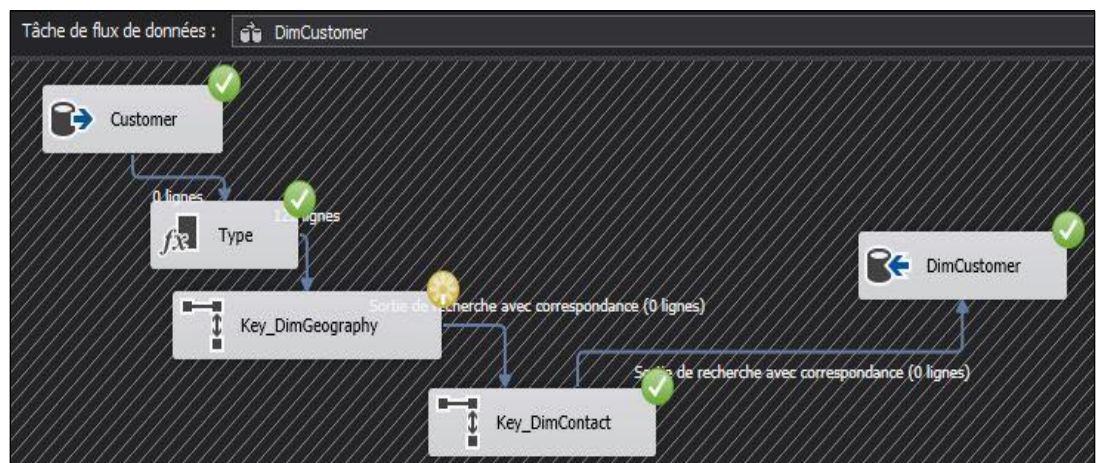
2. DIM



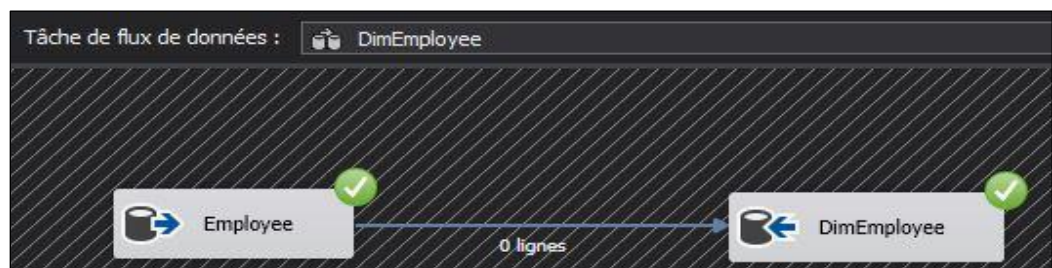
➤ DimCommercial



➤ DimCustomer



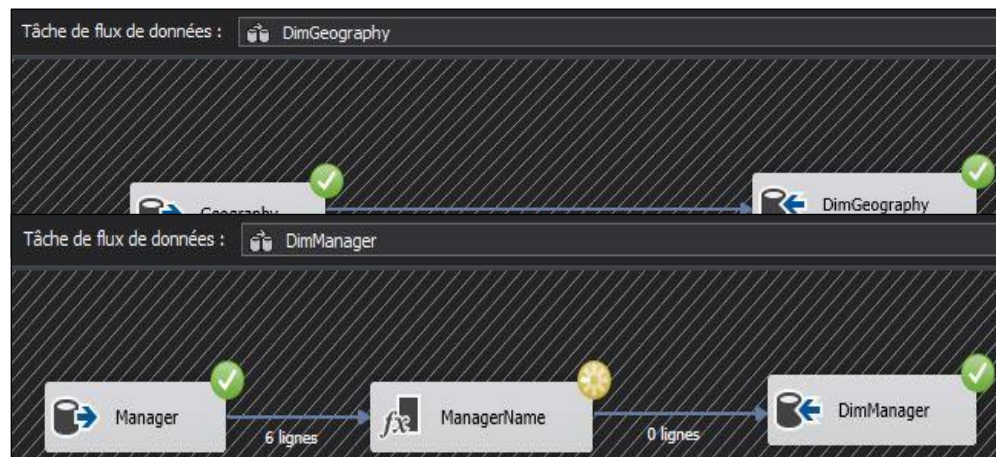
➤ DimEmployee



➤ DimFournisseur



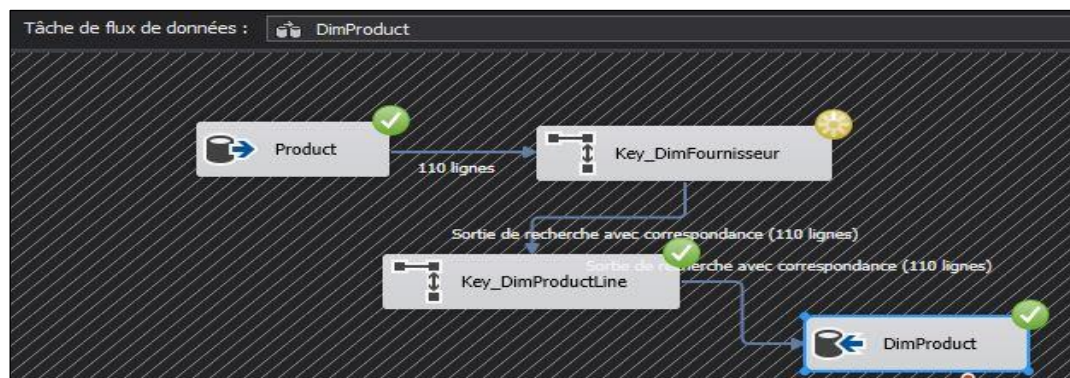
➤ DimGeography



➤ DimOffice



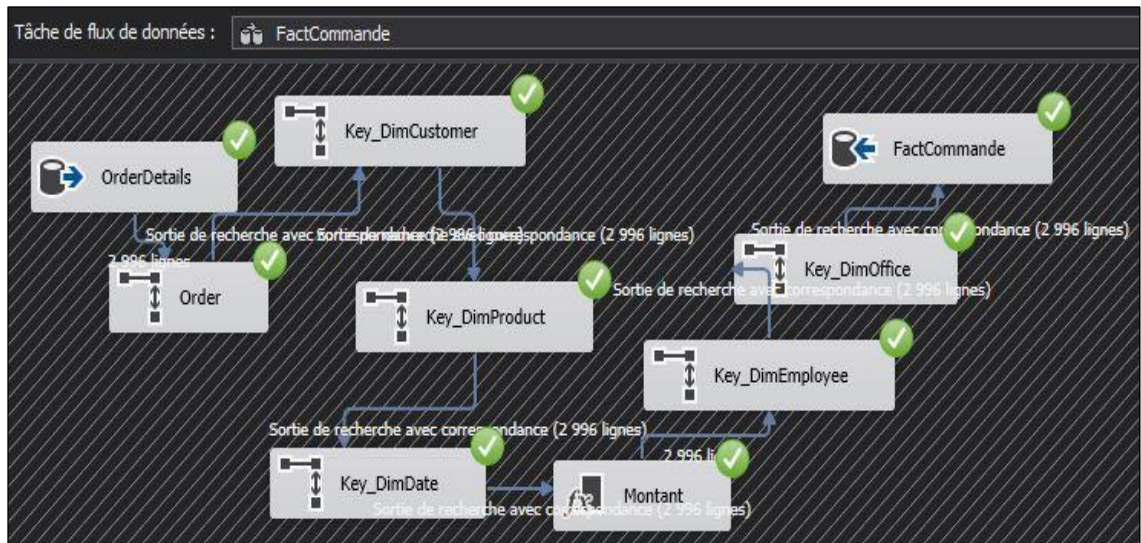
➤ DimProduct



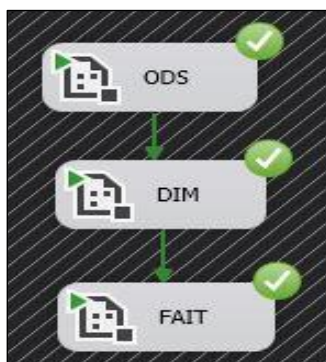
➤ DimProductLine



3. FactCommande



4. MAIN



E. Final Dashboard (Power BI)

We then realize the Dashboard below:

Orders Monitoring :: This dashboard helps to efficiently monitor Smartcar Orders

Focus

Year

All

Quarter

A

Month

Amount

Min

481.50

Max

11,503.14

3.21K

Mean

11.50K

Max

481.50

Min

Amount usd

106K

Orders by Territories

territory	Amount	Total Order
APAC	1,147,176.35	370
EMEA	4,520,712.28	1415
Japan	457,110.07	137
NA	3,479,191.91	1074
Total	9,604,190.61	2996



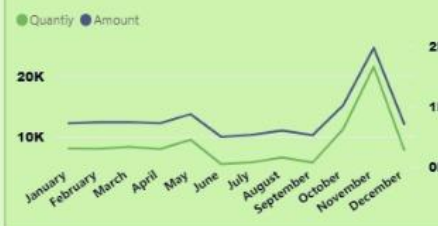
Orders by Countries



Top 2 sale representatives

Code	CommercialName	Quantity
1370	Hernandez Gerard	14231
1165	Jennings Leslie	11854
1401	Castillo Pamela	9290
Total		35375

Orders by month



CONCLUSION:

This project has provided us with a valuable opportunity to comprehensively revisit all the subjects covered during our training. Our expedition commenced by conceptualizing the “MACROBUS_DWH” data warehouse schema and subsequently extended to crafting a decision-support Dashboard. This endeavor entailed not only its full – fledged. implementation but also meticulous functional validation. Nonetheless, it is worth highlighting that the “data warehouse schema design” phase emerged as the most challenging juncture due to its pivotal significance in molding and influencing all ensuing facets of our MACROBUS project.