

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE SAAD DAHLAB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE**



Mini-projet SID 2020-2021

THÈME :

**Conception, Création et
Alimentation d'un entrepôt
de donnée DW_NW pour
l'Interrogation et l'analyse
des besoins**

Réalisé par :

- IBNAISSA MOHAMED CHAKIB
- BELLIL MOUNIR

Table des matières

INTRODUCTION	4
PRESENTATION DE L'ETUDE DE CAS	5
MODELISATION DE L'ENTREPOT DE DONNEES :	6
Model en étoile :	6
Model en flocons :	8
Model en constellation :	9
IMPLIMENTATION.....	11
Création de l'entrepôt de données DW_NW :	11
ALIMENTATION DE DATA WEARHOUSE	16
Conclusion.....	22

Résumé

De nos Jours, Dans le monde de l'entreprise, l'information est une matière précieuse qu'il faut veiller à cultiver au quotidien. Dans ce sens les systèmes d'informations ont connu un véritable boom et ont toute leur place dans les différentes fonctions qui gouverne l'entreprise.

Donc la masse d'information elle a un impact sur le rendement de l'entreprise. Elle a un impact sur le rendement de l'entreprise, elle peut porter l'entreprise sur le chemin de succès comme elle peut la basculer vers le chemin de l'échec. Donc il est nécessaire de mettre en place un système pour les décideurs de l'entreprise pour bien gérer l'entreprise pour la guider vers le chemin de succès.

Pour ce faire, nous avons consacré ce travail pour créer un entrepôt de données et l'interroger en utilisant des outils OLTP et OLAP, on a utilisé la modélisation en étoile et la modélisation en flocons plus le modèle en constellation pour simplifier les besoins des décideurs, un SGBDR (fondé sur SQL) pour créer et alimenter l'entrepôt de données en plus de sa on a utilisé power bi pour interroger l'entrepôt.

Mot clés : SGBDR, SQL, OLAP, OLTP, entrepôt de données, MS POWER BI, modèle en Etoile, modelé en flocons, model en constellation

Abstract

These days, in the corporate world, information is a precious material that must be cultivated

On a daily basis. In this sense, information systems have experienced a real boom and have their place in the various functions that govern the company .

So the mass of information has an impact on the performance of the company. It has an impact on the performance of the company, it can carry the company on the path of success as it can tilt it towards the path of failure, so it is necessary to put in the place a system for business decision makers to manage the business well to guide it to the path of success.

To do this, we dedicated this work to create a data warehouse and query it using OLTP and OLAP tools, we used star modeling and snowflake modeling plus fact-flake model to simplify the needs of decision makers, SGBDR (base on SQL) to create and feed the data warehouse in addition to its power bi was used to query the warehouse.

Key words : SGBDR, SQL, OLAP, OLTP, DATA WAREHOUSE, MS POWER BI, Star model, snowflake model, fact-flake model

INTRODUCTION

Avec la diffusion de l'informatique dans tous les domaines d'activité, L'entreprise produit et traite de grandes quantités de données électroniques. Ces données sont stockées dans une base de données du système opérationnel de l'entreprise.

Il est difficile d'utiliser ces données pour l'analyse et l'aide à la décision ; elle est réalisée généralement de manière imparfaite par les décideurs grâce à des moyens traditionnelle (requêtes SQL, vues ...).

Ces moyens paraissent peu adaptés pour servir de l'analyse et l'aide à la décision. Ces bases de données opérationnelles utilisent le modèle relationnel ; celui-ci convient bien aux applications gérant l'activité quotidienne de l'entreprise, mais s'avère inadapté au décisionnel. Pour ce décalage, les entreprises ont recours à des systèmes d'aide à la décision spécifiques, basés sur l'approche des **entrepôts de données**.

“A **data warehouse** is a copy of transaction data specifically structured for query and analysis.” — Ralph Kimball

Un **entrepôt de données** est une zone de stockage où seules les informations critiques pour l'entreprise sont extraites d'une énorme base de données normalisée pour faciliter le processus de prise de décision pour la gestion.

Dans ce mini projet, nous présentons les étapes détaillées de la conception et la mise en place de notre entrepôt de données de modélisation à la création et l'alimentation qui se fait dans un environnement OLTP, et l'Interrogation de l'entrepôt avec l'analyse des données pour répondre aux besoins.

PRESENTATION DE L'ETUDE DE CAS

Le travail à réaliser est de créer un entrepôt de données pour répondre à des besoins d'analyse à partir d'une base de données opérationnel.

Pour ce travail nous utilisons la base de données **Northwind**. Qui sera notre source de données pour la création et l'alimentation de l'entrepôt de données **DW_NW**.

Pour le stockage de l'entrepôt de données on a choisi un SGBD relationnel qu'est **Microsoft SQL Server 2019**. Le même avec la base de données source **Northwind**.

- Base de données source : **Northwind**.
- SGBD relationnel : **Microsoft SQL Server 2019**.
- L'entrepôt de données : **DW_NW**.
- Un environnement OLAP : **MS Power BI**.

Notre travail est organisé selon le plan suivant :

D'abord, nous modélisons les six (06) besoins d'analyse avec des schémas en étoile et en flocons de neige. Après on a Donner le modèle en constellation avec la fusion des six (06) modèles en étoiles et en unifiant les dimensions communes.

Deuxièmement, après la modélisation nous entamons la création de l'entrepôt de données **DW_NW** et toutes les tables présentes sur le modèle en constellation avec un script SQL de création.

En finissons la création arrive la partie alimentation qui consiste extraire les données à partir de la source **Northwind** les transformer et les charger dans les tables de l'entrepôt de données **DW_NW** avec un script SQL d'alimentation.

En fin qu'on a l'entrepôt de données **DW_NW** crée et alimenté et toutes les données qu'on a besoin sont disponible on commence l'interrogation de l'entrepôt et répondre aux besoins d'analyse avec des requêtes SQL et quelques besoins avec un environnement OLAP (MS Power BI).

MODELISATION DE L'ENTREPOT DE DONNEES :

Model en étoile :

Il consiste à extraire **les tables de faits (TF)** et les **tables de dimension (TD)** d'après les besoins d'analyse.

- Analyse de la Valeur des Stocks (VAL_STK) des produits de Northwind par fournisseur (*ID, Nom, Ville, pays*) et catégorie (*ID, nom*) ; La **figure 1** représente le schéma en étoile en analyse de la Valeur des Stocks des produits par fournisseur et catégorie. Dans ce cas, au centre une table de faits appelée TF VAL_STK et autour les différentes tables de dimensions : TD Fournisseur et TD Catégorie.

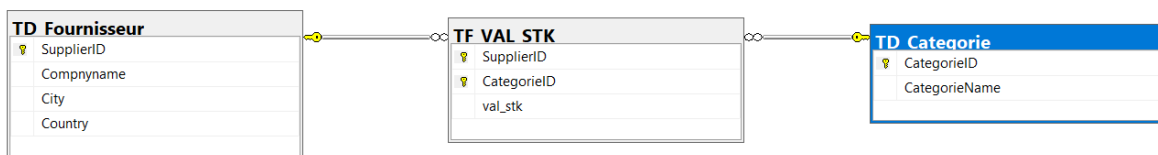


Figure 1 : Analyse de la Valeur des Stocks (VAL_STK)

- Analyse du Nombre de produits (NB_PROD) par Fournisseur (ID, Nom, Ville, Pays), Catégorie (ID, Nom) et Niveau des Ventes (**Level1** pour quantités vendues > 50, **Level2** pour 10<= quantités vendues <=50, **Level3** pour quantités vendues < 10) ; La **figure 2** représente le schéma en étoile en analyse de nombre de produit par fournisseur, catégorie et niveau des ventes. Dans ce cas, au centre une table de faits appelée TF NBR_PROD et autour les différentes tables de dimensions : TD Fournisseur, TD Catégorie et TD NIVVENTE.

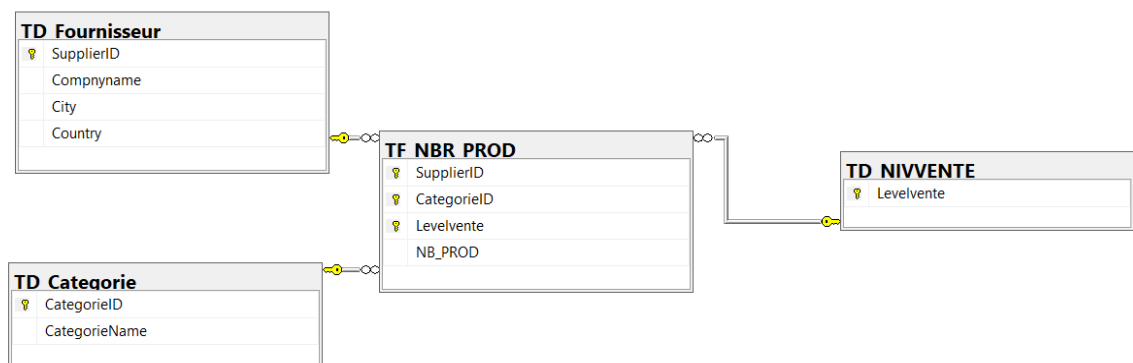


Figure 2 : Analyse du Nombre de produits (NB_PROD)

- Analyse des Chiffres d’Affaires Commandés (CA_CMD) par Année de commande, Catégorie des produits (*ID, Nom*), Client (*ID, Nom, Ville, Pays*), Employé (*ID, Nom, Prénom, Ville, Pays*) ; La **figure 3** représente le schéma en étoile en des Chiffres d’Affaires des produits Commandés par année, catégorie, client et employé. Dans ce cas, au centre une table de faits appelée TF CA_CMD et autour les différentes tables de dimensions : TD AnnéeCommande, TD Catégorie, TD Client et TD Employee.

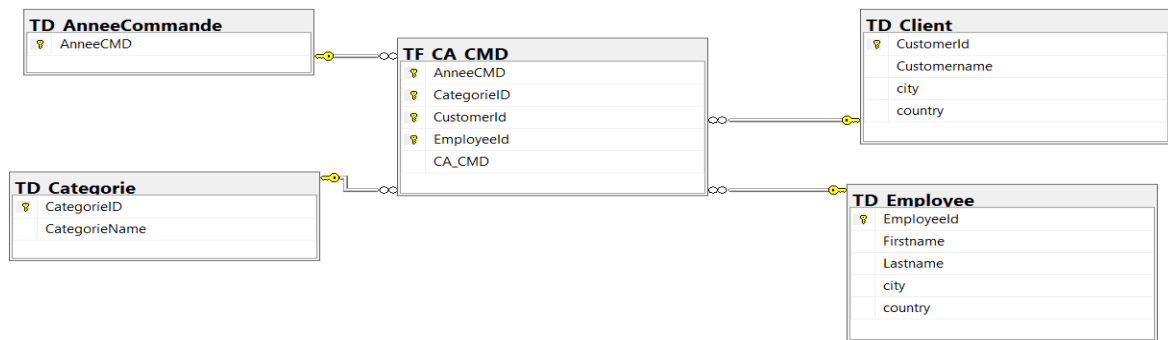


Figure 3 : Analyse des Chiffres d’Affaires Commandés (CA_CMD)

- Analyse des Chiffres d’Affaires Livrés (CA_LIV) par Année de livraison, Catégorie des produits (*ID, Nom*), Client (*ID, Nom, Ville, Pays*), Transporteur (*ID, Nom*) ; La **figure 4** représente le schéma en étoile en analyse des Chiffres d’Affaires des produits livrés par année, catégorie, client et transporteur. Dans ce cas, au centre une table de faits appelée TF CA_LIV et autour les différentes tables de dimensions : TD AnnéeLIV, TD Catégorie, TD Client et TD Transporteur.

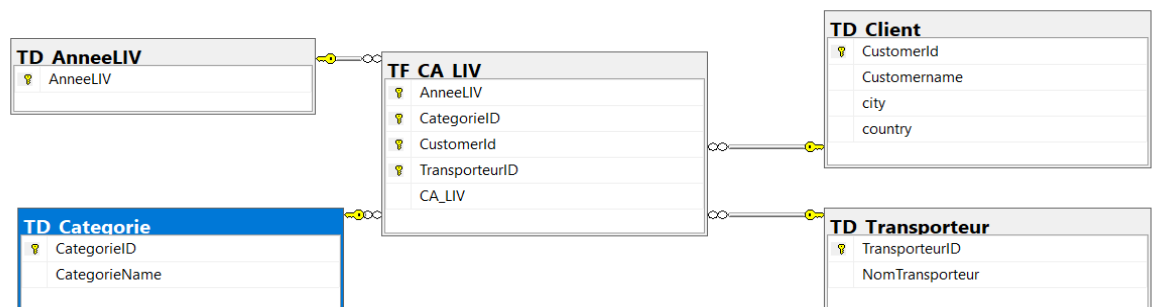


Figure 4 : Analyse des Chiffres d’Affaires Livrés (CA_LIV)

- Analyse du Nombre d'Employés (NB_EMP) par Age, Ville et Pays ; La **figure 5** représente le schéma en étoile en analyse du nombre d'employés par âge, ville et pays. Dans ce cas, au centre une table de faits appelée TF NB_EMP et autour les différentes tables de dimensions : TD Age et TD Ville.



Figure 5 : Analyse du Nombre d'Employés (NB_EMP)

- Analyse du Nombre de Clients par Ville, Pays et Niveau des Commandes (**Level1** pour nb commandes > 50, **Level2** pour 15<= nb commandes <=50, **Level3** pour nb commandes < 15) ; La **figure 6** représente le schéma en étoile en analyse du nombre de client par ville, pays et niveau des commandes. Dans ce cas, au centre une table de faits appelée TF NB_EMP et autour les différentes tables de dimensions : TD Ville et TD NIVCMD.

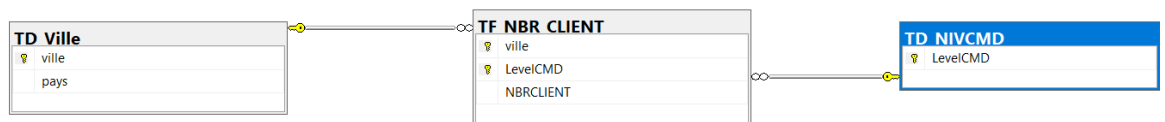


Figure 6 : Analyse du Nombre de Clients (NBR_CLIENT)

Model en flocons :

Il s'agit de model en étoile plus une normalisation dans notre cas on a que deux changement :

- Analyse du Nombre d'Employés (NB_EMP) par Age, Ville et Pays ; La figure 7 montre le schéma en flocon de neige avec les dimensions Age et Ville éclatées en sous hiérarchies (hiérarchie alternative), la dimension Ville a été éclatée en deux hiérarchies alternatives, TD Ville et TD Pays.

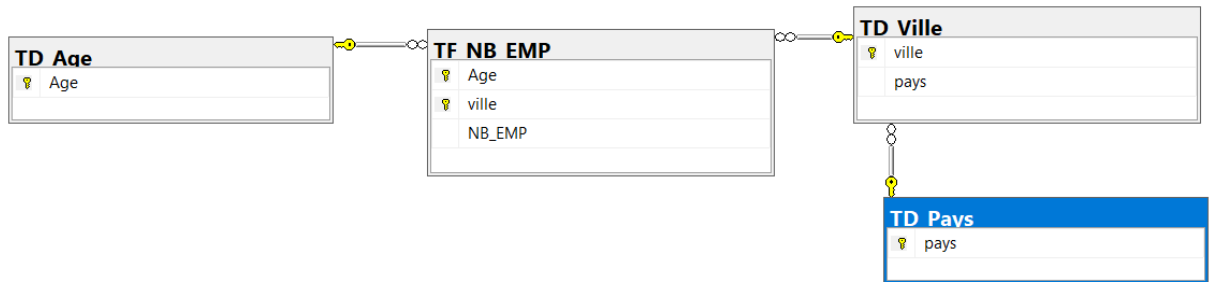


Figure 7 : Analyse du Nombre d'Employés (NB_EMP) en FLOCONS

- Analyse du Nombre de Clients par Ville, Pays et Niveau des Commandes (**Level1** pour nb commandes > 50, **Level2** pour $15 \leq \text{nb commandes} \leq 50$, **Level3** pour nb commandes < 15) ; La **figure 8** montre le schéma en flocon de neige avec les dimensions Niveau des Commandes et Ville éclatées en sous hiérarchies (hiérarchie alternative), la dimension Ville a été éclatée en deux hiérarchies alternatives, TD Ville et TD Pays.

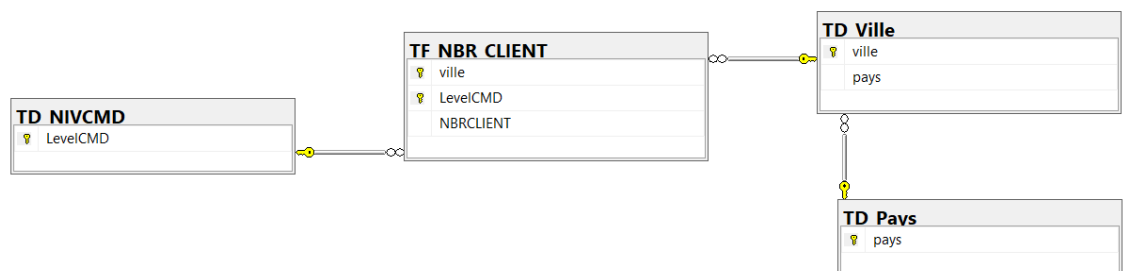


Figure 8 : Analyse du Nombre de Clients par Ville(NBR_CLIENT) en FLOCONS

Model en constellation :

La modélisation en constellation consiste à fusionner plusieurs modèles en étoile qui peuvent utiliser des dimensions communes.

La **figure 9** montre le schéma en constellation qui est composé de 6 relations de faits. Qui sont la Valeur des Stocks (VAL_STK), Nombre de produits (NBR_PROD), d'Affaires Commandés (CA_CMD), Chiffres d'Affaires Livrés (CA_LIV), Nombre d'Employés (NB_EMP), Nombre de Clients (NBR_CLIENT). Avec les différentes dimensions Fournisseur, Catégorie, Niveau des Ventes, Année de commande, Client, Employé, Année de livraison, Transporteur, Age, Ville et Niveau des Commandes.

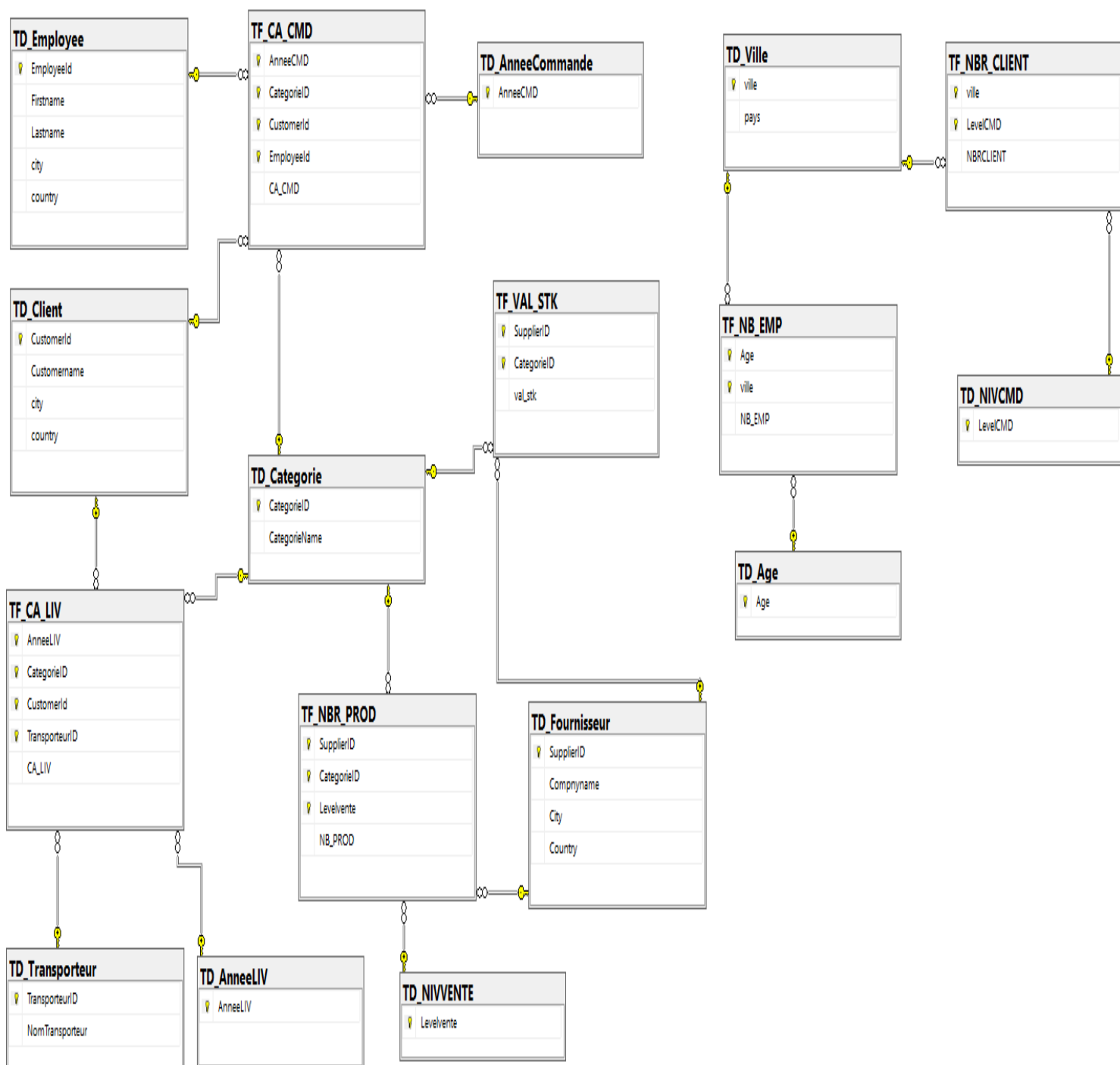


Figure 9 : model en constellation

IMPLIMENTATION

Création de l'entrepôt de données DW_NW :

Pour la création de l'entrepôt de données DW_NW on utilise la clause CREATE DATABASE et USE pour accéder à la base. Comme suit :

```
--1--  
CREATE DATABASE DW_NW  
--2--  
USE DW_NW
```

Création les tables de dimension et les tables de fait

En utilisant la clause **CREATE TABLE** pour créer toutes les tables de dimension en premier et les tables de fait après. Ajouter des contraintes avec **ALTER TABLE constraint** pour les **primary key** et les **foreign key**. Dans le script sql suivant :

Script 1 :

--Création tables de dimension --

--TD_Fournisseur--

```
CREATE TABLE TD_Fournisseur(  
    SupplierID int not null,  
    Compnyname nvarchar(40) not null,  
    city nvarchar(15) not null,  
);
```

--TD_Categorie--

```
CREATE TABLE TD_Categorie(  
    CategorieID int not null,  
    CategorieName nvarchar(15) not null  
);
```

--TD_NIVVENTE--

```
CREATE TABLE TD_NIVVENTE(  
    Levelvente nvarchar(20) not null  
);
```

--TD_AnneeCommande--

```
CREATE TABLE TD_AnneeCommande(  
    AnneeCMD int not null  
);
```

--TD_Client--

```
CREATE TABLE TD_Client(  
    CustomerId nchar(5) not null ,  
    Customername nvarchar(40) ,  
    city nvarchar(15) not null  
);
```

--TD_Employee--

```
CREATE TABLE TD_Employee (  
    EmployeeId int not null ,  
    Firstname nvarchar(10) not null,  
    Lastname nvarchar(20) not null,  
    city nvarchar(15) not null  
);
```

--TD_AnneeLIV--

```
CREATE TABLE TD_AnneeLIV(  
    AnneeLIV int not null  
);
```

--TD_Transporteur--

```
CREATE TABLE TD_Transporteur (  
    TransporteurID int not null ,  
    NomTransporteur nvarchar(40)  
);
```

--TD_Age--

```
CREATE TABLE TD_Age(  
    Age int not null  
);
```

--TD_Ville--

```
CREATE TABLE TD_Ville(  
    ville nvarchar(15) not null ,  
    pays nvarchar(15)  
);
```

--TD_NIVCMD--

```
CREATE TABLE TD_NIVCMD(
    LevelCMD nvarchar(20) not null
);
```

--Ajouter des contraintes pour les tables de dimensions --

```
ALTER TABLE TD_NIVCMD ADD constraint LevelCMD primary key (LevelCMD)
ALTER TABLE TD_Ville ADD constraint ville primary key (ville)
ALTER TABLE TD_Age ADD constraint Age primary key (Age)
ALTER TABLE TD_AnneeLIV ADD constraint AnneeLIV primary key (AnneeLIV)
ALTER TABLE TD_Transporteur ADD constraint TransporteurID primary key (TransporteurID)
ALTER TABLE TD_Categorie ADD constraint CategorieID primary key (CategorieID)
ALTER TABLE TD_NIVVENTE ADD constraint Levelvente primary key (Levelvente)
ALTER TABLE TD_AnneeCommande ADD constraint AnneeCMD primary key (AnneeCMD)
ALTER TABLE TD_Fournisseur ADD
constraint SupplierID primary key (SupplierID),
constraint cityFournisseur foreign key (city) references TD_Ville (ville)
ALTER TABLE TD_Client ADD
constraint CustomerId primary key (CustomerId),
constraint cityClient foreign key (city) references TD_Ville (ville)
ALTER TABLE TD_Employee ADD
constraint EmployeeId primary key (EmployeeId),
constraint cityEmployee foreign key (city) references TD_Ville (ville)
```

--Création tables de Faits--

--TF_VAL_STK--

```
CREATE TABLE TF_VAL_STK(
    SupplierID int not null,
    CategorieID int not null,
    val_stk float
);
```

--TF_NBR_PROD--

```
CREATE TABLE TF_NBR_PROD(
    SupplierID int not null,
```

```

        CategoriID int not null,

        Levelvente nvarchar(20) not null,

        NB_PROD int

    );

--TF_CA_CMD--

    CREATE TABLE TF_CA_CMD(

        AnneeCMD int not null,

        CategoriID int not null,

        CustomerId nchar(5) not null,

        EmployeeID int not null,

        CA_CMD float);

--TF_CA_LIV--

    CREATE TABLE TF_CA_LIV(

        AnneeLIV int not null,

        CategoriID int not null,

        CustomerId nchar(5) not null,

        TransporteurID int not null,

        CA_LIV float

    );

--TF_NB_EMP--

    CREATE TABLE TF_NB_EMP(

        Age int not null,

        ville nvarchar(15) not null,

        NB_EMP int

    );

--TF_NBR_CLIENT--

    CREATE TABLE TF_NBR_CLIENT(

        ville nvarchar(15) not null,

        LevelCMD nvarchar(20) not null,

        NBRCLIENT int

    );

```

--Ajouter des contraintes pour les tables de dimensions--

ALTER TABLE TF_VAL_STK ADD

constraint PK_VAL_STK primary key (SupplierID, CategorieID),

constraint FK_VAL_STK_SupplierID foreign key (SupplierID) references TD_Fournisseur (SupplierID),

constraint FK_VAL_STK_CategorieID foreign key (CategorieID) references TD_Categorie (CategorieID)

ALTER TABLE TF_NBR_PROD ADD

constraint PK_NBR_PROD primary key (SupplierID, CategorieID,Levelvente),

constraint FK_NBR_PROD_SupplierID foreign key (SupplierID) references TD_Fournisseur (SupplierID),

constraint FK_NBR_PROD_CategorieID foreign key (CategorieID) references TD_Categorie (CategorieID),

constraint FK_NBR_PROD_Levelvente foreign key (Levelvente) references TD_NIVVENTE (Levelvente)

ALTER TABLE TF_CA_CMD ADD

constraint PK_CA_CMD primary key (AnneeCMD, CategorieID,CustomerId,EmployeeId),

constraint FK_CA_CMD_AnneeCMD foreign key (AnneeCMD) references TD_AnneeCommande (AnneeCMD),

constraint FK_CA_CMD_CategorieID foreign key (CategorieID) references TD_Categorie (CategorieID),

constraint FK_CA_CMD_CustomerId foreign key (CustomerId) references TD_Client (CustomerId),

constraint FK_CA_CMD_EmployeeId foreign key (EmployeeId) references TD_Employee (EmployeeId)

ALTER TABLE TF_CA_LIV ADD

constraint PK_CA_LIV primary key (AnneeLIV, CategorieID,CustomerId,TransporteurID),

constraint FK_CA_LIV_AnneeLIV foreign key (AnneeLIV) references TD_AnneeLIV (AnneeLIV),

constraint FK_CA_LIV_CategorieID foreign key (CategorieID) references TD_Categorie (CategorieID),

constraint FK_CA_LIV_CustomerId foreign key (CustomerId) references TD_Client (CustomerId),

constraint FK_CA_LIV_TransporteurID foreign key (TransporteurID) references TD_Transporteur (TransporteurID)

ALTER TABLE TF_NB_EMP ADD

constraint PK_NB_EMP primary key (Age, ville),

constraint FK_NB_EMP_Age foreign key (Age) references TD_Age (Age),

constraint FK_NB_EMP_ville foreign key (ville) references TD_Ville (ville)

ALTER TABLE TF_NBR_CLIENT ADD

constraint PK_NBR_CLIENT primary key (ville,LevelCMD),

constraint FK_NBR_CLIENT_ville foreign key (ville) references TD_Ville (ville),

constraint FK_NBR_CLIENT_LevelCMD foreign key (LevelCMD) references TD_NIVCMD (LevelCMD)

ALIMENTATION DE DATA WEARHOUSE

--Alimentation tables de dimensions--

use northwind

--TD_NIVCMD--

```
go
INSERT INTO DW_NW.dbo.TD_NIVCMD(LevelCMD)
SELECT DISTINCT
    CASE
        WHEN Count(OrderID) <15 THEN 'LEVEL3'
        WHEN Count(OrderID) >=15 and Count(OrderID) <=50 THEN 'LEVEL2'
        ELSE 'LEVEL1 '
    END
from Orders
GROUP BY CustomerID
```

--TD_Age--

```
go
INSERT INTO DW_NW.dbo.TD_Age (DW_NW.dbo.TD_Age.Age)
SELECT DISTINCT DATEDIFF(year,BirthDate,GETDATE()) from Employees
WHERE cast(DATEDIFF(year,BirthDate,GETDATE()) as varchar(4)) COLLATE FRENCH_CI_AS not in( select
cast(DW_NW.dbo.TD_Age.Age as varchar(4)) from DW_NW.dbo.TD_Age)
```

--TD_Transporteur--

```
go
INSERT INTO
DW_NW.dbo.TD_Transporteur(DW_NW.dbo.TD_Transporteur.TransporteurID,DW_NW.dbo.TD_Transporteur.
NomTransporteur)
SELECT ShipperID,CompanyName from Shippers
WHERE cast(ShipperID as varchar(4))+cast(CompanyName as varchar(4)) COLLATE FRENCH_CI_AS not in ( select
cast ( DW_NW.dbo.TD_Transporteur.TransporteurID as
varchar(4))+cast(DW_NW.dbo.TD_Transporteur.NomTransporteur as varchar(4)) from
DW_NW.dbo.TD_Transporteur)
```

--TD_AnnéeLIV--

```
go
INSERT INTO DW_NW.dbo.TD_AnneeLIV( DW_NW.dbo.TD_AnneeLIV.AnneeLIV)
SELECT DISTINCT year(ShippedDate)from Orders where year(ShippedDate) IS NOT NULL
and cast(year(ShippedDate) as varchar(4)) COLLATE FRENCH_CI_AS not in (select
cast(DW_NW.dbo.TD_AnneeLIV.AnneeLIV as varchar(4)) from DW_NW.dbo.TD_AnneeLIV)
```

--TD_AnneeCommande--

```
go
INSERT INTO DW_NW.dbo.TD_AnneeCommande ( DW_NW.dbo.TD_AnneeCommande.AnneeCMD)
SELECT DISTINCT year(OrderDate) from Orders
where cast(year(OrderDate) as varchar(4)) COLLATE FRENCH_CI_AS not in (select
cast(DW_NW.dbo.TD_AnneeCommande.AnneeCMD as varchar(4)) from DW_NW.dbo.TD_AnneeCommande)
```

--TD_NIVVENT--

```
go
INSERT INTO DW_NW.dbo.TD_NIVVENTE(DW_NW.dbo.TD_NIVVENTE.Levelvente)
SELECT DISTINCT CASE
    WHEN Quantity < 10 THEN 'LEVEL3'
    WHEN Quantity >= 10 and Quantity <=50 THEN 'LEVEL2'
    ELSE 'LEVEL1'
```

```

END
from [Order Details]
where CASE
    WHEN Quantity < 10 THEN 'LEVEL3'
    WHEN Quantity >= 10 and Quantity <=50 THEN 'LEVEL2'
    ELSE 'LEVEL1'
END COLLATE FRENCH_CI_AS not in (select DW_NW.dbo.TD_NIVVENTE.Levelvente from
DW_NW.dbo.TD_NIVVENTE)

--TD_categorie--
go
INSERT INTO DW_NW.dbo.TD_Categorie
(DW_NW.dbo.TD_Categorie.CategorieID,DW_NW.dbo.TD_Categorie.CategorieName)
SELECT CategoryID,CategoryName from Categories
where cast(CategoryID as varchar(4))+CategoryName COLLATE FRENCH_CI_AS not in (select cast(
DW_NW.dbo.TD_Categorie.CategorieID as varchar(4))+DW_NW.dbo.TD_Categorie.CategorieName from
DW_NW.dbo.TD_Categorie)

--TD_Ville--
go
INSERT INTO DW_NW.dbo.TD_Ville (DW_NW.dbo.TD_Ville.ville,DW_NW.dbo.TD_Ville.pays)
SELECT  City, Country from Employees
WHERE City IS NOT NULL and City+ Country COLLATE FRENCH_CI_AS not in (select
DW_NW.dbo.TD_Ville.ville+DW_NW.dbo.TD_Ville.pays from DW_NW.dbo.TD_Ville)
UNION
SELECT  City, Country from Customers
WHERE City IS NOT NULL and City+ Country COLLATE FRENCH_CI_AS not in (select
DW_NW.dbo.TD_Ville.ville+DW_NW.dbo.TD_Ville.pays from DW_NW.dbo.TD_Ville)
UNION
SELECT  City, Country from Suppliers
WHERE City IS NOT NULL and City+ Country COLLATE FRENCH_CI_AS not in (select
DW_NW.dbo.TD_Ville.ville+DW_NW.dbo.TD_Ville.pays from DW_NW.dbo.TD_Ville)

--TD_Client--
go
INSERT INTO
DW_NW.dbo.TD_Client(DW_NW.dbo.TD_Client.CustomerId,DW_NW.dbo.TD_Client.Customername,DW_NW.d
bo.TD_Client.city)
SELECT CustomerID,CompanyName,City from Customers
where CustomerID+CompanyName+City COLLATE FRENCH_CI_AS not in (select
DW_NW.dbo.TD_Client.CustomerId+DW_NW.dbo.TD_Client.Customername+DW_NW.dbo.TD_Client.city from
DW_NW.dbo.TD_Client)

--TD_Fournisseur--
go
INSERT INTO DW_NW.dbo.TD_Fournisseur
(DW_NW.dbo.TD_Fournisseur.SupplierID,DW_NW.dbo.TD_Fournisseur.Compnyname,DW_NW.dbo.TD_Fourni
sreur.City)
SELECT SupplierID, CompanyName, City from Suppliers
where SupplierID+CompanyName+City COLLATE FRENCH_CI_AS not in (select
DW_NW.dbo.TD_Fournisseur.SupplierID+DW_NW.dbo.TD_Fournisseur.Compnyname+DW_NW.dbo.TD_Fourni
sreur.City from DW_NW.dbo.TD_Fournisseur)

```

```
--TD_Employee--
go
INSERT INTO
DW_NW.dbo.TD_Employee(DW_NW.dbo.TD_Employee.EmployeeId,DW_NW.dbo.TD_Employee.Firstname,D
W_NW.dbo.TD_Employee.Lastname,DW_NW.dbo.TD_Employee.city)
SELECT EmployeeId,FirstName,LastName,City
from Employees
where EmployeeId+FirstName+LastName+City COLLATE FRENCH_CI_AS not in (select
DW_NW.dbo.TD_Employee.EmployeeId+DW_NW.dbo.TD_Employee.Firstname+DW_NW.dbo.TD_Employee.La
stname+DW_NW.dbo.TD_Employee.city from DW_NW.dbo.TD_Employee)
```

--Alimentation tables de faits--

```
use northwind
```

```
--1/TF_VAL_STK--
```

```
INSERT INTO
DW_NW.dbo.TF_VAL_STK(DW_NW.dbo.TF_VAL_STK.SupplierID,DW_NW.dbo.TF_VAL_STK.CategorieID,DW_N
W.dbo.TF_VAL_STK.val_stk)
```

```
select SupplierID,CategoryID, SUM(UnitPrice * UnitsInStock) as VAL_STK
```

```
from Products
```

```
WHERE SupplierID+CategoryID not in (select
DW_NW.dbo.TF_VAL_STK.SupplierID+DW_NW.dbo.TF_VAL_STK.CategorieID from DW_NW.dbo.TF_VAL_STK)
```

```
GROUP BY SupplierID,CategoryID
```

```
--2/TF_NB_PROD--
```

```
INSERT INTO
DW_NW.dbo.TF_NBR_PROD(DW_NW.dbo.TF_NBR_PROD.SupplierID,DW_NW.dbo.TF_NBR_PROD.CategorieID,
DW_NW.dbo.TF_NBR_PROD.Levelvente,DW_NW.dbo.TF_NBR_PROD.NB_PROD)
```

```
select DISTINCT P.SupplierID,P.CategoryID, CASE
```

```
    WHEN Quantity < 10 THEN 'LEVEL3'
```

```
    WHEN Quantity >= 10 and Quantity <=50 THEN 'LEVEL2'
```

```
    ELSE 'LEVEL1'
```

```
END AS Levelvente,
```

```
COUNT(OD.ProductID) AS NB_PROD FROM
```

```
Products P JOIN [Order Details] OD ON P.ProductID = OD.ProductID
```

```
WHERE P.SupplierID+P.CategoryID not in (select
DW_NW.dbo.TF_NBR_PROD.SupplierID+DW_NW.dbo.TF_NBR_PROD.CategorieID from
DW_NW.dbo.TF_NBR_PROD)
```

```
GROUP BY SupplierID,CategoryID,CASE
```

```
    WHEN Quantity < 10 THEN 'LEVEL3'
```

```
    WHEN Quantity >= 10 and Quantity <=50 THEN 'LEVEL2'
```

```
    ELSE 'LEVEL1'
```

END

--3/TF_CA_CMD--

INSERT INTO

DW_NW.dbo.TF_CA_CMD(DW_NW.dbo.TF_CA_CMD.AnneeCMD,DW_NW.dbo.TF_CA_CMD.CategorieID,DW_NW.dbo.TF_CA_CMD.CustomerId,DW_NW.dbo.TF_CA_CMD.EmployeeId,DW_NW.dbo.TF_CA_CMD.CA_CMD)

select year(OrderDate) as AnneeCMD,CategoryID, CustomerID, EmployeeID,SUM((OD.UnitPrice*Quantity)-(Discount*100*OD.UnitPrice)/100) as CA_CMD

FROM Orders O JOIN [Order Details] OD ON O.OrderID = OD.OrderID

JOIN Products P ON OD.ProductID= P.ProductID

WHERE ShippedDate IS NULL and cast(year(OrderDate) as char(4)) + cast(CategoryID as char(4)) + cast(CustomerID as char(4))+ cast(EmployeeID as char(4)) COLLATE FRENCH_CI_AS not in (select cast (DW_NW.dbo.TF_CA_CMD.AnneeCMD as char(4))+ cast(DW_NW.dbo.TF_CA_CMD.CategorieID as char(4))+ cast(DW_NW.dbo.TF_CA_CMD.CustomerId as char(4))+ cast(DW_NW.dbo.TF_CA_CMD.EmployeeId as char(4)) from DW_NW.dbo.TF_CA_CMD)

GROUP BY year(OrderDate),CategoryID, CustomerID, EmployeeID

--4/TF_CA_LIV--

INSERT INTO

DW_NW.dbo.TF_CA_LIV(DW_NW.dbo.TF_CA_LIV.AnneeLIV,DW_NW.dbo.TF_CA_LIV.CategorieID,DW_NW.dbo.TF_CA_LIV.CustomerId,DW_NW.dbo.TF_CA_LIV.TransporteurID,DW_NW.dbo.TF_CA_LIV.CA_LIV)

SELECT year(ShippedDate) as AnneeLIV,CategoryID ,CustomerID,ShipVia,SUM((OD.UnitPrice*Quantity)-(Discount*100*OD.UnitPrice)/100) as CA_LIV

FROM Orders O JOIN [Order Details] OD ON O.OrderID = OD.OrderID

JOIN Products P ON OD.ProductID = P.ProductID

WHERE year(ShippedDate) IS NOT NULL and cast(year(ShippedDate) as char(4)) + cast(CategoryID as char(4))+ cast(CustomerID as char(4)) + cast(ShipVia as char(4)) COLLATE FRENCH_CI_AS not in (select cast(DW_NW.dbo.TF_CA_LIV.AnneeLIV as char(4))+cast(DW_NW.dbo.TF_CA_LIV.CategorieID as char(4))+cast(DW_NW.dbo.TF_CA_LIV.CustomerId as char(4))+cast(DW_NW.dbo.TF_CA_LIV.TransporteurID as char(4)) from DW_NW.dbo.TF_CA_LIV)

GROUP BY year(ShippedDate),CategoryID ,CustomerID,ShipVia

--5/TF_NB_EMP--

INSERT INTO

DW_NW.dbo.TF_NB_EMP(DW_NW.dbo.TF_NB_EMP.Age,DW_NW.dbo.TF_NB_EMP.ville,DW_NW.dbo.TF_NB_EMP.NB_EMP)

SELECT DATEDIFF(year,BirthDate,GETDATE()) as Age,City as ville,COUNT(EmployeeID)

FROM Employees

WHERE cast (DATEDIFF(year,BirthDate,GETDATE()) as char(2)) + City COLLATE FRENCH_CI_AS not in (select cast (DW_NW.dbo.TF_NB_EMP.Age as char(2)) + DW_NW.dbo.TF_NB_EMP.ville from DW_NW.dbo.TF_NB_EMP)

GROUP BY DATEDIFF(year,BirthDate,GETDATE()),City

--6/TF_NBR_CLIENT--

INSERT INTO

DW_NW.dbo.TF_NBR_CLIENT(DW_NW.dbo.TF_NBR_CLIENT.ville,DW_NW.dbo.TF_NBR_CLIENT.LevelCMD,DW_NW.dbo.TF_NBR_CLIENT.NBRCLIENT)

SELECT City,CASE

WHEN Count(OrderID) <15 THEN 'LEVEL3'

WHEN Count(OrderID) >=15 and Count(OrderID) <=50 THEN 'LEVEL2'

EISE 'LEVEL1'

END as LevelCMD,COUNT(C.CustomerID) as NBRCLIENT FROM Customers C JOIN Orders O

ON C.CustomerID = O.CustomerID

WHERE City COLLATE FRENCH_CI_AS not in (select DW_NW.dbo.TF_NBR_CLIENT.ville from DW_NW.dbo.TF_NBR_CLIENT)

GROUP BY City

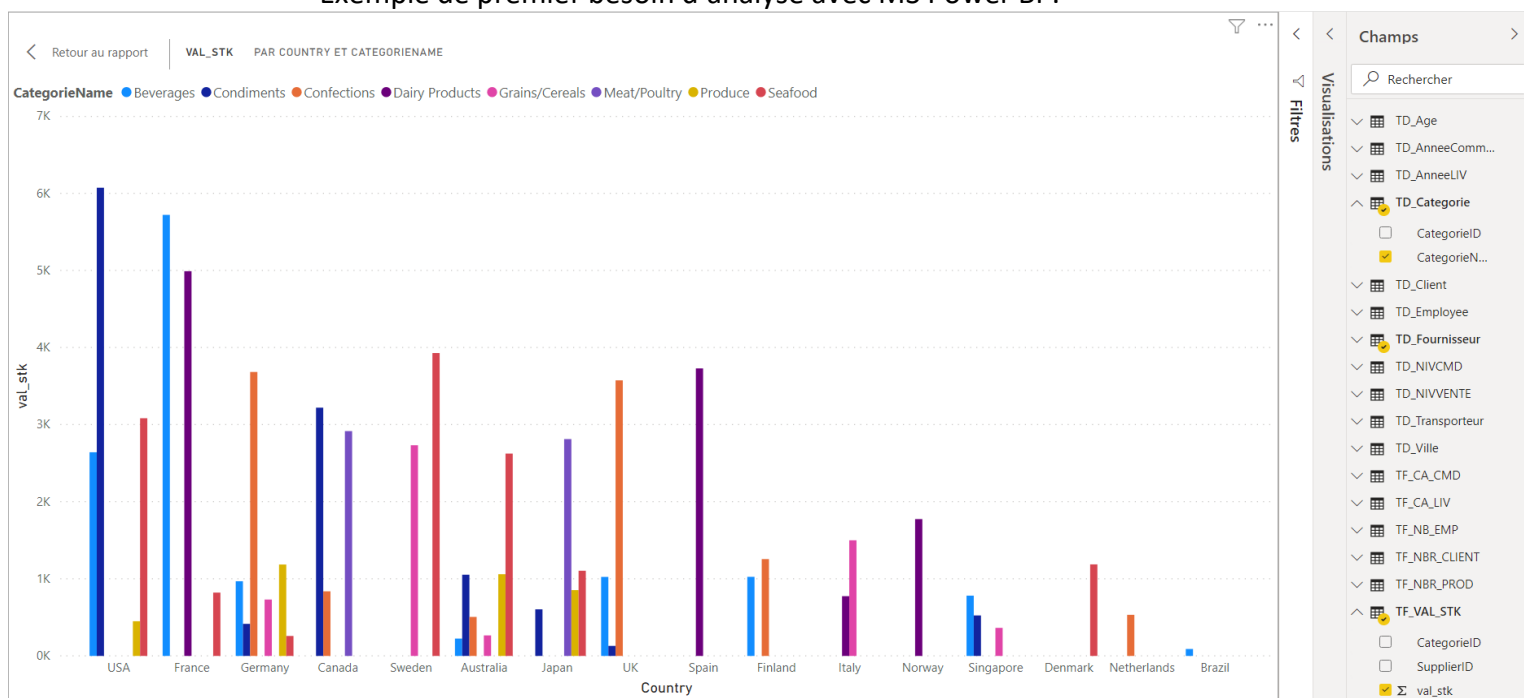
ANALYSE DES BESOINS

- Analyse de la Valeur des Stocks (VAL_STK) des produits de Northwind par origine (*pays du fournisseur*) et catégorie (*Nom*)

```
select V.pays,C.CategorieName, SUM(T.val_stk) as VAL_STK FROM TF_VAL_STK T
JOIN TD_Fournisseur F ON T.SupplierID = F.SupplierID
JOIN TD_Ville V ON V.ville = F.city
JOIN TD_Categorie C ON T.CategorieID= C.CategorieID
GROUP BY V.pays,C.CategorieName
```

MS Power BI :

Exemple de premier besoin d'analyse avec MS Power BI :



- Analyse du Nombre de produits (NB_PROD) par origine (Pays du fournisseur), Nom de la catégorie et Niveau des Ventes (**Level1** pour quantités vendues > 50, **Level2** pour 10<= quantités vendues <=50, **Level3** pour quantités vendues < 10)

```
select V.pays,C.CategorieName,N.Levelvente ,SUM(N.NB_PROD) as NB_PROD FROM
TF_NBR_PROD N JOIN TD_Fournisseur F ON N.SupplierID = F.SupplierID
JOIN TD_Categorie C ON N.CategorieID= C.CategorieID
JOIN TD_Ville V ON V.ville = F.city
GROUP BY V.pays,C.CategorieName,N.Levelvente
```

- Analyse des Chiffres d’Affaires Commandés (CA_CMD) par Année de commande, Nom du Client et nom et prénom de l’Employé

```
SELECT AnneeCMD,C.Customername,E.Firstname,E.Lastname,SUM(CA_CMD) as
CA_CMD FROM TF_CA_CMD T JOIN TD_Client C ON
T.CustomerId = C.CustomerId JOIN TD_Employee E ON T.EmployeeId = E.EmployeeId
GROUP BY AnneeCMD,C.Customername,E.Firstname,E.Lastname
```

- Analyse des Chiffres d’Affaires Livrés (CA_LIV) par Année de livraison, Nom de la Catégorie, Pays de destination (Pays du Client)

```
SELECT AnneeLIV,C.CategorieName,V.pays,SUM(CA_LIV) as CA_LIV FROM TF_CA_LIV
T JOIN TD_Categorie C
ON T.CategorieID = C.CategorieID
JOIN TD_Client Cl ON T.CustomerId = Cl.CustomerId
JOIN TD_Ville V ON V.ville = cl.city
GROUP BY AnneeLIV,C.CategorieName,V.pays
```

- Analyse du Nombre d’Employés (NB_EMP) par Age et Pays

```
SELECT Age,V.pays,SUM(NB_EMP) as NB_EMP FROM TF_NB_EMP T JOIN TD_Ville V
ON T.ville = V.ville
GROUP BY Age,V.pays
```

- Analyse du Nombre de Clients par Pays et Niveau des Commandes (**Level1** pour nb commandes > 50, **Level2** pour 15<= nb commandes <=50, **Level3** pour quantités vendues < 15)

```
SELECT V.pays,LevelCMD,SUM(NBRCLIENT) as NBR_CLIENT FROM TF_NBR_CLIENT T
JOIN TD_Ville V ON T.ville=V.ville
GROUP BY V.pays,LevelCMD
```

Conclusion

Ce mini projet étale essentiellement les étapes de modélisation de l'entrepôt qui se fait dans tous les cas grâce à la modélisation dimensionnelle et l'alimentation des données.

Une fois l'alimentation terminée, l'exploitation des données peut alors se faire par différentes méthodes. L'utilisation d'outil OLAP reste, cependant, l'aspect le plus intéressant dans cette exploitation permettant la navigation dans les données de l'entrepôt à la demande.