



BUSINESS  
SCHOOL

*2ème année Cycle Ingénieur BI*

## **Rapport du Mini Projet BI**

***Elaboré par : Chorfan Wael***

***Encadré par : Jaffel Saber***

## Sommaire :

Remerciement .....	<i>page3</i>
0) Introduction .....	<i>page3</i>
1) Problématique .....	<i>page3</i>
2) Objectifs .....	<i>page4</i>
3) Flux de travail(WorkFlow) .....	<i>page4</i>
4) Exécution :	
4.A) SSIS .....	<i>Page6</i>
4. B) SSAS .....	<i>Page16</i>
4. C) SSRS .....	<i>Page20</i>
4. D) Power BI .....	<i>page21</i>
Conclusion .....	<i>page23</i>

### **Remerciement :**

*Je tiens à remercier Monsieur Jaffel Saber pour l'expérience enrichissante et pleine d'intérêt durant les séances du cours d'informatique décisionnelle.*

## **0) Introduction**

Ce mini projet qui consiste à l'intégration et l'analyse des données de gestion des ventes d'une Entreprise de vente des produits alimentaires vient dans le but d'aider l'entreprise cible à lire ses données d'une façon pertinente et donc l'aider à la décision pour augmenter son chiffre d'affaire.

## **1) Problématique :**

Dans le cadre de ce mini projet, on dispose d'une base de données initiale « EntrepriseDB » contenant les tables concernant :

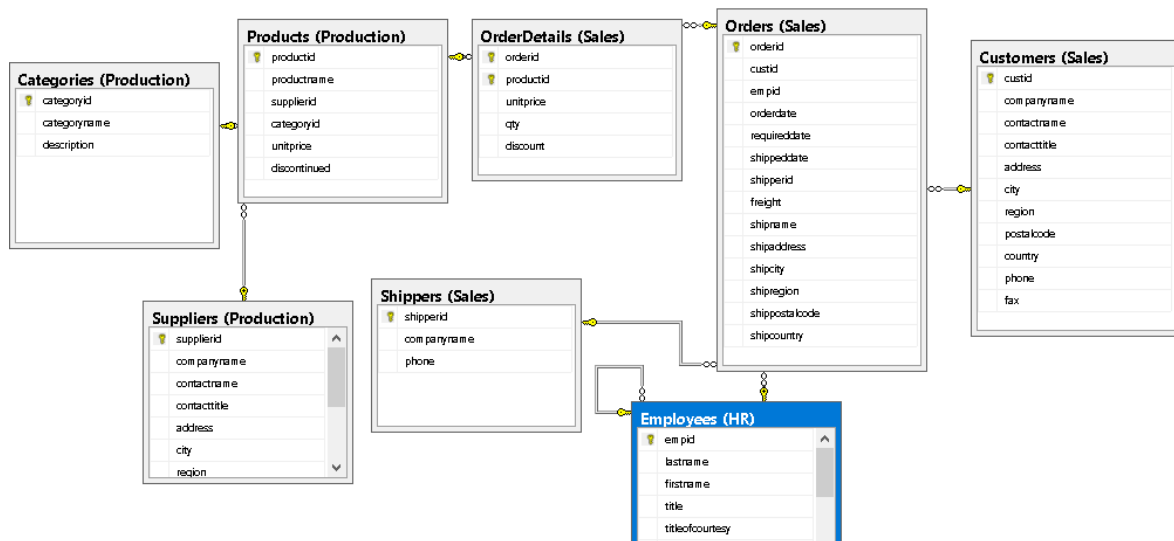
- Les Ressources Humaines,
- Les ventes,
- La production,

D'une entreprise d'approvisionnement des produits alimentaires, qui contient des données brutes dont le DE (dessin d'enregistrement) n'est pas adapté à l'analyse(SSAS)

L'entreprise veut améliorer ses décisions quant à la quantité des produits qu'elle doit fournir en termes de chiffre d'affaire.

On recourt alors aux outils **d'intégration, d'analyse** et de « **Reporting** » offertes par **SQL Server** pour pouvoir manipuler ces données à fin de les mettre sous un format pertinent pour aider les directeurs de département commercial à prendre la bonne décision correspondante lors du l'approvisionnement des produits.

Le diagramme de la base de données initiale :



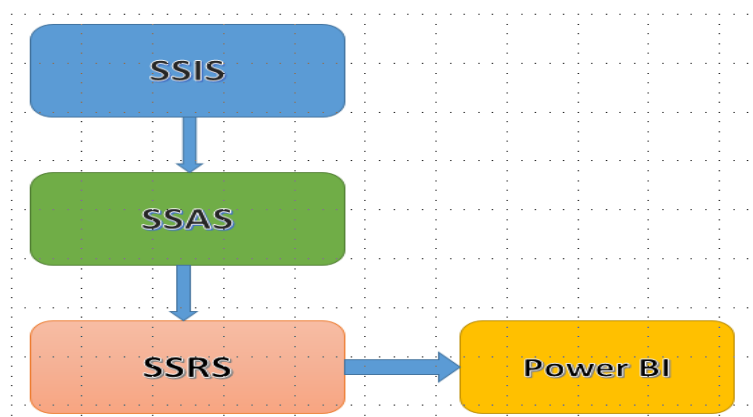
## 2) Objectifs :

Transformer les données de la BD initiale en informations pertinentes voire en connaissances.

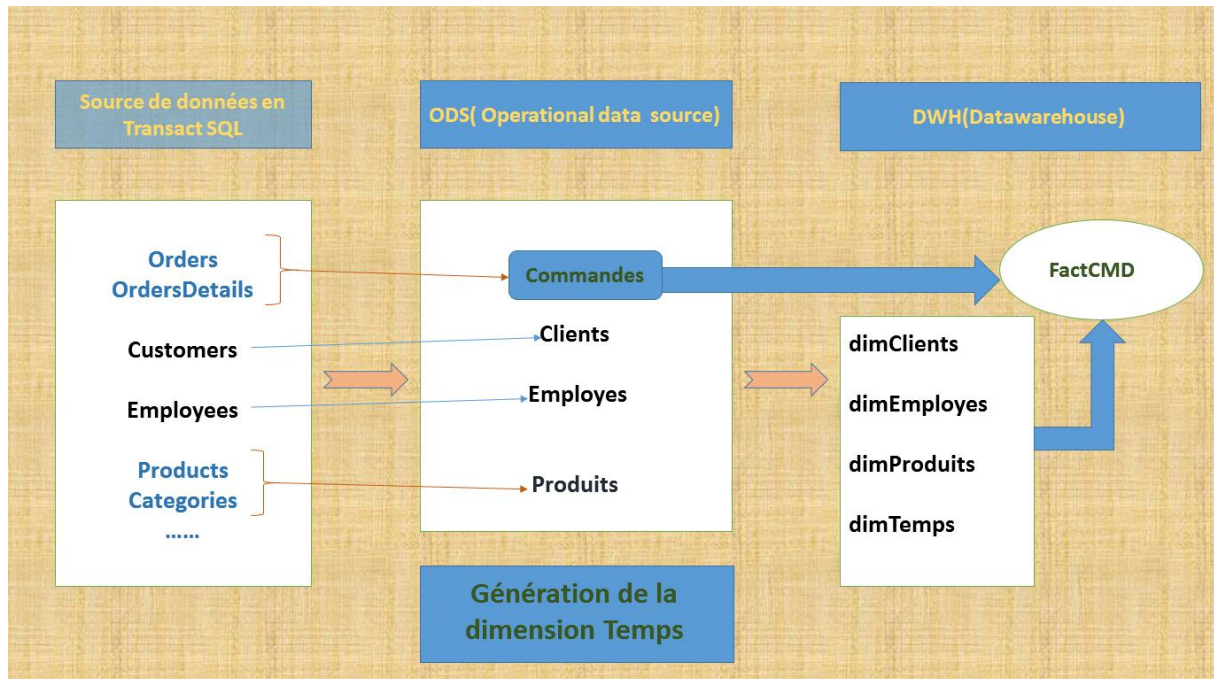
Notre objectif final est de déterminer le chiffre d'affaire par chaque Commande pour un certain Employé, un certain Client, une certaine date et un(ou plusieurs) produit(s).

## 3) Flux de travail (WorkFlow)

**Le WorkFlow Général :**



## Le WorkFlow SSIS :



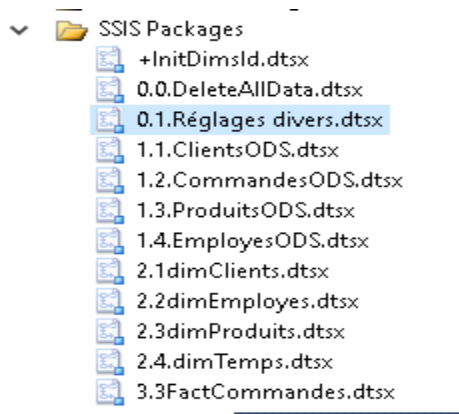
### 4) Exécution

On va s'intéresser à la gestion des ventes (voire **Commandes**), celles sont décrites par les tables **Orders**(ou **Commandes**) et **OrderDetails**, alors on va s'intéresser à intégrer les données nécessaires à partir de ces deux tables ainsi que les données liées aux **commandes** qui sont les tables suivantes :

- **Products**
- **Customers**
- **Employees**
- **Orders**
- **Order Details**

#### 4-A) SSIS:

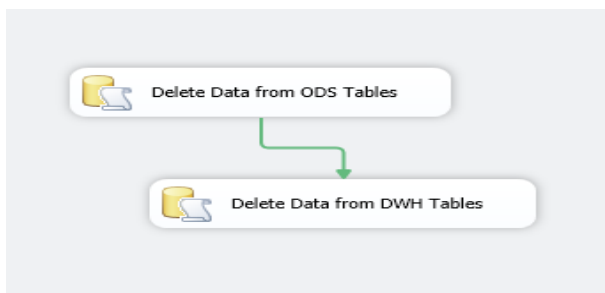
La création de plusieurs « packages.dtsx » nous permet de diviser les tâches et être plus efficace.



#### La description et l'explication des packages :

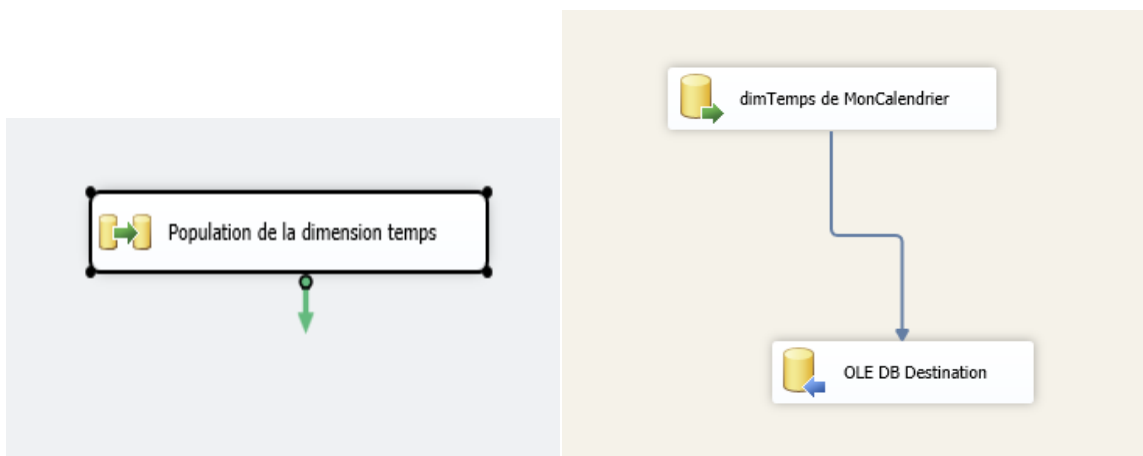
##### 0.0. DeleteAllData :

Permet à effacer tous les données dans les 3 bases.



##### 0.1. Réglages divers :

Ce package est consacré à tous les opérations aléatoires qui consistent au réglage et nettoyage des champs spécifiques qu'on désire mettre sous un certain format, ainsi que remplir les données dans la table dimension temps,



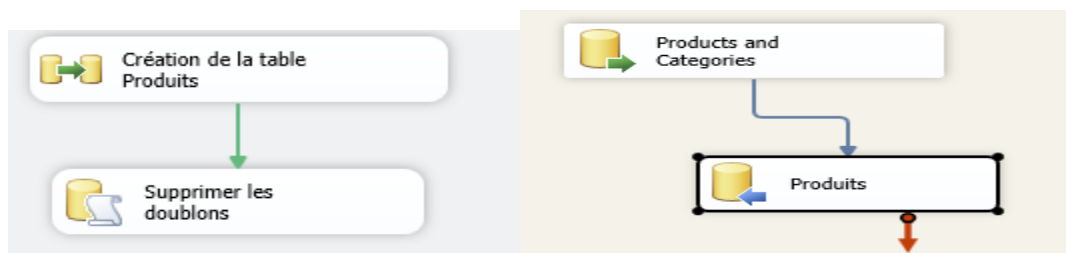
### 0.1.ClientsODS

Permet d'intégrer les données de la table « Customers » dans base « EntrepriseDB » vers la table « Clients » dans la base « ODS » avec la suppression des doublons.

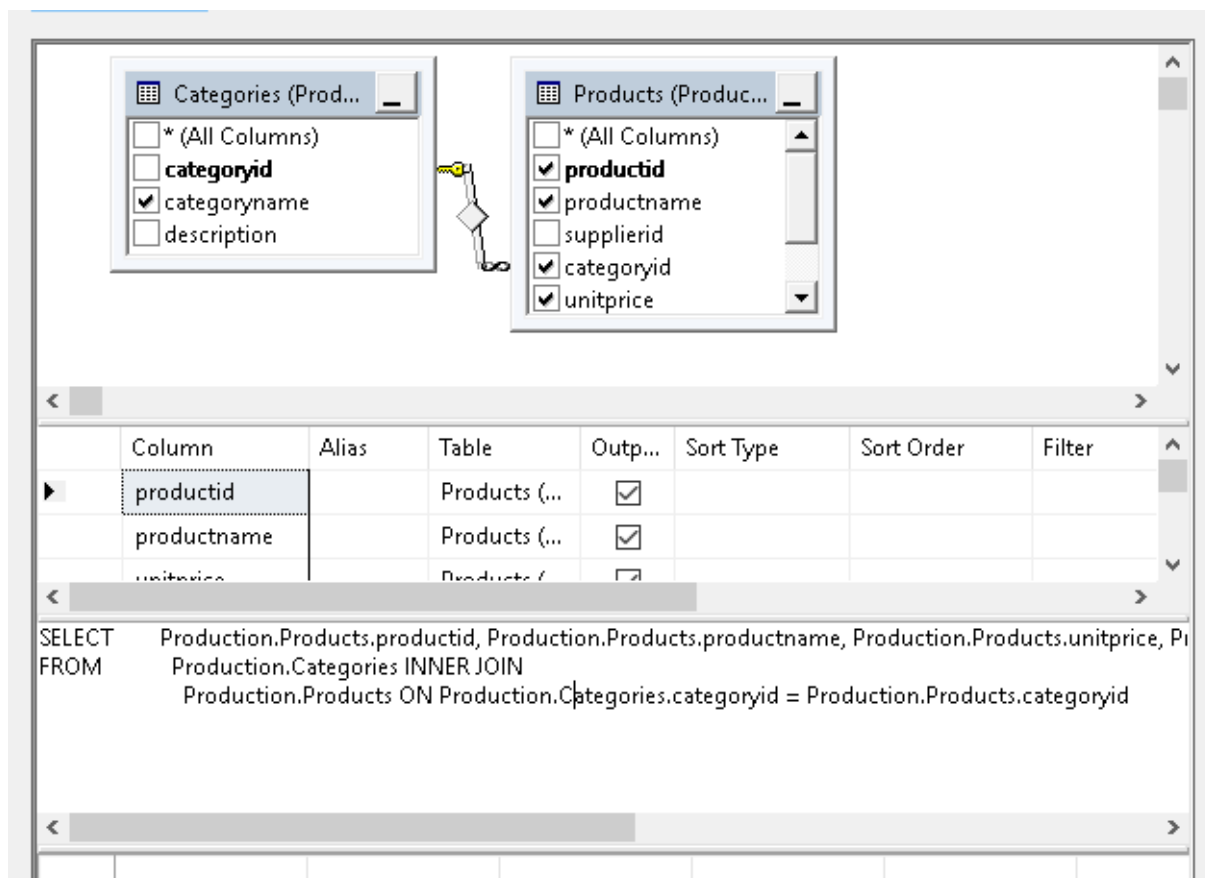


### 1.2.CommandesODS

Permet d'intégrer les données des tables « Order » et « OrderDetails » dans la tables« Commandes » dans la base « ODS » avec la suppression des doublons.



Lors de l'importation des données, on a ajouté une colonne de chiffre d'affaire « CaCde » en sélectionnant les champs nécessaires permettant de le calculer.



### 1.3.ProduitsODS

Permet d'intégrer les données de la table « Products » et « Categories » dans la base « EntrepriseDB » vers la table « Produits » dans la base « ODS » avec la suppression des doublons.

(idem pour 1.1)

### 1.3.EmloyesODS

Permet d'intégrer les données de la table « Employees » dans la base « EntrepriseDB » vers la table « Employes » dans la base « ODS » avec la suppression des doublons.

(idem pour 1.1)



On dispose maintenant de la base «ODS », et on va créer les dimensions dans la base « DWH » :

## 2.1dimClients

### Etape 1 :

OLE DB Source à partir de la table « Clients » de la base « ODS » :

OLE DB connection manager:

LocalHost.ODS

Data access mode:

Table or view

Name of the table or the view:

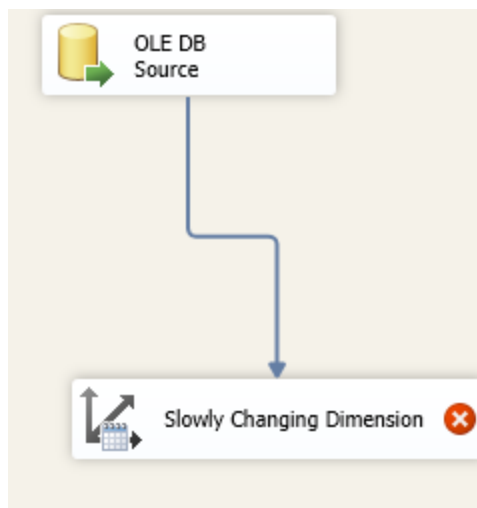
[dbo].[Clients]

Query result (up to the first 200 rows):

CodeClt	nomClt	nomContact	fonct
1	Customer ...	Allen, Mic...	Sales
2	Customer ...	Hassall, M...	Own
3	Customer ...	Peoples, J...	Own
4	Customer ...	Arndt, To...	Sales
5	Customer ...	Higginbot...	Orde
6	Customer ...	Poland, C...	Sales
7	Customer ...	Bansal, D...	Mark
8	Customer ...	Ilyina, Julia	Own
9	Customer ...	Raghav, ...	Own
10	Customer ...	Bassols, P...	Acco
11	Customer ...	Jaffe, David	Sales
12	Customer ...	Ray, Mike	Sales
13	Customer ...	Benito, Al...	Mark
14	Customer ...	Jelitto, Ja...	Own
15	Customer ...	Richardso...	Sales

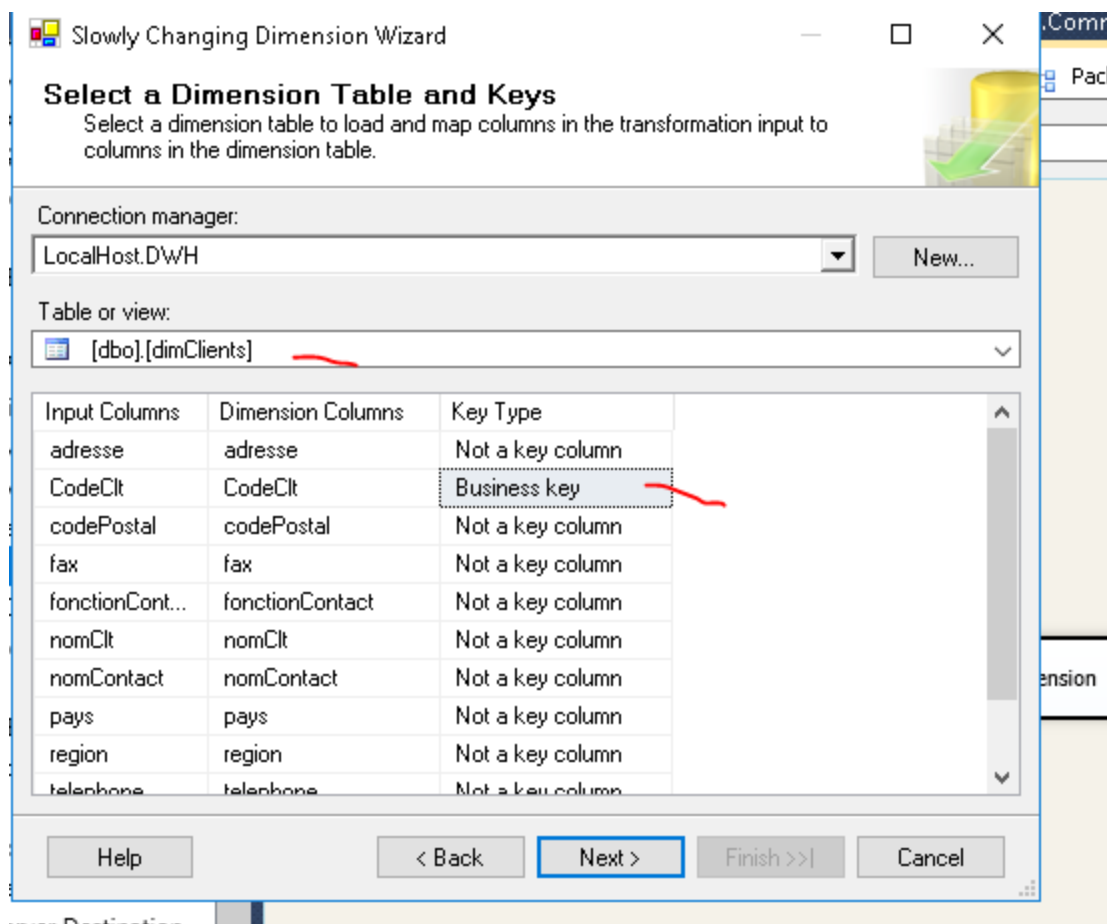
### Etape 2 :

On fait la liaison avec un composant « [Slowly Changing Dimension](#) » :



### Etape 3 :

On choisit la table « Clients » et le Business Key qui est le CodeClt (code de client) en précisant le type des données si elles sont fixes (par exemple nom ,prénom ..) ou qu'elles peuvent changer (fax,pays, telephone ,ville ...)



**Slowly Changing Dimension Columns**  
Manage the changes to column data in your slowly changing dimensions by setting the change type for dimension columns.

**Fixed Attribute**  
Select this type when the value in a column should not change. Changes are treated as errors.

**Changing Attribute**  
Select this type when changed values should overwrite existing values. This is a Type 1 change.

**Historical Attribute**  
Select this type when changes in column values are saved in new records. Previous values are saved in records marked as outdated. This is a Type 2 change.

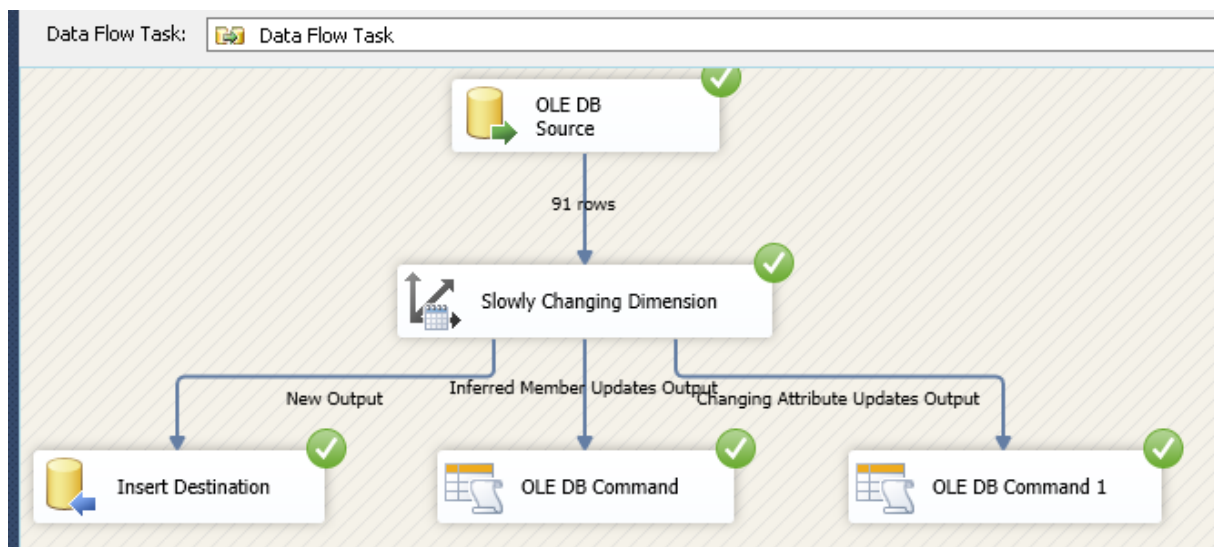
Select a change type for slowly changing dimension columns:

Dimension Columns	Change Type
adresse	Changing at...
codePostal	Changing at...
fax	Changing at...
nomClt	Fixed attribute
fonctionContact	Changing at...
nomContact	Fixed attribute
pays	Changing at...
region	Changing at...
telephone	Changing at...
ville	Changing at...

Remove

Help < Back Next > Finish >>| Cancel

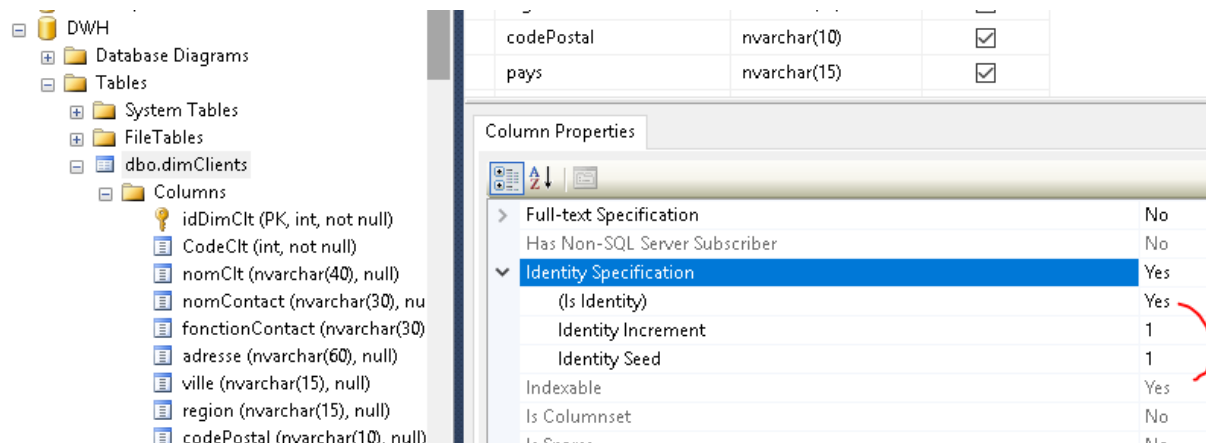
Et on obtient ce schéma :



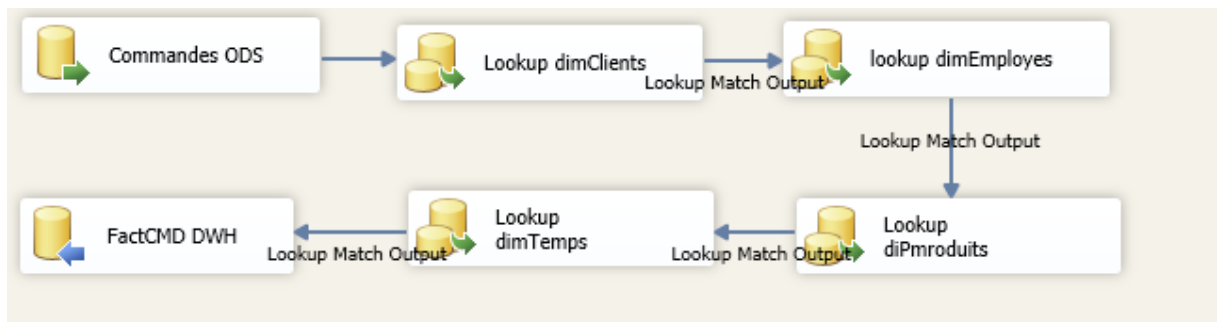
Les mêmes étapes se répètent pour les autres dimensions dans les packages (2.3.dimProduits, 2.2.dimEmployes et 2.4.dimTemps) [2.3. DimProduits](#) || [2.2.dimEmployes](#) || [2.4.dimTemps](#).

On obtient les données nécessaires qui nous permettent de construire notre « DataWarehouse », mais on doit d'abord ajouter des Id(Identifiants) pour chaque table pour la préparer à l'analyse .

Les ID sont ajoutées manuellement dans la base DWH en mettant la propriété « identity » à « yes » )

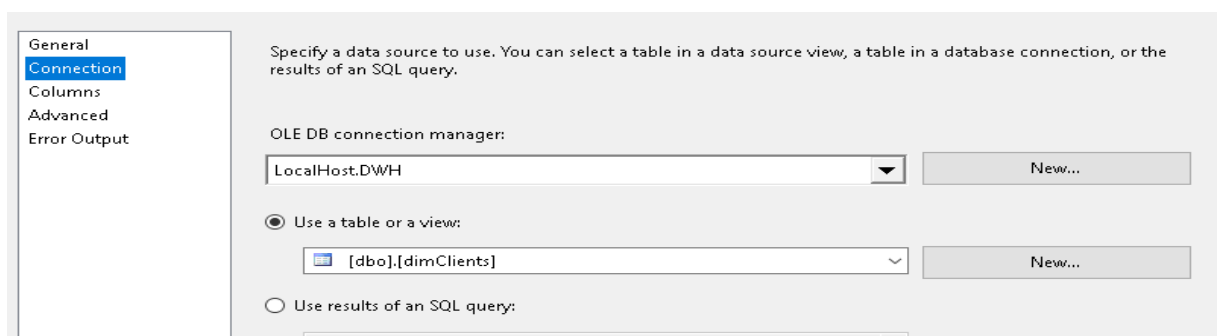


On revient sur l'SSIS dans le package 3.3 et on fait la liaison de l'OLE DB destination à des composants « Lookup » ( en cascade) afin de chercher les éléments par leur ID et les intégrer dans la table « FactCMD » à l'aide d'un « OLE DB Destination »

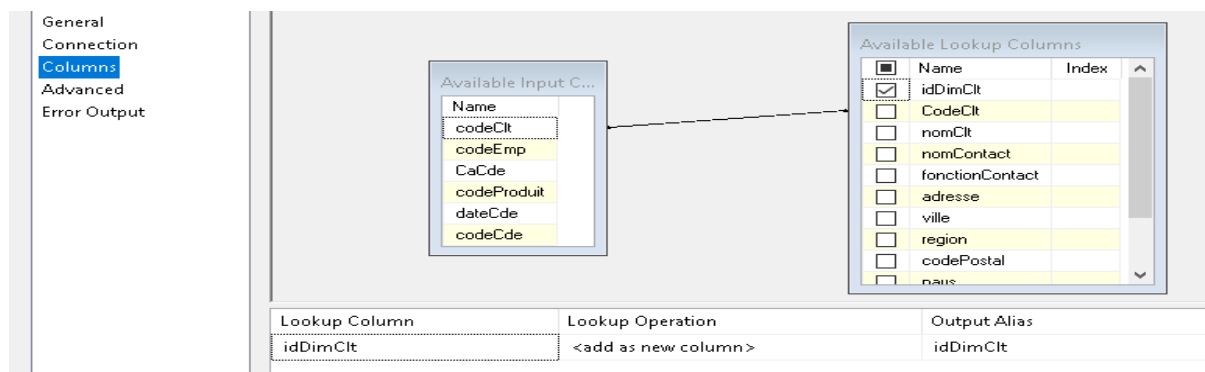


Tous les éléments « look up » ont le même fonctionnement, c'est sélectionner chaque ligne par son Id pour construire des hiérarchies et les classer par les jointures définies tel que (le CodeClt ,le CodeEmp, le CodeProduit et le CodeCde).

Par exemple pour le composant qui sélectionne les colonnes de la table « dimClients » :



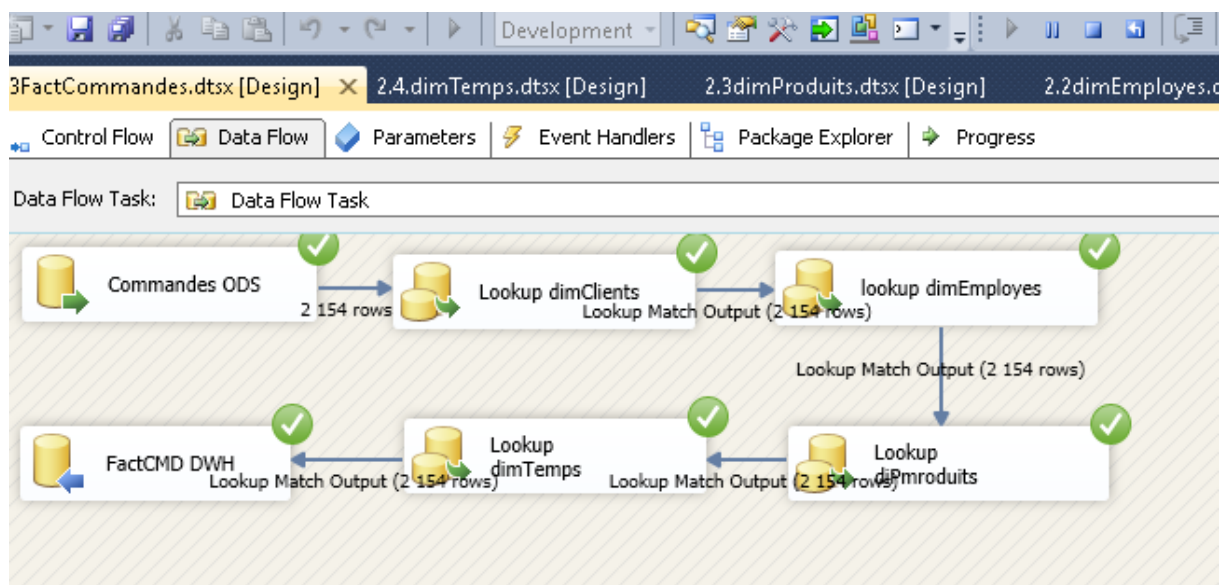
On coche l'idDimClt et on lie le codeClt de la table « Commandes » à son analogue dans la table « dimClients »



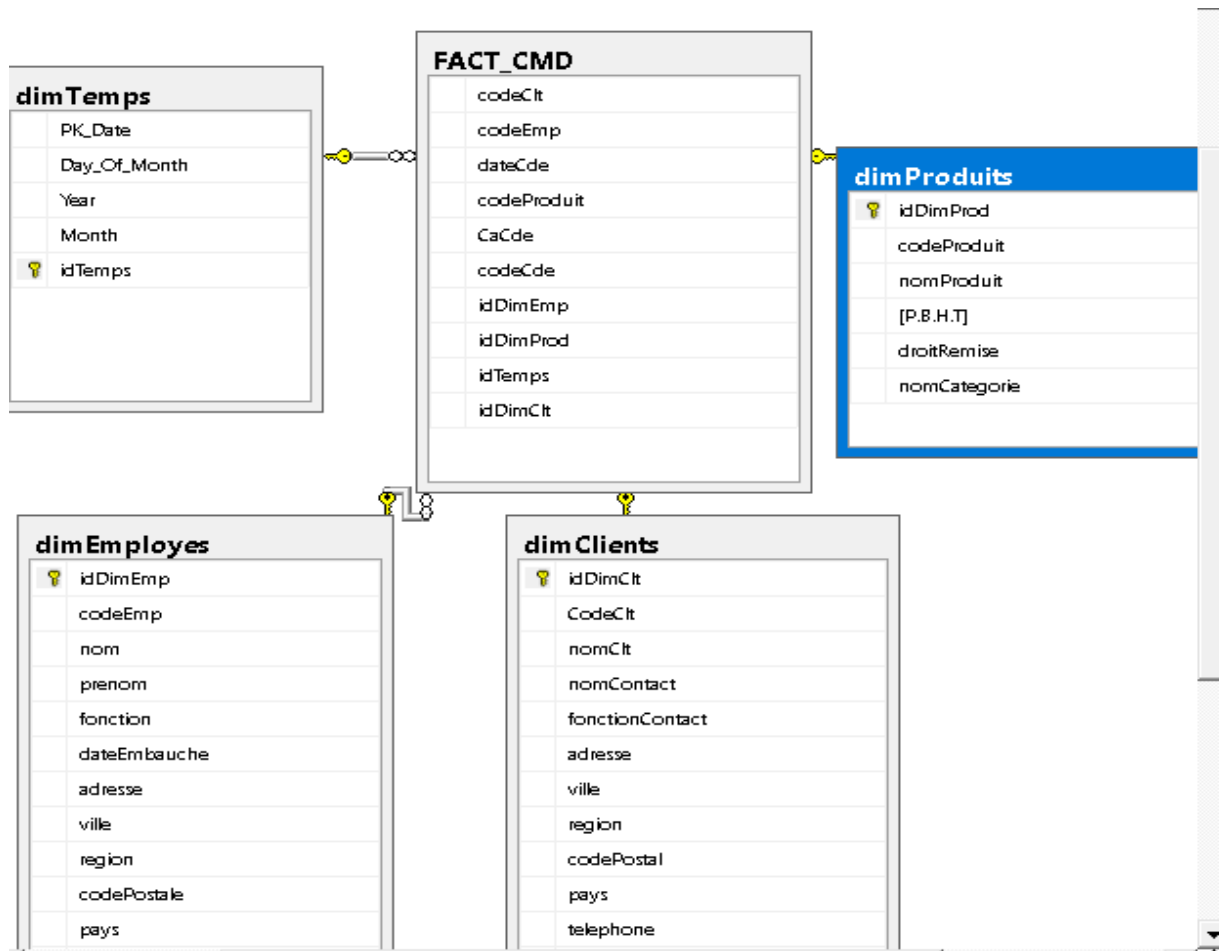
Oubien « drag&drop » (pour le « mapping ») :



On exécute le package 3.3.FactCommandes :



Et on ajoute les liaisons des clés étrangères à la table « FactCMD » avec les autres tables pour obtenir un entrepôt de données prêt à l'**analyse** selon le chiffre d'affaire (CaCde) de chaque commande.



Avant l'analyse on va vérifier que les données sont importées et déployés convenablement :

On va alors observer une ligne de Commande, par exemple dans la base « EntrepriseDB » on a pour la commande possédant le codeCde(10251) les données suivantes :

Dans la table « Orders »

CodeClt = 84 ; codeEmp=3 ; dateCde=2006-07-08

	orderid	custid	empid	orderdate	requireddate	shippeddate
1	10248	85	5	2006-07-04 00:00:00.000	2006-08-01 00:00:00.000	2006-07-16 00:00:00.000
2	10249	79	6	2006-07-05 00:00:00.000	2006-08-16 00:00:00.000	2006-07-10 00:00:00.000
3	10250	34	4	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-12 00:00:00.000
4	10251	84	3	2006-07-08 00:00:00.000	2006-08-05 00:00:00.000	2006-07-15 00:00:00.000
5	10252	76	4	2006-07-09 00:00:00.000	2006-08-06 00:00:00.000	2006-07-11 00:00:00.000

Et dans la table « OrderDetails » :

Cette commande a 3 produits :

	orderid	productid	unitprice	qty	discount
7	10250	51	42.40	35	0.150
8	10250	65	16.80	15	0.150
9	10251	22	16.80	6	0.050
10	10251	57	15.60	15	0.050
11	10251	65	16.80	20	0.000
12	10252	20	64.80	40	0.050
13	10252	33	2.00	25	0.050

codeProd = 22, P.U.H.T=16.80, Qte=6, Remise =0.050

codeProd = 57, P.U.H.T=15.60, Qte=15, Remise =0.050

codeProd = 65, P.U.H.T=16.80, Qte=20 , Remise =0.000

Le calcul du chiffre d'affaire pour cette demande sous la formule utilisé dans le « Query builder » lors de la création de la table commandes ( $Qte * P.U.H.T * (1 - Remise)$ ) donne un chiffre d'affaire égal à :

CaCde = CaCde 1+ CaCde 2+CaCde 3

$$= 16.80 * 6 * (1 - 0.050) + 15.60 * 15 * (1 - 0.050) + 16.80 * 20 * (1 - 0.000)$$

$$= 95.76 + 222.3 + 336$$

$$\text{CaCde} = 654.06$$

Dans le « Browser » de SSAS on met dans les dimensions pour l'analyse : codeClt = 84 ; codeEmp=3 ; dateCde=2006-07-08 ; codeProduit =22,57 et 65

$$\Rightarrow \text{CaCde} = 654.06$$

Dimension	Hierarchy	Operator	Filter Expression	Parame
Dim Clients	Id Dim Clt	Equal	{ 84 }	<input type="checkbox"/>
Dim Employes	Id Dim Emp	Equal	{ 3 }	<input type="checkbox"/>
Dim Produits	Id Dim Prod	Equal	{ 65, 22, 57 }	<input type="checkbox"/>
Dim Temps	Id Temps	Equal	{ 20060708 }	<input type="checkbox"/>

Ca Cde

654.06

Pour la commande au client possédant le code 84 qui consiste au produits avec les codes 22,57 et 65 livrés par l'employé possédant le code 3 à la date 2006-07-08 :

le chiffre d'affaire est :654.06

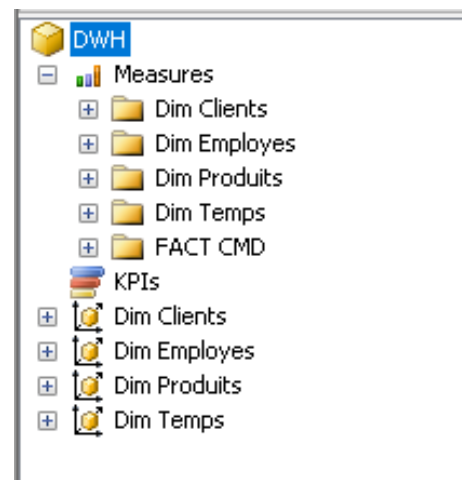
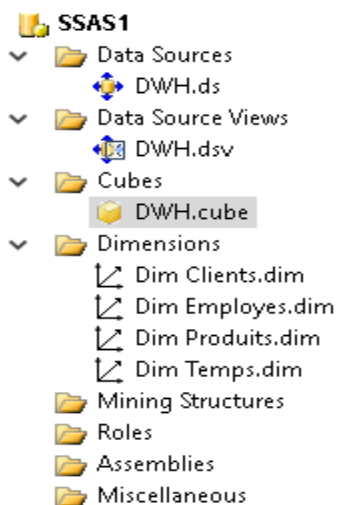
#### 4-B) SSAS:

Après avoir créé notre entrepôt de données et avoir vérifié l'intégration dans la base « DWH », on va les analyser dans la partie SSAS.

On va créer 2 « KPI » :

- 1.Le chiffre d'affaire total d'un client.
- 2.Le chiffre d'affaire total d'un produit.

La création et le déploiement du cube OLAP sont effectués avec succès.

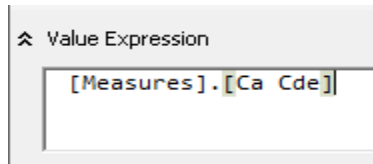




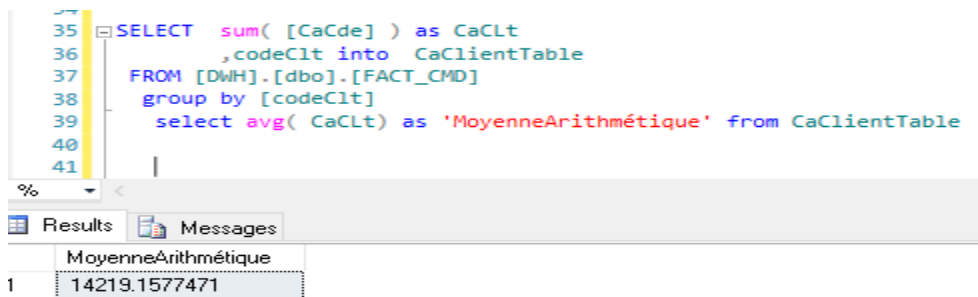
## KPI1

On va créer le KPI1 : Chiffre d'affaire totale par client

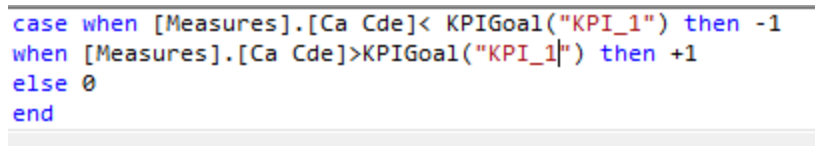
O Value : Le chiffre d'affaire






O Goal = La moyenne arithmétique des chiffres d'affaire par client. (14219.1577471)



O Status



Le chiffre d'affaire totale pour le client numéro 3 il est inférieur à 14219.15 :

Dimension	Hierarchy	Operator	Filter Expression			
Dim Clients	 Id Dim Clt	Equal	{3}			
<div>&lt;Select dimensions&gt;</div>						
<div>&lt;</div>						
Display Structure		Value	Goal	Status	Trend	Weight
 KPI_1		7023.98	14219.15			

Le chiffre d'affaire totale pour le client numéro 7 il est supérieur à 14219.15 :

Dim Clients	Id Dim Clt	Operator	Filter Expression
		Equal	{7}
<Select dimensions>			
<			
Display Structure	Value	Goal	Status
KPI_1	18534.08	14219.15	😊

Sur Excel :

1	Row Labels	KPI_1 Goal	KPI_1 Status	KPI_2
2	1	14219.15	🔴	4273
3	2	14219.15	🔴	1402.95
4	3	14219.15	🔴	7023.9775
5	4	14219.15	🔴	13390.65
6	5	14219.15	🟢	24927.5775
7	6	14219.15	🔴	3239.8
8	7	14219.15	🟢	18534.08
9	8	14219.15	🔴	4232.85
10	9	14219.15	🟢	21963.2525

### KPI2 :

On va créer le KPI2 : Chiffre d'affaire par produit.

O Value = Le chiffre d'affaire.

Value Expression

[Measures].[Ca Cde]

```

3 SELECT sum( [CaCde] ) as CaProd
4         ,codeProduit into CaProdTable
5 FROM [DWH].[dbo].[FACT_CMD]
6 group by [codeProduit]
7
8 select * from CaProdTable

```

	CaProd	codeProduit
1	4313.70000000	23
2	5883.00000000	46
3	21942.36000000	69
4	1784.82500000	15
5	3044.00000000	3
6	24900.13000000	72
7	19849.14450000	26
8	7137.00000000	6

O Goal = La moyenne arithmétique du chiffres d'affaire par produit .(d'une autre façon)

```

13 declare @S int
14 SELECT @S=(sum( [CaCde] ))
15 FROM [DWH].[dbo].[FACT_CMD]
16 --print 'la somme des chiffres d'affaire est : '+ convert(varchar(8), @S)
17
18 declare @Cnt int
19 SELECT @Cnt= count ([codeProduit]) FROM [DWH].[dbo].[dimProduits]
20 --print 'l nombre de produits est : '+ convert(varchar(8), @Cnt)
21
22 print 'la moyenne des chiffres d'affaire par produit est : '+ convert(varchar(8), @S/@Cnt)

```

91 %

Messages

la moyenne des chiffres d'affaire par produit est : 16435

## O Status

```
status expression:
case when [Measures].[Ca Cde]< KPIGoal("KPI_2") then -1
when [Measures].[Ca Cde]>KPIGoal("KPI_2") then +1
else 0
end
```

Le chiffre d'affaire de produit numéro 6 est inférieur à 16435 :

Dimension	Hierarchy	Operator	Filter Expression
Dim Produits	Id Dim Prod	Equal	{6}
< Select dimensions			
<			
Display Structure	Value	Goal	Status
KPI_2	7137	16435	☹️

Le chiffre d'affaire de produit numéro 7 est supérieur à 16435 :

Dimension	Hierarchy	Operator	Filter Expression
Dim Produits	Id Dim Prod	Equal	{7}
< Select dimensions			
<			
Display Structure	Value	Goal	Status
KPI_2	22044.3	16435	😊

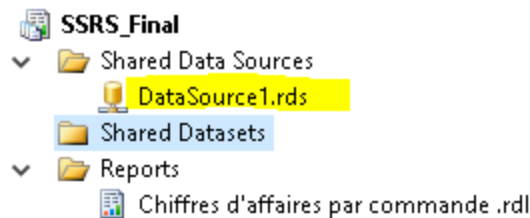
Sur Excel :

1	Row Labels	KPI_2	KPI_2 Goal	KPI_2 Status
2	1	12788.1	16435	🔴
3	2	16355.96	16435	🔴
4	3	3044	16435	🔴
5	4	8567.9	16435	🔴
6	5	5347.2	16435	🔴
7	6	7137	16435	🔴
8	7	22044.3	16435	🟢
9	8	12772	16435	🔴

## 4-C) SSRS :

Cette partie consiste à générer les rapports à partir de la base de données précédente.

On ajoute une « shared datasource » pour générer les différents rapports qu'on veut sans recréer de nouveau une « datasource » pour chaque rapport.



**Rapport 1 :** Qui consiste à reporter le chiffre d'affaire pour chaque client.

*Remarque : Dans le « query builder » on doit faire la sommation des chiffres d'affaire pour chaque client.*



### Design the Query

Specify a query to execute to get the data

Use a query builder to design your query.

Query Builder...

Query string:

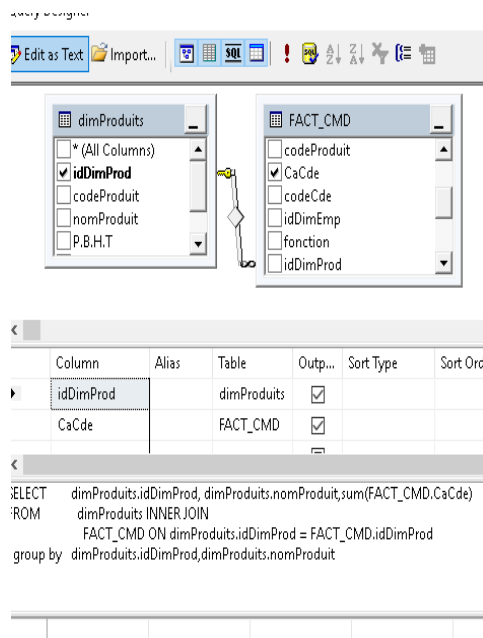
```
SELECT sum( [CaCde] ) as CaProd  
      ,codeClt  
FROM [DWH].[dbo].[FACT_CMD]  
group by codeClt
```

Help < Back

Ca Prod	code Clt
11666.9000000	23
16076.6000000	46
1467.2900000	69
3810.7500000	15
27363.6050000	89
7023.9775000	3
16215.3250000	72
3172.1600000	26
3239.8000000	6
7176.2150000	49
836.7000000	29
11441.6300000	75

**Rapport 2 :** Qui consiste à reporter le chiffre d'affaire pour chaque produit.

*Remarque : Dans le « Query builder » on doit faire la sommation des chiffres d'affaire pour chaque produit et on a ajouté la colonne nom Produit pour observer directement le nom du produit avec le chiffre d'affaire correspondant.*



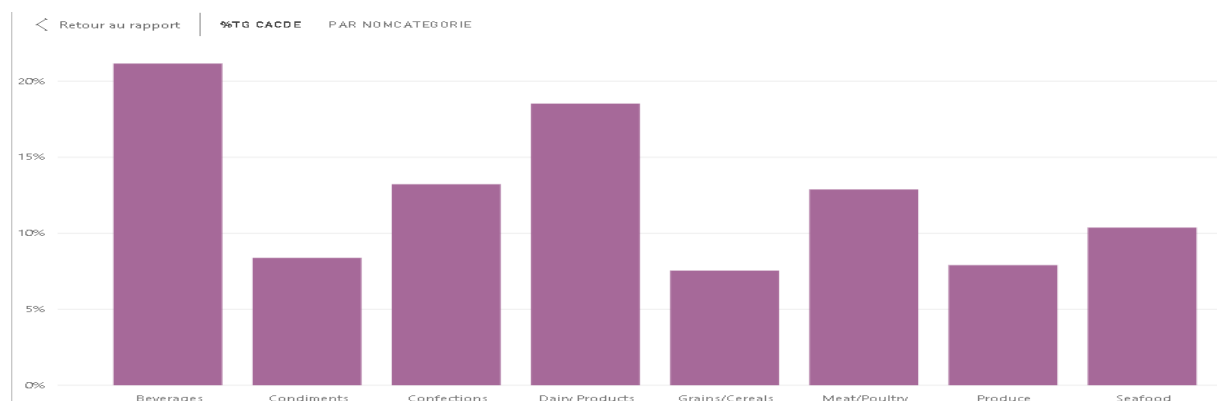
**Chiffre d'affaire par pr**

id Dim Prod	nom Produit	Ca Produit
23	Product JUDZ	4313.70000000
46	Product CBRRL	5883.00000000
69	Product COAXA	21942.36000000
15	Product KSZOI	1784.82500000
3	Product IMEHJ	3044.00000000
72	Product GEEEO	24900.13000000
26	Product HLGZA	19849.14450000
6	Product VAIIV	7137.00000000

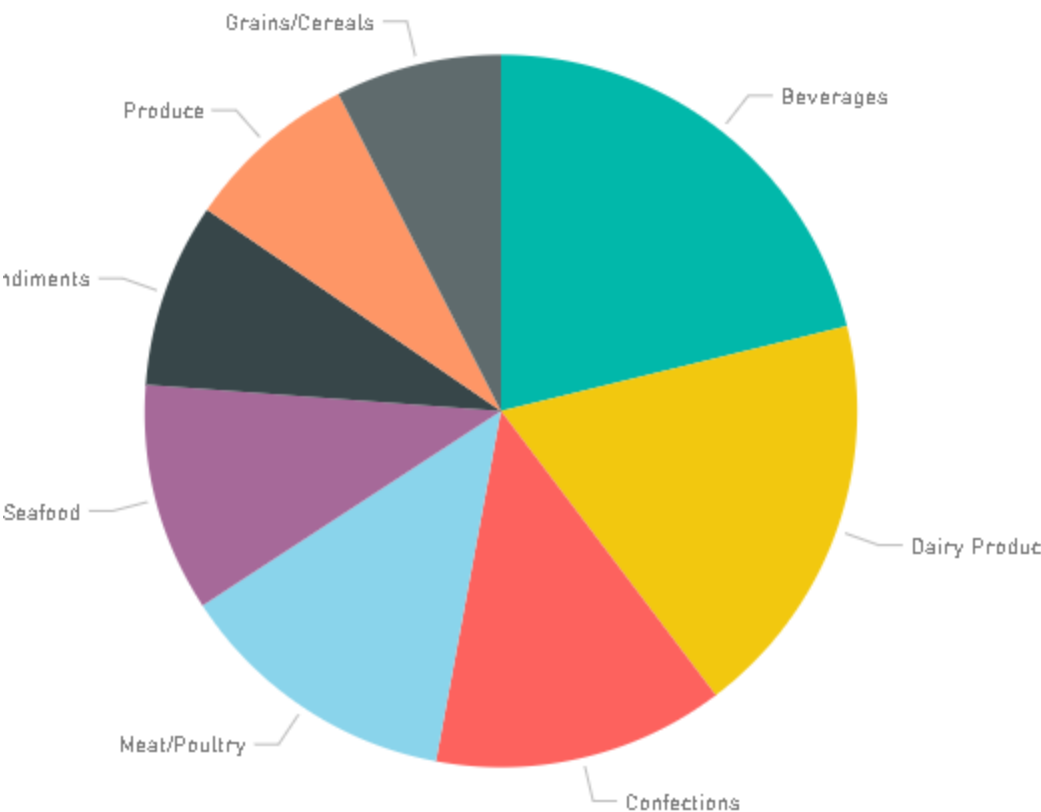
## 4-D) Power BI :

L'outil SSRS est limité par rapport au Power BI, ce dernier donne plus de possibilités ainsi que des graphes et différentes formes mieux significatifs et attractifs.

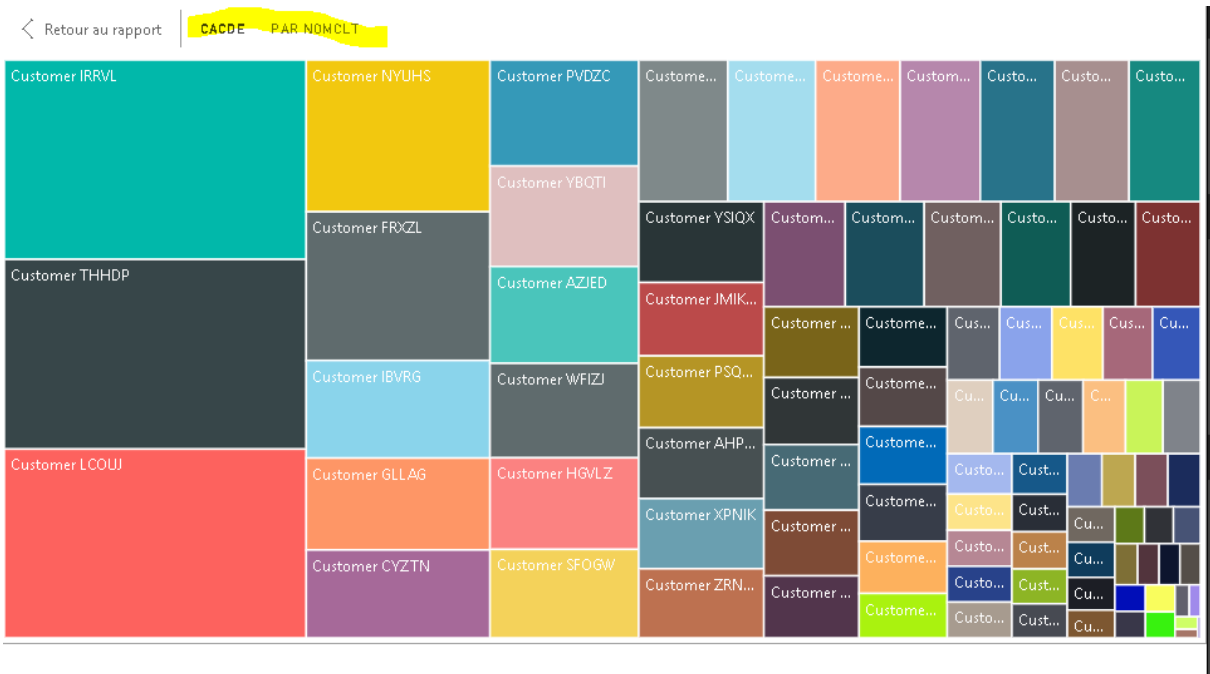
Sur Power BI on va construire le chiffre d'affaire pour chaque catégorie.



Les mêmes données représentées avec un autre graphe.



Le KPI1 (Chiffre d'affaire par client)



Le KPI 2(Chiffre d'affaire par produit)



**5) Conclusion :** Ce projet a été utile pour ma formation en tant que futur ingénieur en BI, il m'a permis d'acquérir de nouvelles compétences techniques des outils MS-SQL Server et d'assimiler de nouvelles notions dans le domaine du BI.

/\*\*\*\*\*\*Fin\*\*\*\*\*\*/

