



# Rapport de projet Northwind

**Master spécialisé en science des données**

**Module : Business intelligence**

**Sujet :**

Projet: Conception et implémentation d'un datawarehouse  
sur la base de données "Northwind"

*(PENTAHO SUITE)*

**Réalisé par :**

**JAQUIRI Younes**

**BASTA Mohammed**

**Encadré par :**

**Mdm JIHAD Zaher**

Chapitre 1 : Présentation du projet .....	6
Chapitre 2 : Design du schéma logique et dimensionnel.....	10
❖ 2.1 les business drivers et les objectifs métiers .....	11
❖ 2.2 les processus métiers clés, KPI et éventuelles requêtes décisionnelles.....	12
❖ 2.3 schéma dimensionnel .....	14
❖ 2.4 Estimation de la taille du Datamart.....	16
Chapitre 3 : Intégration des données.....	22
Introduction	
❖ 3.1 Extraction des données.....	23
❖ 3.2 Transformation sur les données.....	23
❖ 3.3 Chargement des données.....	24
❖ 3.2 Démonstration .....	24
Chapitre 4 : Mondrian Schéma Design.....	42
Introduction	
❖ 4.1 Les dimensions .....	44
❖ 4.2 Les mesures.....	46
Chapitre 5 : MDX Querying.....	48

## TABLE DES MATIÈRES

---

Chapitre 6 : Datamining.....	53
Chapitre 7 : Reporting.....	59
Références .....	63

## LISTE DES FIGURES

---

Figure 1 la structure de la table de base de données Northwind	8
Figure 2 diagramme de priorisation des processus	13
Figure 3 matrice en bus des données des dimensions	14
Figure 4 schéma en étoile pour le Datamart	15
Figure 5 documentation sur la taille des variables sous mysql	17
Figure 6 connexion à la base de donnée datawarehouse	24
Figure 7 connexion à la base des données source	25
Figure 8 ETL création des dimensions	25
Figure 9 ETL création de la dimension Produit	26
Figure 10 ETL création de la dimension Produit en détails	26
Figure 11 ETL création de la dimension Employe	26
Figure 12 ETL création de la dimension Employé en détails	27
Figure 13 ETL création de la dimension Client	27
Figure 14 ETL création de la dimension Client en détails	28
Figure 15 ETL création de la dimension Fournisseur	28
Figure 16 ETL création la dimension Fournisseur en détails	28
Figure 17 ETL création de la dimension Expéditeur	29
Figure 18 ETL création de la dimension Expéditeur en détails	29
Figure 19 ETL création de la dimension Région	30
Figure 20 ETL création de la dimension région en détails	30
Figure 21 ETL création de la dimension Temps	31
Figure 22 ETL création des paramètre dans la dimension temps	32
Figure 23 ETL création des paramètres dans la dimension temps	32
Figure 24 ETL calcul de la différence entre la date de début et de fin	33
Figure 25 ETL duplication de la valeur calculée de la dimension temps	34
Figure 26 ETL la dimension temps avant le mapping	34
Figure 27 ETL sélection des colonnes de la dimension temps	35
Figure 28 ETL chargement des données dans la dimension temps	36
Figure 29 ETL création de la table de faits commande	37
Figure 30 ETL extraction des données depuis la base source	37
Figure 31 ETL création des variables de la table de faits	38
Figure 32 ETL récupération de la clé de dim_time	39
Figure 33 ETL récupération de la clé de dim_region	40
Figure 34 ETL chargement des données dans la table de faits	41
Figure 35 schéma Mondrian création de la table de faits	43
Figure 37 création de la table time en XML	44
Figure 38 les niveaux de la table Temps en XML	44
Figure 39 code XML de la création de la dimension Région	44
Figure 36 schéma Mondrian du data warehouse	44
Figure 40 code XML de la création de la dimension Employé	45
Figure 41 code XML de la création de la dimension Client	45
Figure 42 code XML de la création de la dimension Expéditeur	45
Figure 43 code XML de la création de la dimension Fournisseur	46

## LISTE DES FIGURES

---

Figure 44 la création de la dimension Produit dans le schéma Mondrian	46
Figure 45 la mesure Prix en XML	46
Figure 46 la mesure Quantité en XML	46
Figure 47 la mesure total en XML	47
Figure 48 la mesure nombre de jr de livraison en XML	47
Figure 49 la requête décisionnelle 1	49
Figure 50 la requête décisionnelle 2	49
Figure 51 la requête décisionnelle 3	50
Figure 52 la requête décisionnelle 4	50
Figure 53 la requête décisionnelle 5	51
Figure 54 la requête décisionnelle 6	51
Figure 55 la requête décisionnelle 7	52
Figure 56 data mining extraction des données	55
Figure 57 data mining transfert des données en arff	55
Figure 58 data mining chargement des données dans weka	56
Figure 59 data mining phase de classification	56
Figure 60 data mining phase de classification 2	57
Figure 61 data mining PDI	58
Figure 62 data mining PDI en détails	58
Figure 63 le tableau de bord de Northwind	60
Figure 64 tableau de bord de Northwind partie 1	61
Figure 65 tableau de bord partie 2	62
Figure 66 tableau de bord partie 3	62
Figure 67 tableau de bord partie 4	62

# Chapitre 1 :

## Présentation du projet

### INTRODUCTION :

Ce projet est une conception et implémentation d'un datawarehouse sur la base de données "Northwind", **datawarehouse** est une base de données dédiée au stockage de l'ensemble des données utilisées dans le cadre de la prise de décision et de l'analyse décisionnelle. Le Data Warehouse est exclusivement réservé à cet usage.

*Le Data Warehouse n'est pas une simple copie des données de production. Le data warehouse est organisé et structuré.*

Père du concept, Bill Immon dans son livre "Building the Data Warehouse" (John Wiley and Son 1996) le décrit ainsi :

*"Subject oriented, integrated, nonvolatile, time variant collection of data in support of management decisions."*

### NORTHWIND :

La base de données Northwind est un exemple de base de données utilisée par Microsoft pour illustrer les fonctionnalités de certains de ses produits, notamment SQL Server et Microsoft Access. La base de données contient les données sur les ventes de Northwind Traders, une société d'importation et d'exportation d'aliments de spécialité fictifs.

Bien que le code enseigné dans cette classe ne soit pas spécifique aux produits Microsoft, nous utilisons la base de données Northwind pour beaucoup de nos exemples parce que beaucoup de personnes le connaissent déjà et parce qu'il existe de nombreuses ressources pour l'apprentissage connexe.

Le diagramme ci-dessous montre la structure de la table de base de données Northwind.

# PRESENTATION DU PROJET

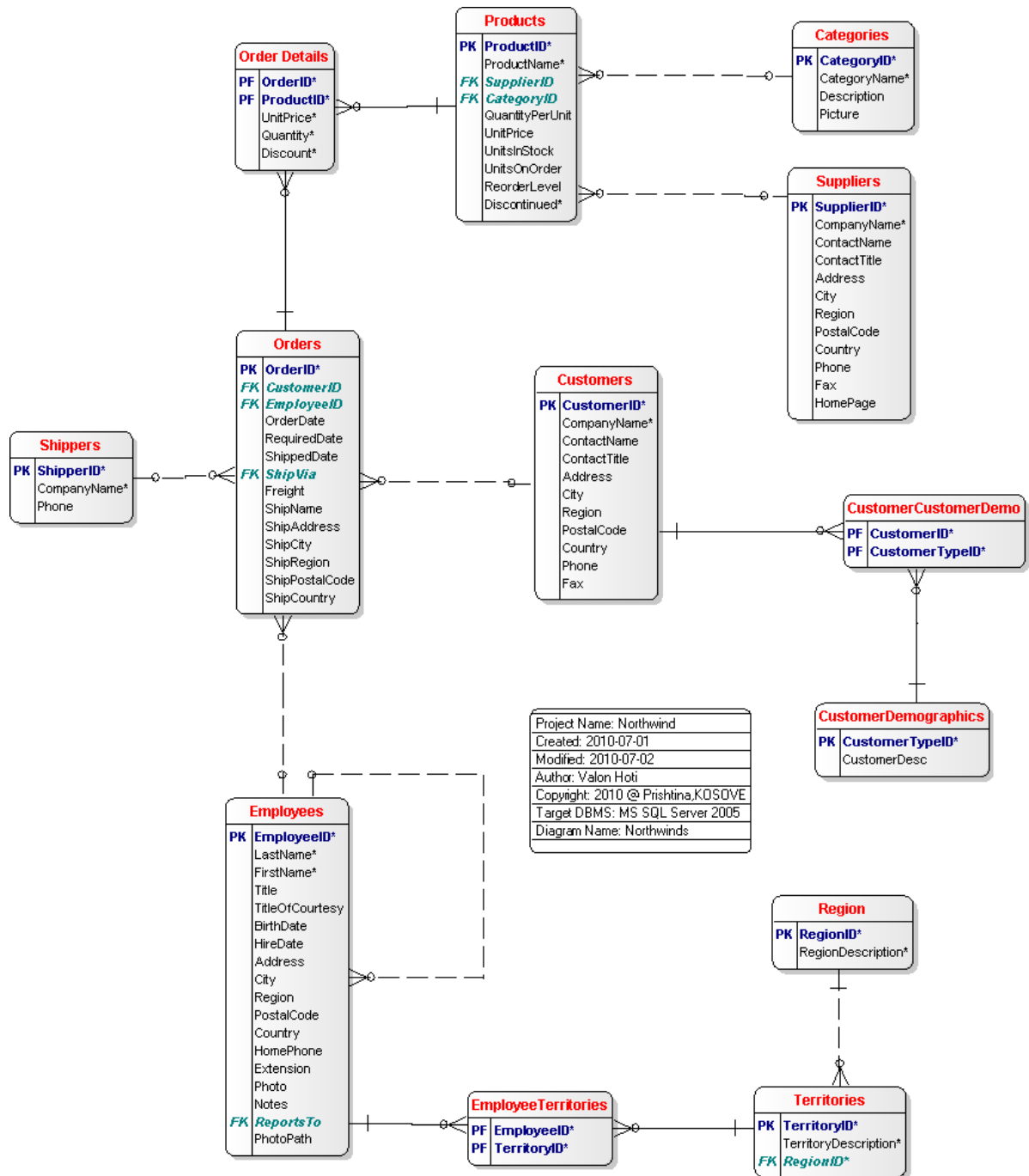


Figure 1 la structure de la table de base de données Northwind



### PENTAHO :

Pentaho est une solution d'informatique décisionnelle open source entièrement développée en Java. Elle porte sur toute la chaîne décisionnelle.

Il possède donc, les caractéristiques d'une couverture globale des fonctionnalités de la **Business Intelligence**, qui sont les suivantes :

- Reporting,
- Tableaux de bord,
- Analyse ad hoc,
- Analyse multidimensionnelle (OLAP),
- Intégration de données,
- Data Mining.




# Chapitre 2 :

Design du schéma logique et  
dimensionnel

## 2.1 Analyser des business drivers et des objectifs métiers

Les business drivers ou les pilotes de croissance sont ce qui va influencer l'atteinte des résultats et le succès de cette entreprise dans son marché.

Pour structurer notre réponse nous proposons de présenter les business drivers sous 3 forme :

-  **Le personnel** : les gestionnaires préparent et identifient les objectifs , ils prennent des décisions critiques et sont en innovation pour faire progresser l'entreprise, ils doivent avoir de la résistance et de la vigueur d'affaires et ils doivent être utiles pour bien gérer leur affaires associées.  
Et cela est offert par une bonne gestion et une conduite à la hauteur des données correspondantes aux employées, cette gestion peut être basée sur la capacité de construire de nouvelles relations commerciales, le moral des employés (suivre des heures supplémentaires volontaires, l'absentéisme et de congés de maladie) etc.
-  **Les produits** : pour augmenter les bénéfices de l'entreprise c'est très important d'observer le changement des ventes par rapport au temps et par rapport à la région, le suivi des produits intéressés par un client ou une région représentatif des clients qui y appartiennent et le développement des commandes dans une période de l'année.  
On doit pas oublier la gestion des remises offertes aux clients fidèles qui est liée aux coûts des produits voulus par ces clients, leurs activités, leurs motivations et les saisons mortes.
-  **La cible** : c'est un critère intéressant tant que l'entreprise a un contact direct aux clients, eux qui font de cette entreprise son existence. Ce qui nous mène à prendre en considération le suivi des clients et leurs activités au sein de l'organisation afin d'avoir la possibilité de piloter le système de vente et augmenter la performance.

Les objectifs métier indiquent ce que l'entreprise souhaite accomplir. Ils permettent de passer du système présent au système futur.

Liste des objectifs métier :

- Améliorer la satisfaction des clients via le profilage, la personnalisation et l'analyse de la valeur du capital client.
- Augmenter la productivité des procédures d'approvisionnement et d'achat via une analyse des dépenses.
- Minimiser les coûts par rapport au marché et par rapport aux concurrents.
- Garantir les conditions convenables au personnel.
- Prédéfinir les changements sur le marché et la cible.
- Classer les clients par fidélité.
- Minimiser les temps de réponse et de livraison.

## 2.2 Identifier les processus métiers clés, KPI et éventuelles requêtes décisionnelles

processus métiers clés:

- ✓ Processus client : **La vente des produits** (commande) PM1
- ✓ Processus support , (responsable de la valeur ajoutée aux clients) : **La mise à jour du catalogue produit** PM2
- ✓ Processus interne : **Roulement de personnel ou le processus d'embauche.** PM3

Les facteurs d'évaluation :

1. Les fichiers source .
2. Relation PM avec l'activité.

Les valeurs ajoutés aux processus métier :

1. Réalisables / Moins réalisables
2. PM le plus important dans l'activité

Diagramme de priorisation des processus :

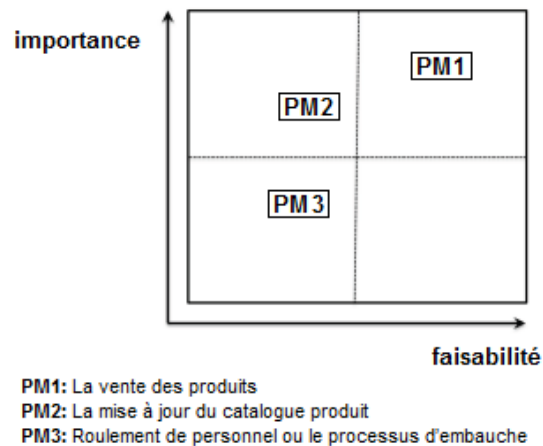


Figure 2 diagramme de priorisation des processus

Conclusion : le processus métier le plus faisable et le plus important dans l'activité est : la vente des produits –**commande**

## KPI:

- ✓ Le nombre de clients.
- ✓ Le coût d'un produit.
- ✓ Profits,
- ✓ Taux de rétention,
- ✓ Délai moyen de livraison

## requêtes décisionnelles:

- Quelle est la durée moyenne d'expédition par pays ?
- Quels sont les produits les plus commandés par un client ?
- Quel est le client actifs d'une année ?
- Quels sont les produits les plus commandés dans une région à une saison ?
- Quelle est la durée moyenne d'expédition par chaque expéditeur ?
- Quels sont les montants des commandes traitées par employé et/ou sous sa responsabilité ?
- Quels sont les clients ayant commander le plus ?

## 2.3 Proposer et implémenter un schéma dimensionnel pour le datawarehouse/datamart :

Après le choix du processus métier, on a identifié les dimensions :


Pour accomplir cette tâche on a :

- Définit le granularité :

Pour une base de données contenant commande, produits, client, employé, fournisseur et expéditeur, en ajoutant deux tables région et temps, on a comme grain : produit fournit par un fournisseur, vendu par un employé et livré par un expéditeur à un client dans une région chaque jour.

- Créé les dimensions :

Matrice en bus de données



Produit	Client	Employé	Fournisseur	Expéditeur	Région	Temps
nom_catégorie	nom_entreprise	superviseur	fonction_fournisseur	nom_expéditeur	pays	date
nom_produit	fonction_client	fonction_employé	nom_fournisseur		ville	year
prix_produit	nom_client	nom_employé			code_postal	saison
						mois
						semaine
						jour
les mesures de la table de faits: quantité, prix, montant						

Figure 3 matrice en bus des données des dimensions

Le schéma en étoile :

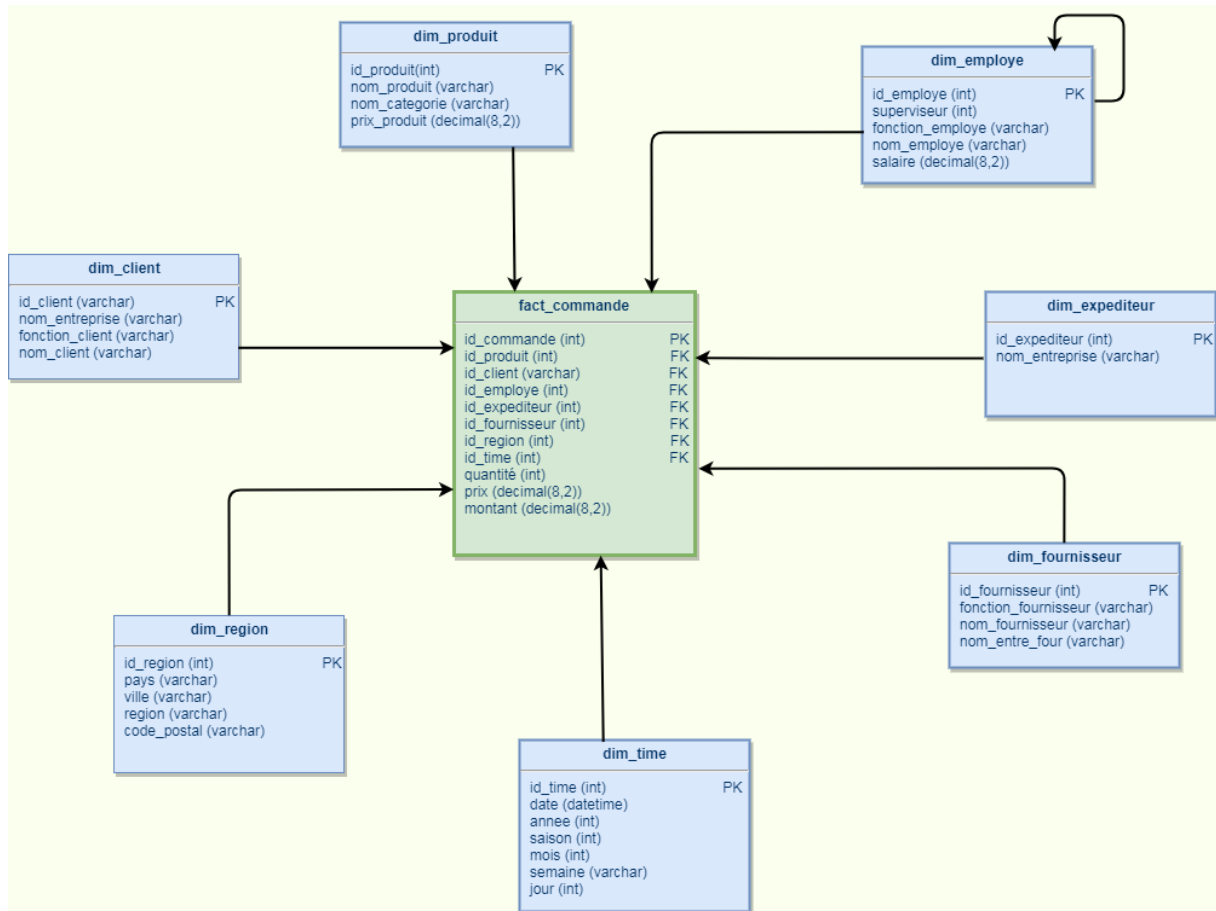


Figure 4 schéma en étoile pour le Datamart

Description :

- ✚ **La table de faits (fact\_commande)** contient en plus de la clé primaire (id\_commande), les mesures qui représentent les différents attributs calculables qui vont répondre aux différentes requêtes décisionnelles et les clés étrangères qui la lient avec les dimensions.

✚ Les dimensions qui fournissent le contexte :

- ✚ **Client (dim\_client)** : contient la clé primaire et une hiérarchie de nom\_entreprise >> fonction\_client >> nom\_client.
- ✚ **Produit (dim\_produit)** : une clé primaire qui lie la dimension avec la table de faits, une hiérarchie nom\_categorie >> nom\_produit et l'attribut de prix\_produit construit cette dimension.
- ✚ **Employé (dim\_employee)** : comme toute autre dimension elle contient la clé primaire qui la définit afin de se connecter à fact\_commande mais celle-ci inclut une bidule particulière qui n'existe pas dans les autres c'est la hiérarchie prent-children qui relie l'employé avec son superviseur dans un seul niveau ce qui nous aidera à récupérer les commandes faites par une équipe où il y a un superviseur.
- ✚ **Expéditeur (dim\_expediteur)** : une dimension simple qui nous aidera à lier la commande avec la livraison.
- ✚ **Fournisseur (dim\_fournisseur)** : dimension contient une clé primaire et une hiérarchie
- ✚ **Région (dim\_région)** : représente la dimension de localisation pour la table de faits.
- ✚ **Temps (dim\_time)** : chaque schéma dimensionnel pour le data-warehouse doit avoir le celle-ci comme une dimension afin d'exécuter les requêtes dans leur contexte temporaire.

## **pourquoi on a choisit le schéma en étoile ?**

- tout simplement c'est parce que le schéma en étoile est plus performant dû au fait qu'il y a moins de jointures à faire que sur un modèle en flocons et est plus simple que le schéma en flocons.

## 2.4 Estimer la taille du datamart/ datawarehouse :

Pour calculer la taille du datamart on a besoin de calculer la taille de chaque table et puis estimer la taille de la table de faits, et pour calculer la taille d'une table de dimension on aura besoin de la taille de chaque attribut.  
Etant donné chaque valeur d'une ligne a sa propre taille il va falloir prendre la taille max.



## Date and Time Type Storage Requirements

For [TIME](#), [DATETIME](#), and [TIMESTAMP](#) columns, the storage required for tables created before MySQL 5.6.4 differs from tables created from 5.6.4 on. This is due to a change in 5.6.4 that permits these types to have a fractional part, which requires from 0 to 3 bytes.

Data Type	Storage Required Before MySQL 5.6.4	Storage Required as of MySQL 5.6.4
<a href="#">YEAR</a>	1 byte	1 byte
<a href="#">DATE</a>	3 bytes	3 bytes
<a href="#">TIME</a>	3 bytes	3 bytes + fractional seconds storage
<a href="#">DATETIME</a>	8 bytes	5 bytes + fractional seconds storage
<a href="#">TIMESTAMP</a>	4 bytes	4 bytes + fractional seconds storage

## Numeric Type Storage Requirements

Data Type	Storage Required
<a href="#">TINYINT</a>	1 byte
<a href="#">SMALLINT</a>	2 bytes
<a href="#">MEDIUMINT</a>	3 bytes
<a href="#">INT</a> , <a href="#">INTEGER</a>	4 bytes
<a href="#">BIGINT</a>	8 bytes
<a href="#">FLOAT</a> ( <i>p</i> )	4 bytes if $0 \leq p \leq 24$ , 8 bytes if $25 \leq p \leq 53$
<a href="#">FLOAT</a>	4 bytes
<a href="#">DOUBLE</a> [ <a href="#">PRECISION</a> ], <a href="#">REAL</a>	8 bytes
<a href="#">DECIMAL</a> ( <i>M</i> , <i>D</i> ), <a href="#">NUMERIC</a> ( <i>M</i> , <i>D</i> )	Varies; see following discussion
<a href="#">BIT</a> ( <i>M</i> )	approximately $(M+7)/8$ bytes

Values for [DECIMAL](#) (and [NUMERIC](#)) columns are represented using a binary format that packs nine decimal (base 10) digits into four bytes. Storage for the integer and fractional parts of each value are determined separately. Each multiple of nine digits requires four bytes, and the “leftover” digits require some fraction of four bytes. The storage required for excess digits is given by the following table.

## String Type Storage Requirements

In the following table, *M* represents the declared column length in characters for nonbinary string types and bytes for binary string types. *L* represents the actual length in bytes of a given string value.

Data Type	Storage Required
<a href="#">CHAR</a> ( <i>M</i> )	$M \times w$ bytes, $0 \leq M \leq 255$ , where <i>w</i> is the number of bytes required for the maximum-length character in the character set. See <a href="#">Section 14.8.1.2, “The Physical Row Structure of an InnoDB Table”</a> for information about <a href="#">CHAR</a> data type storage requirements for InnoDB tables.
<a href="#">BINARY</a> ( <i>M</i> )	<i>M</i> bytes, $0 \leq M \leq 255$
<a href="#">VARCHAR</a> ( <i>M</i> ), <a href="#">VARBINARY</a> ( <i>M</i> )	<i>L</i> + 1 bytes if column values require 0 – 255 bytes, <i>L</i> + 2 bytes if values may require more than 255 bytes
<a href="#">TINYBLOB</a> , <a href="#">TINYTEXT</a>	<i>L</i> + 1 bytes, where $L < 2^8$
<a href="#">BLOB</a> , <a href="#">TEXT</a>	<i>L</i> + 2 bytes, where $L < 2^{16}$
<a href="#">MEDIUMBLOB</a> , <a href="#">MEDIUMTEXT</a>	<i>L</i> + 3 bytes, where $L < 2^{24}$
<a href="#">LONGBLOB</a> , <a href="#">LONGTEXT</a>	<i>L</i> + 4 bytes, where $L < 2^{32}$
<a href="#">ENUM</a> ('value1', 'value2', ...)	1 or 2 bytes, depending on the number of enumeration values (65,535 values maximum)
<a href="#">SET</a> ('value1', 'value2', ...)	1, 2, 3, 4, or 8 bytes, depending on the number of set members (64 members maximum)

Figure 5 documentation sur la taille des variables sous mysql

## Dimension client : 93 lignes

Cette table contient 4 attributs de type varchar tel que :

- Id\_client : varchar(5)
- Nom\_entreprise : varchar(50)
- Fonction\_client : varchar(50)
- Nom\_client : varchar(50)

TailleMax(ligne) = TailleMax(Id\_client) + TailleMax(nom\_entreprise) +  
TailleMax(fonction\_client) + TailleMax(nom\_client)  
= 6+51+51+51  
=159octets.

**TailleMax(dim\_client) = 93 \* tailleMax(ligne)**  
**= 14 787 octets**

## Dimension employé : 9 lignes

Cette table contient 2 attributs de type int et 2 de type varchar tel que :

- Id\_employe : int(11)
- Superviseur : int(11)
- Fonction\_employe : varchar(50)
- Nom\_employe : varchar(50)

TailleMax(ligne) = TailleMax(Id\_employe) + TailleMax(superviseur) +  
TailleMax(fonction\_employe) + TailleMax(nom\_employe)  
=11\*4+11\*4+51+51  
= 190octets.

**TailleMax(dim\_employe) = 9 \* tailleMax(ligne)**  
**= 1 710 octets**

## Dimension expéditeur : 3 lignes

Cette table contient un attribut de type int et un autre de type varchar tel que :

- Id\_expediteur : int(11)
- Nom\_entreprise : varchar(50)

TailleMax(ligne) = TailleMax(Id\_expediteur) + TailleMax(nom\_expediteur)  
=11\*4+51  
= 95octets.

**TailleMax(dim\_expediteur) = 3 \* tailleMax(ligne)**  
**= 285 octets**

## Dimension fournisseur : 29 lignes

Cette table contient un attribut de type int, 3 de type varchar tel que :

- Id\_fournisseur : int(11)
- Fonction\_fournisseur : varchar(50)
- Nom\_fournisseur : varchar(50)
- Nom\_entreprise\_fournisseur : varchar(50)

$$\begin{aligned}\text{TailleMax(ligne)} &= \text{TailleMax(Id\_fournisseur)} + \text{TailleMax(fonction\_fournisseur)} + \\ &\text{TailleMax(nom\_fournisseur)} + \text{TailleMax(nom\_entreprise\_fournisseur)} \\ &= 11*4 + 51 + 51 + 51 \\ &= 197 \text{ octets.}\end{aligned}$$

$$\begin{aligned}\text{TailleMax(dim\_expediteur)} &= 29 * \text{tailleMax(ligne)} \\ &= 5\,713 \text{ octets}\end{aligned}$$

## Dimension produit : 77 lignes

Cette table contient un attribut de type int, 2 de type varchar et un de type float tel que :

- Id\_produit : int(11)
- Nom\_produit : varchar(50)
- Nom\_categorie : varchar(50)
- Prix\_produit : float

$$\begin{aligned}\text{TailleMax(ligne)} &= \text{TailleMax(Id\_produit)} + \text{TailleMax(nom\_produit)} + \\ &\text{TailleMax(nom\_categorie)} + \text{TailleMax(prix\_produit)} \\ &= 11*4 + 51 + 51 + 4 \\ &= 150 \text{ octets.}\end{aligned}$$

$$\begin{aligned}\text{TailleMax(dim\_produit)} &= 77 * \text{tailleMax(ligne)} \\ &= 11\,550 \text{ octets}\end{aligned}$$

## Dimension région : 85 lignes

Cette table contient un attribut de type int et 5 de type varchar tel que :

- Id\_region : int(11)
- Pays : varchar(15)
- Ville : varchar(15)
- Region : varchar(15)
- Code\_postal : varchar(10)

$$\begin{aligned}\text{TailleMax(ligne)} &= \text{TailleMax(Id\_region)} + \text{TailleMax(pays)} + \text{TailleMax(ville)} + \\ &\text{TailleMax(region)} + \text{TailleMax(code\_postal)} \\ &= 11*4 + 16 + 16 + 16 + 16 + 11 \\ &= 119 \text{ octets.}\end{aligned}$$

$$\begin{aligned}\text{TailleMax(dim\_region)} &= 77 * \text{tailleMax(ligne)} \\ &= 9\,163 \text{ octets}\end{aligned}$$

## Dimension temps : 675 lignes

Cette table contient 5 attributs de type int, 4 de type varchar et un attribut de type datetime tel que :

- Id\_time : int(11)
- Date : datetime
- Year : int(11)
- Saison : int(11)
- Mois : int(11)
- Jour : int(11)
- Saison\_label : varchar(25)
- Mois\_label : varchar(25)
- Jour\_label : varchar(25)
- Semaine : varchar(50)

$$\begin{aligned}\text{TailleMax(ligne)} &= \text{TailleMax(Id\_time)} + \text{TailleMax(Date)} + \text{TailleMax(Year)} + \\ &\text{TailleMax(Saison)} + \text{TailleMax(Mois)} + \text{TailleMax(Jour)} + \text{TailleMax(Saison\_label)} + \\ &\text{TailleMax(Mois\_label)} + \text{TailleMax(Jour\_label)} + \text{TailleMax(Semaine)} \\ &= 11*4+8+11*4+11*4+11*4+11*4+26+26+26+51 \\ &= 357 \text{ octets.}\end{aligned}$$

$$\begin{aligned}\text{TailleMax(dim\_temps)} &= 675 * \text{tailleMax(ligne)} \\ &= 133\,875 \text{ octets}\end{aligned}$$

## Estimation de la taille de la table de faits commande :

Une clé primaire, 7 clés externes et 5 mesures :

- Id\_commande : int(11)
- Id\_produit
- Id\_client
- Id\_employe
- Id\_expediteur
- Id\_fournisseur
- Id\_region
- Id\_time
- Quantite : int(11)
- Prix : float
- Remise : float
- Montant : float
- Nb\_jour : int(11)

Dimension client : 93 lignes  
Dimension employé : 9 lignes  
Dimension expéditeur : 3 lignes  
Dimension fournisseur : 29 lignes  
Dimension produit : 77 lignes  
Dimension région : 85 lignes  
Dimension temps : 675 lignes

Donc on a **nbreLigneMax** =  $93 \times 9 \times 3 \times 29 \times 77 \times 85 \times 675 = 321\ 705\ 239\ 625$  lignes

**TailleMax(ligne\_commande)** = TailleMax(id\_commande) + TailleMax(id\_produit) +  
TailleMax(id\_client) + TailleMax(id\_employe) + TailleMax(id\_expéditeur) +  
TailleMax(id\_fournisseur) + TailleMax(id\_region) + TailleMax(id\_temps) +  
TailleMax(quantité) + TailleMax(prix) + TailleMax(remise) + TailleMax(montant) +  
TailleMax(nb\_jour)  
 $= 4 \times 11 + 6 + 4 \times 11 + 4 \times 11 + 4 \times 11 + 4 \times 11 + 4 \times 11 + 4 \times 11 +$   
 $4 \times 11 + 4 + 4 + 4 + 4 \times 11$   
**= 414 octets**

**TailleMax(commande)** = **nbreLigneMax \* TailleMax(ligne\_commande)**  
**= 133 185 969 204 750 octets**

La taille maximale du datamart et la somme de la taille maximale de la table de faits et la taille des tables de dimensions :

**TailleMax(datamart)** = 14 787 octets + 1 710 octets + 285 octets + 5 713 octets + 11 550 octets + 9 163 octets + 133 875 octets + 133 185 969 204 750 octets  
**= 133 185 969 381 833 octets**  
**= 133 To 185 Go 969 Mo 381 Ko 833 o**

Donc on obtient une taille maximale du datamart très grande, et cela dû au fait que les variables sont déclarées avec une grand longueur.

# Chapitre 3 :

## Intégration des données

## INTRODUCTION :

Le principe est simple il s'agit d'alimenter les entrepôts de données. Les ETL s'occupent d extraire les données depuis leurs sources, les transformer et puis les charger.

### 3.1 Extraire les données depuis la base initiale :

La première étape concerne l'extraction des données et doit permettre de se connecter aux bases. Il est important, lorsque l'on extrait les données, de pouvoir les analyser, il faut donc connaître les propriétés de celle-ci, cela parût simple puisque la source provient du SGBD MySQL. L'extraction peut aussi s'occuper de vérifier les erreurs des sources.

L'étape d'extraction est donc très importante. Elle doit être performante et complète pour pouvoir disposer d'un outil ETL.

### 3.2 Effectuer les transformations nécessaires :

Cette seconde étape a pour objectif la transformation des données. Elle est bien évidemment indispensable si l'on veut obtenir des cibles différentes des sources.

Cette étape va permettre de joindre les différentes sources selon les clés. Elle va aussi permettre de filtrer les données.

Une partie importante de l'étape de transformation est de pouvoir effectuer des calculs.

En bref, cette étape doit permettre d'effectuer toutes les transformations que l'on souhaite appliquer aux données sources. Il ne faut pas oublier la sélection ou le découpage de colonnes, la traductions des valeurs, la fusion, la gestion des erreurs et encore de nombreuses autres fonctionnalités.

### 3.3 Charger les données dans le datamart / Datawarehouse proposé :

La dernière étape s'occupe de charger les données dans des cibles hétérogènes, le chargement va permettre d'insérer ou de mettre à jour les données cibles et comme dans les deux étapes précédentes, il doit aussi gérer les erreurs. Le chargement n'est pas à négliger pour un bon outil ETL, il doit être complet et performant.

### 3.4 Démonstration :

Pour la démonstration de notre projet nous vous proposons de suivre les prises d'écran qui expliqueront tout le déroulement de notre partie ETL :

D'abord on doit lier une connexion avec la base de données qu'on a créée pour contenir les données de notre data warehouse, cette base sous le nom de northwind\_dw. Cette connexion est de type MySQL.

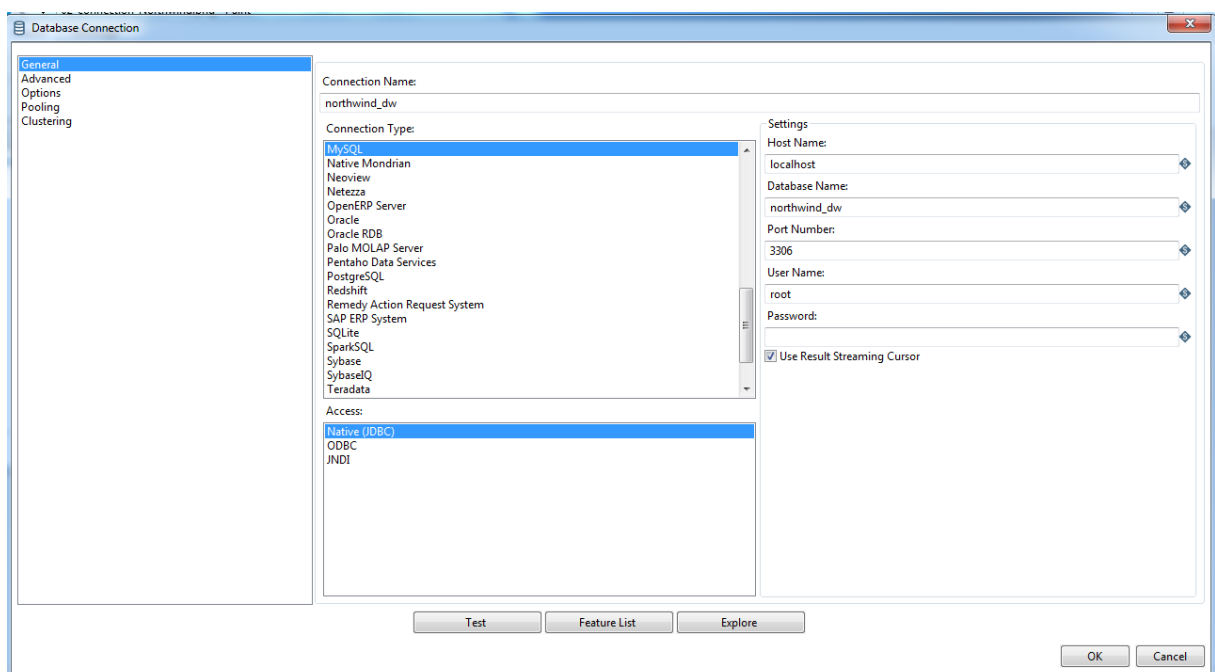


Figure 6 connexion à la base de donnée datawarehouse



Et pour extraire les données il faut établir une deuxième connexion mais cette fois avec une base existante sous le nom de northwind\_source, celle-ci contient les données qu'on a besoin d'analyser (notre source des données).

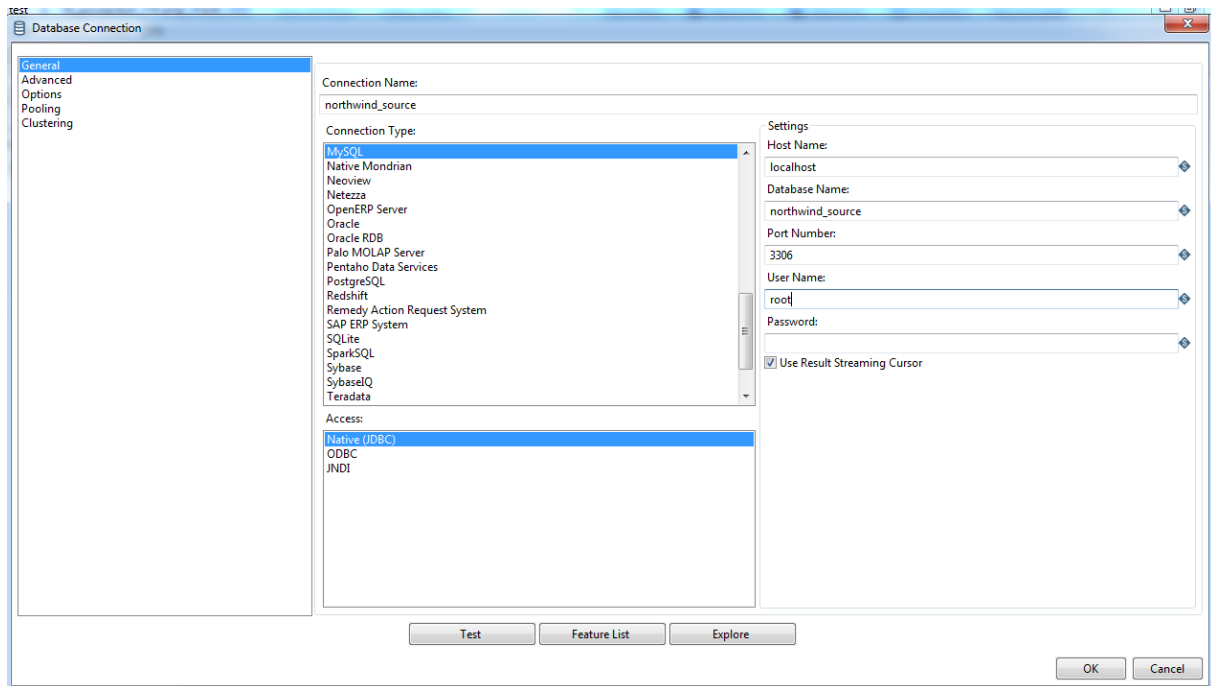


Figure 7 connexion à la base des données source

Et maintenant comme on a lié notre ETL avec les deux bases de données, on va extraire les données pour chaque dimension par des requêtes sql et les transformer pour avoir des dimensions bien fines en sortie.

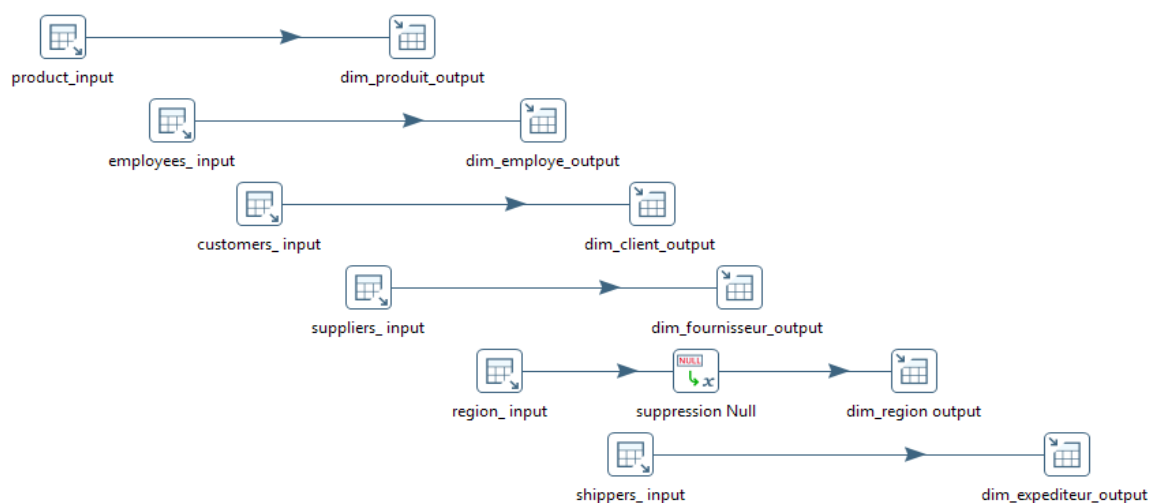


Figure 8 ETL création des dimensions

- Commençant par la dimension **Produit** :



Figure 9 ETL création de la dimension Produit

Pour plus de détails :

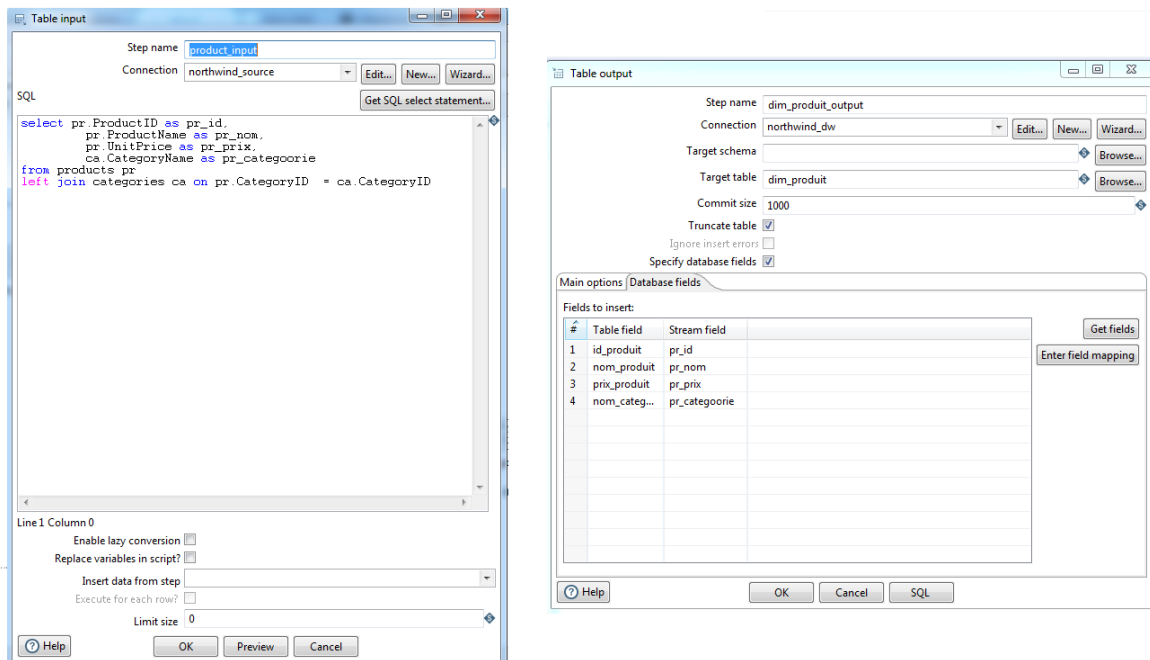


Figure 10 ETL création de la dimension Produit en détails

Description : la fenêtre à gauche a pour but d'exécuter la requête et la deuxième permet de remplir la dimension par les lignes extraites, c'est la table de sortie.

*Et c'est le même cas pour la plupart d'autres dimensions sauf que chacune a ses propres attributs et données.*

- Dimension **Employé**:



Figure 11 ETL création de la dimension Employe

Pour plus de détails :

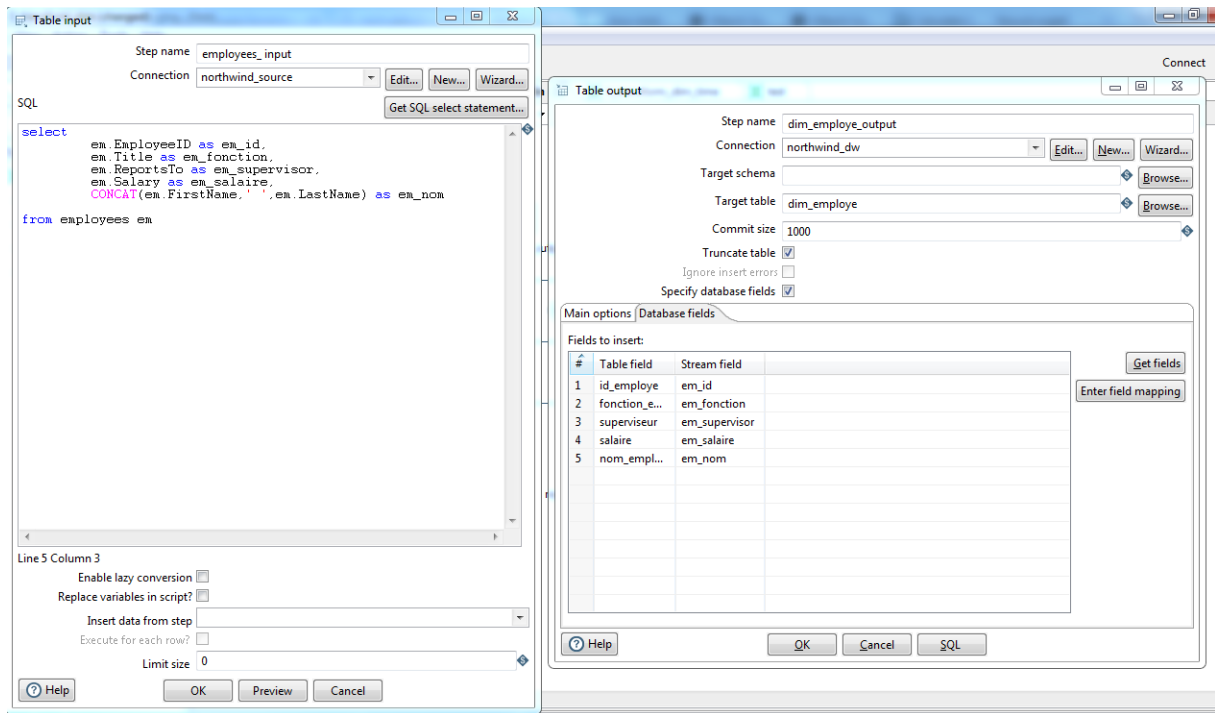


Figure 12 ETL création de la dimension Employé en détails

Descriptions : la dimension Employe a ses propres colonnes et une particularité c'est qu'une colonne (nom\_employe) est obtenue par la concaténation de deux autres colonnes (c'est indiqué dans la requête) et c'est une transformation faite au niveau de la phase d'extraction.

## ■ Dimension **Client**:



Figure 13 ETL création de la dimension Client

Pour plus de détails :

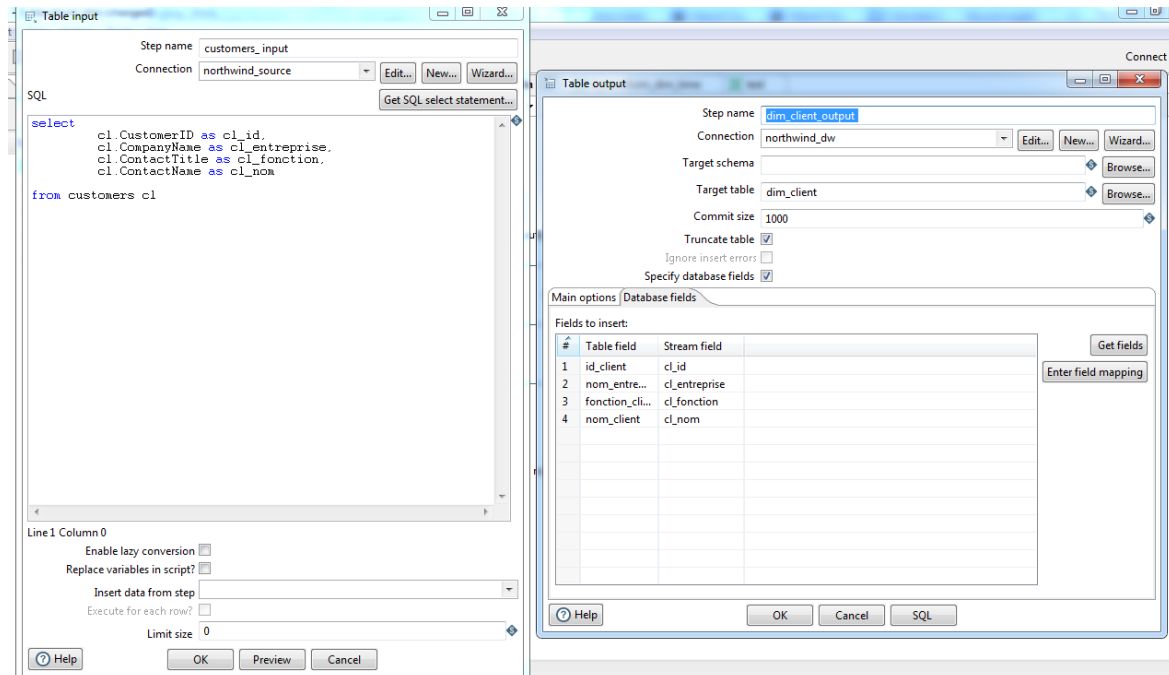


Figure 14 ETL création de la dimension Client en détails

Descriptions : une requête simple et chargement normal des données

## ■ Dimension Fournisseur :



Figure 15 ETL création de la dimension Fournisseur

Pour plus de détails :

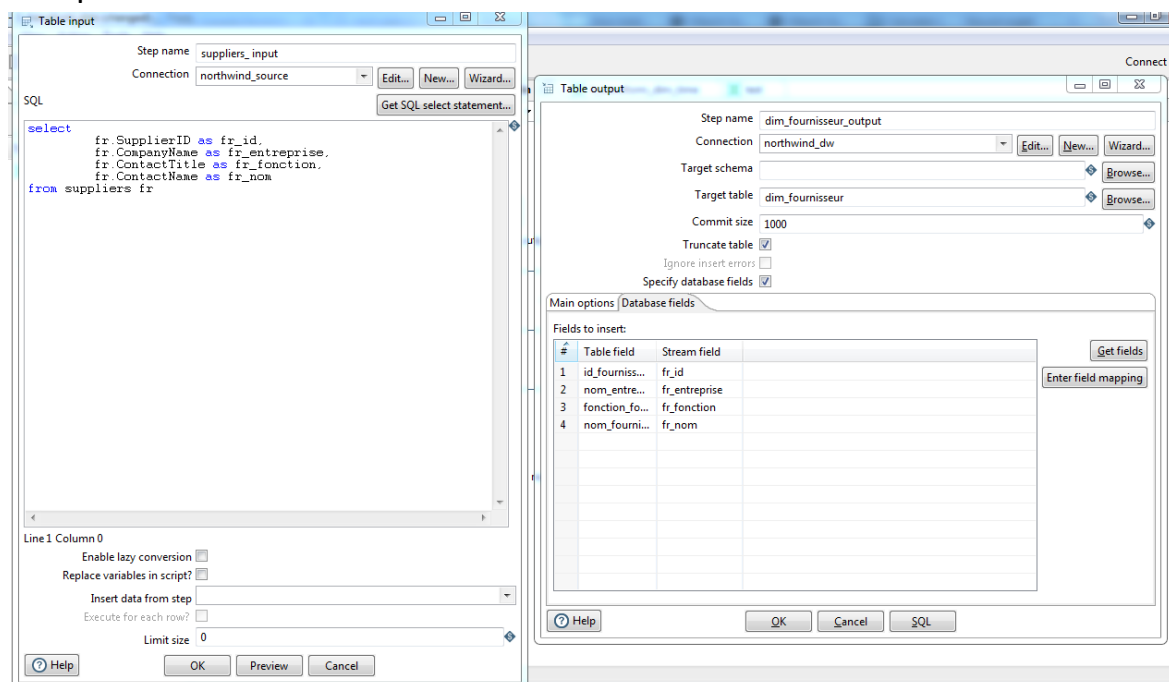


Figure 16 ETL création la dimension Fournisseur en détails

Description : rien de mystère, extraire les données et les charger c'est ce qu'on est entrain de faire pour chaque dimension et c'est pareil pour celle-ci.

### ■ Dimension **Expéditeur** :



Figure 17 ETL création de la dimension Expéditeur

Pour plus de détails :

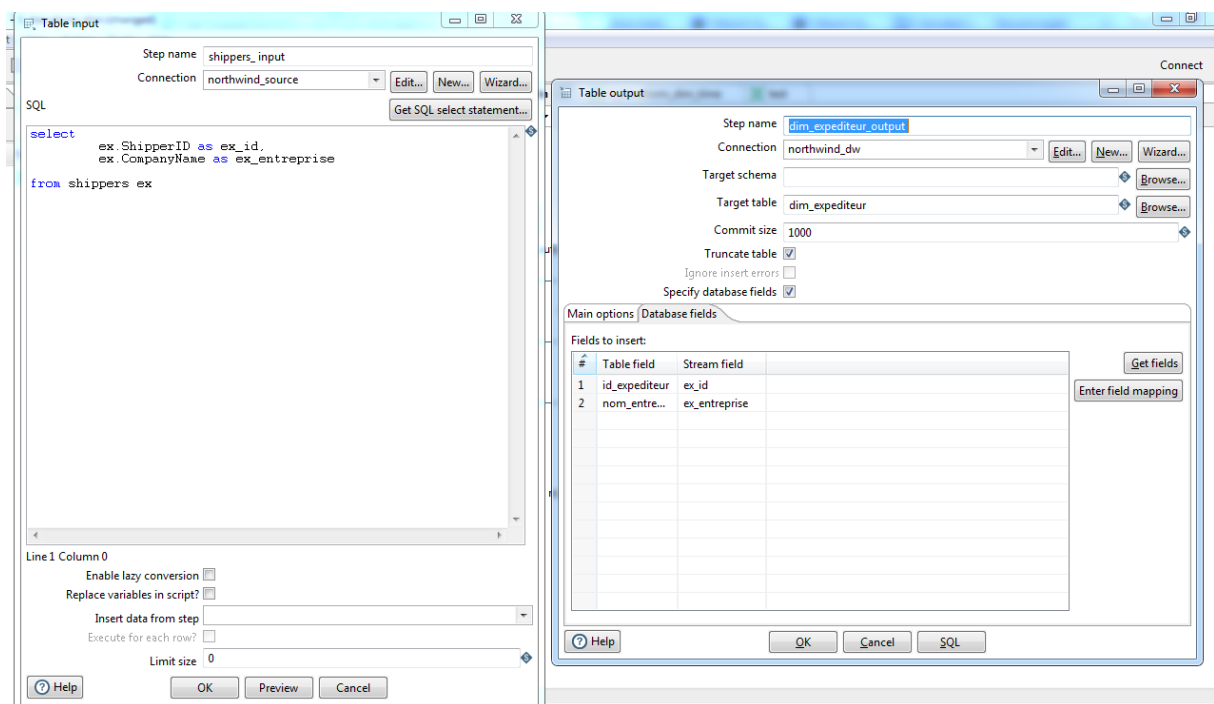


Figure 18 ETL création de la dimension Expéditeur en détails

Maintenant, après avoir expliqué comment on a construit les dimensions ayant les données extraites directement à partir de la source dans notre data warehouse, notre schéma a besoin de deux choses très importantes dans le contexte de notre analyse c'est le fait de pouvoir localiser les données et les mettre dans leurs contextes temporaires, d'où l'utilité de construire deux tables de dimensions, un peu différents des autres.

En bref, la table qui représente la localisation n'existe pas dans notre base de source mais ses données sont extraites à partir d'une requête qui sélectionne juste les colonnes représentant la géolocalisation de la table commande. L'autre table est un peu plus compliquée, cette fois-ci on a construit une dimension sans avoir récupéré une seule valeur du Northwind\_source. On verra cela en détails ci-dessous.

## ■ Dimension **Région** :



Figure 19 ETL création de la dimension Région

Sur cette table, on a effectué une transformation en supprimant les valeurs nulles

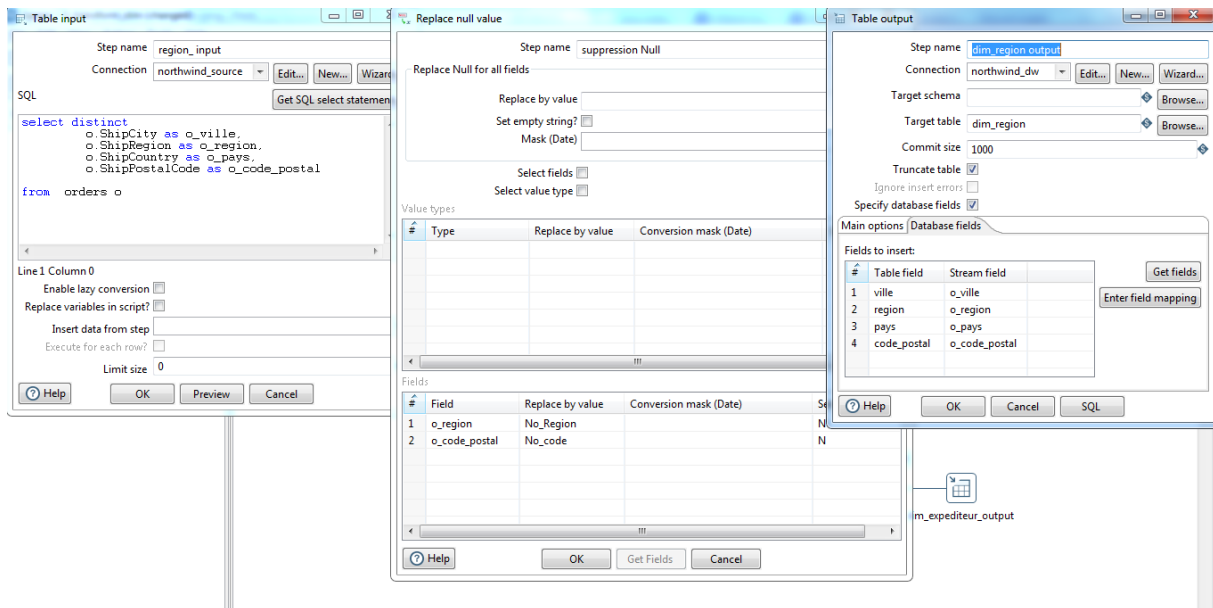
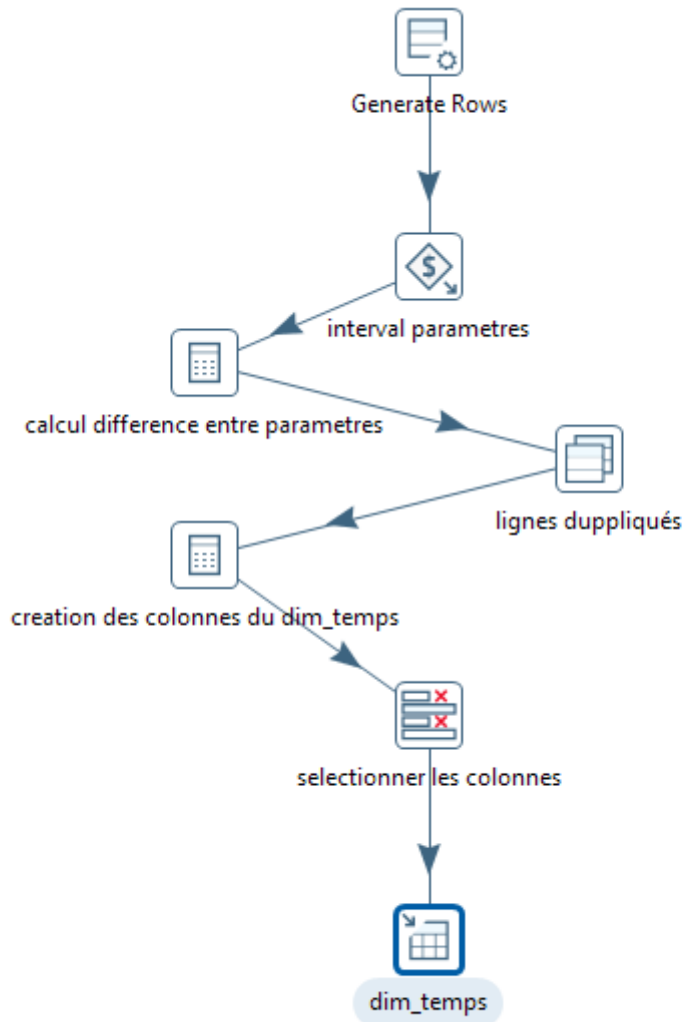


Figure 20 ETL création de la dimension région en détails

Descriptions : la fenêtre en milieu illustre une transformation par suppression des valeurs nulles appliquée sur les colonnes région et code\_postal en remplaçant leurs données nulles respectivement par no\_region et no\_code. La dernière étape c'est comme pour tous les autres dimensions on doit mapper les colonnes pour charger les données dans la dimension Région.

- **Dimension Temps :**



### Figure 21 ETL création de la dimension Temps

Etant donnée la différence de la dimension de temps avec les autres, nous allons détailler en plus notre processus de sa construction.

D'abord, on comprendra le principe de ce processus. La construction de cette dimension se fait soit en remplissant la table par une requête en sql qui remplit les données de toute la période des commandes faites au sein de l'entreprise, soit en créant une transformation qui génère automatiquement les données pour la dimension de temps. Vu que la deuxième méthode est plus pratique, automatique et exerçant le concept d'ETL mieux que l'autre méthode, on a choisit de suivre la deuxième pratique.

## Etape 1 :

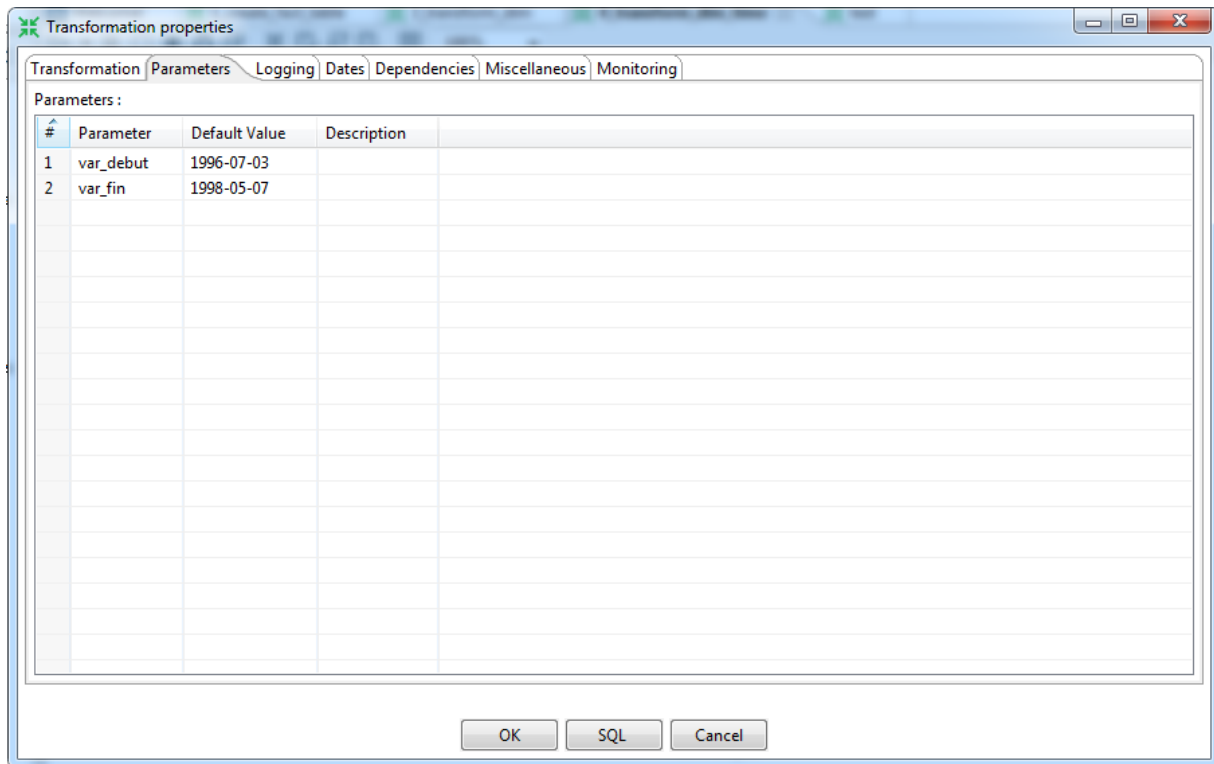


Figure 22 ETL création des paramètre dans la dimension temps

Descriptions : créer deux paramètres var\_debut et var\_fin contenant respectivement la date initiale et la date finale.

## Etape 2 :

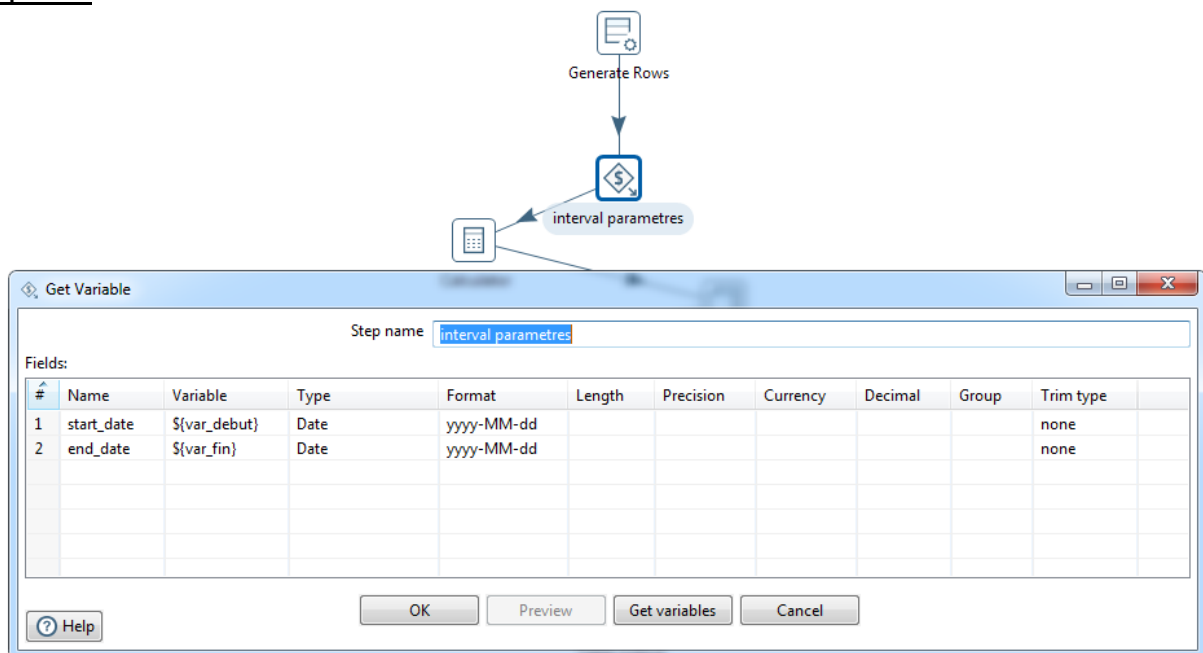


Figure 23 ETL création des paramètres dans la dimension temps





## Etape 4 :

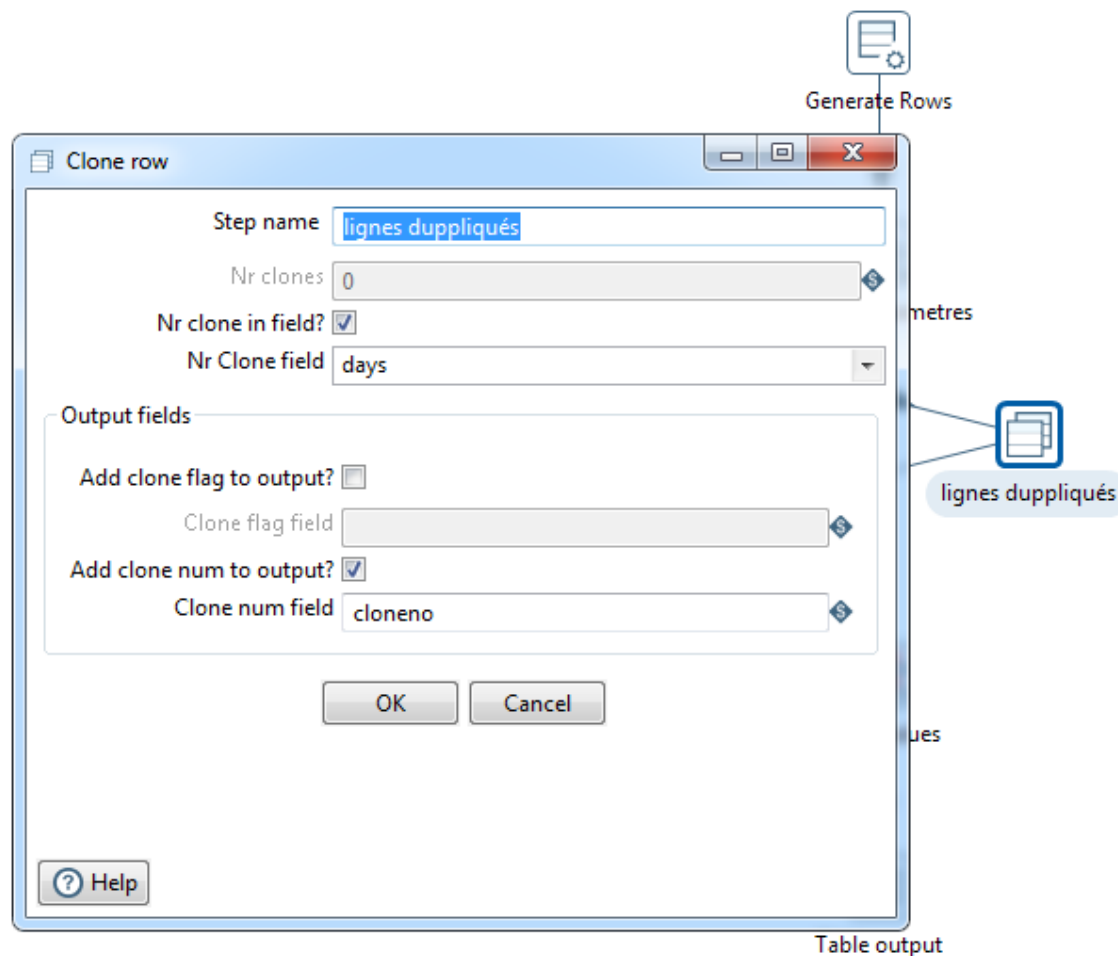


Figure 25 ETL duplication de la valeur calculée de la dimension temps

Description : dupliquer la valeur calculée sur les lignes de la dimension de temps.

## Etape 5 :

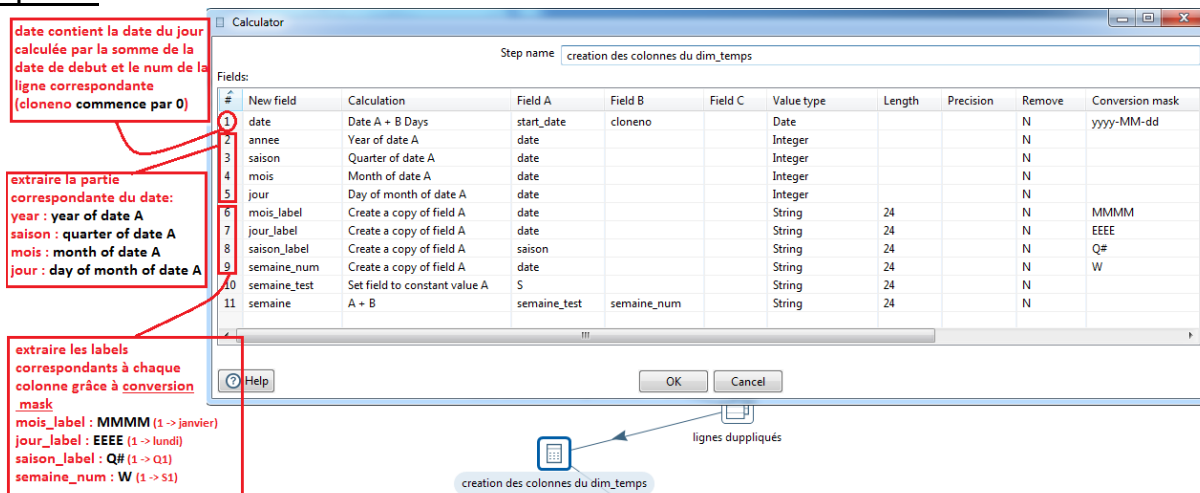


Figure 26 ETL la dimension temps avant le mapping

Description : l'illustration décrit bien cette étape, chaque colonne est calculée par une fonction donnée, si par exemple on veut des une colonne qui contient les jours par semaine on doit utiliser la fonction « day of week of date A » et ainsi de suite. Donc cette table contient 11 attributs chacun correspond à une valeur précisé qui va aider lors de notre analyse.

### Etape 6 :

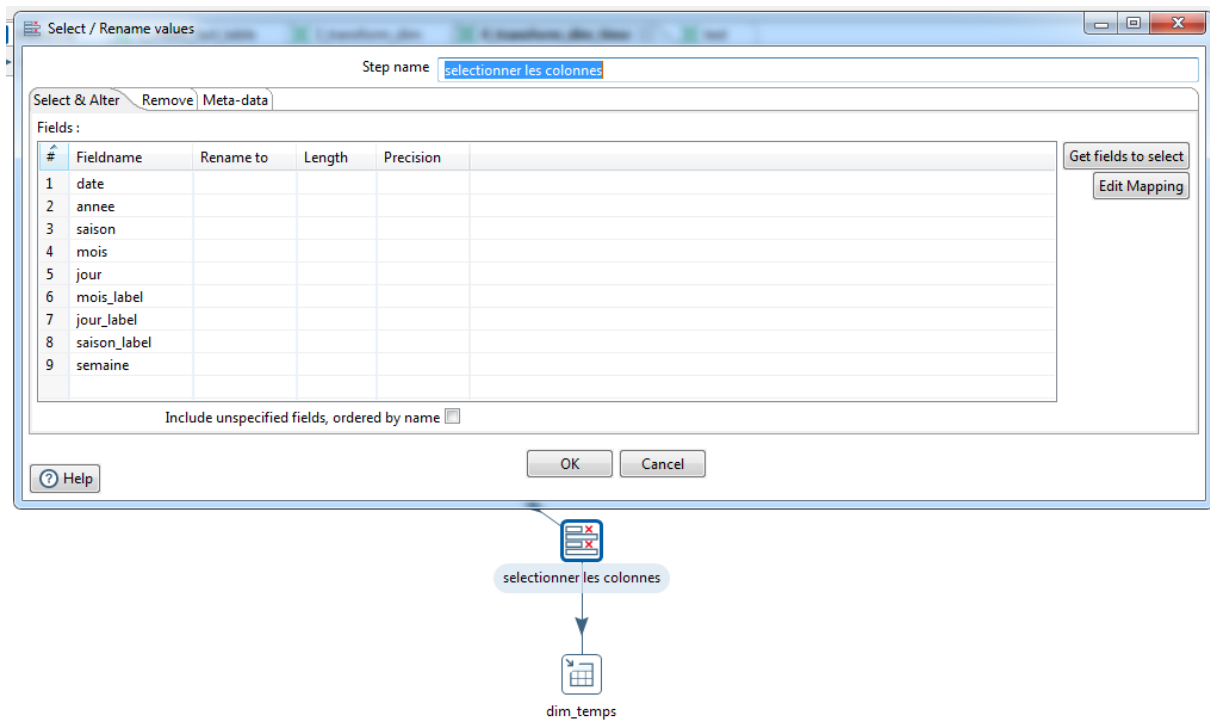


Figure 27 ETL sélection des colonnes de la dimension temps

Description : cette étape consiste à sélectionner ce qu'on a besoin, après l'élimination des colonnes négligeables et éventuelles, notre table se limite sur une hiérarchie qui facilitera notre lecture à partir du dashboard (le tableau de bord).

### Etape 7 :

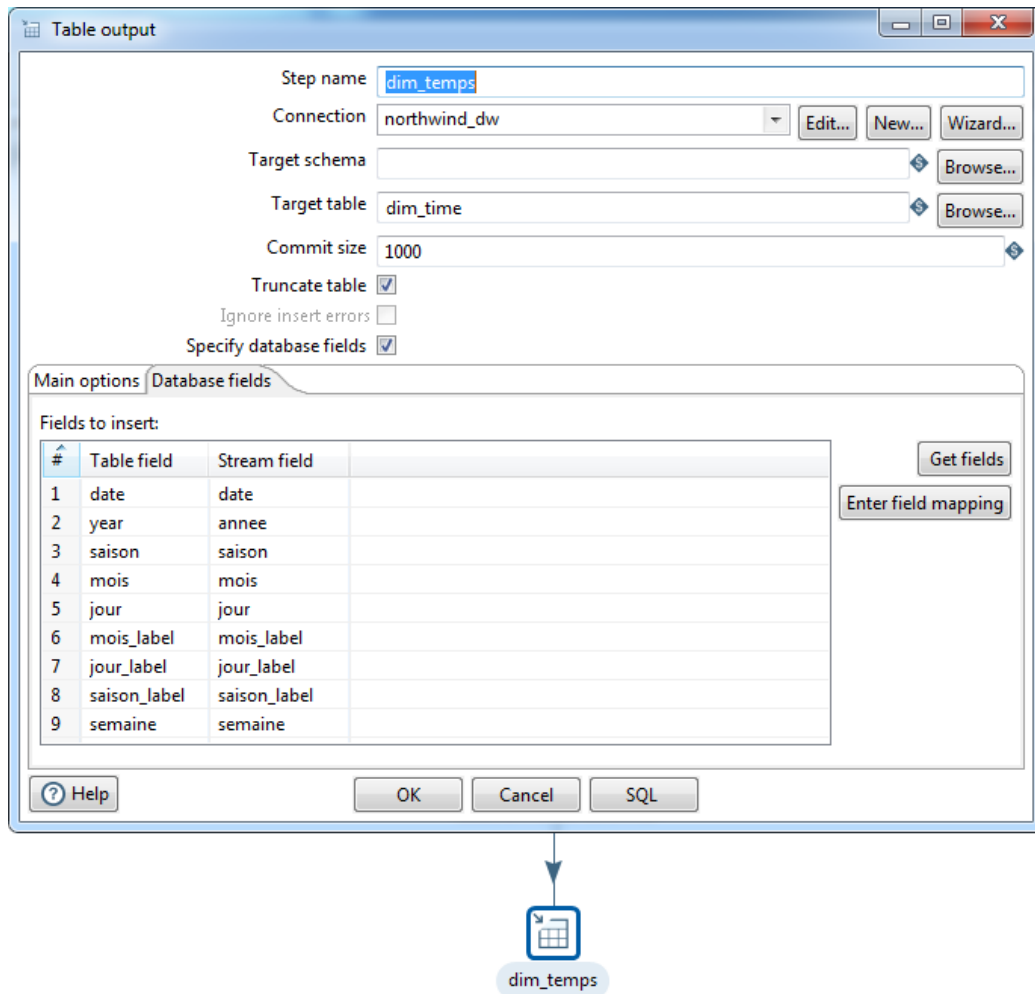


Figure 28 ETL chargement des données dans la dimension temps

Description : à la fin de cet ensemble de transformations sur la table courante, il est temps de charger les données sur la dimension de temps, d'autre terme mapper les colonnes précédentes avec les nouvelles colonnes.

Après avoir finir la partie ETL sur les dimensions, on est sur le point de terminer. La dernière tâche dans cette partie de PDI est de faire l'ETL sur la table de faits.

Premièrement on a extrait les données à partir d'une requête sql en important les données depuis la base de source, cette requête contient les clés des autres dimension et quelques mesures, puis on y exerce les transformations nécessaires pour les charger dans notre nouvelle base.

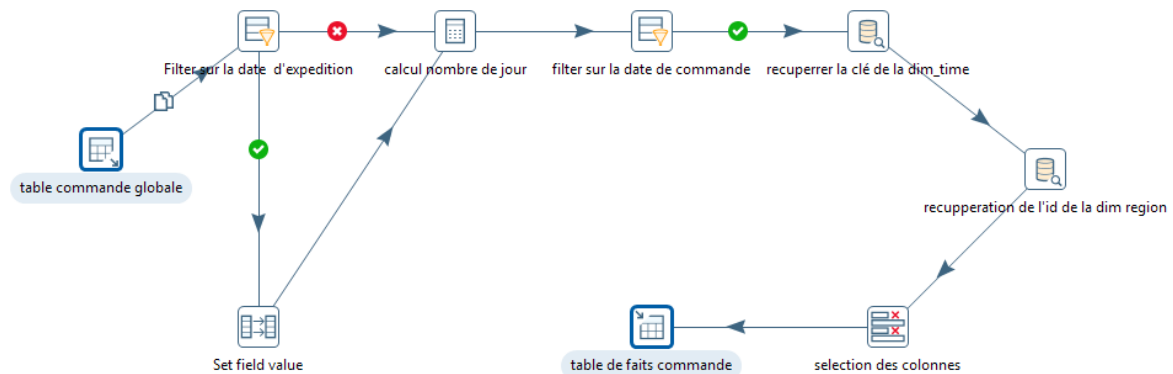


Figure 29 ETL création de la table de faits commande

## Etape 1 : extraction des données

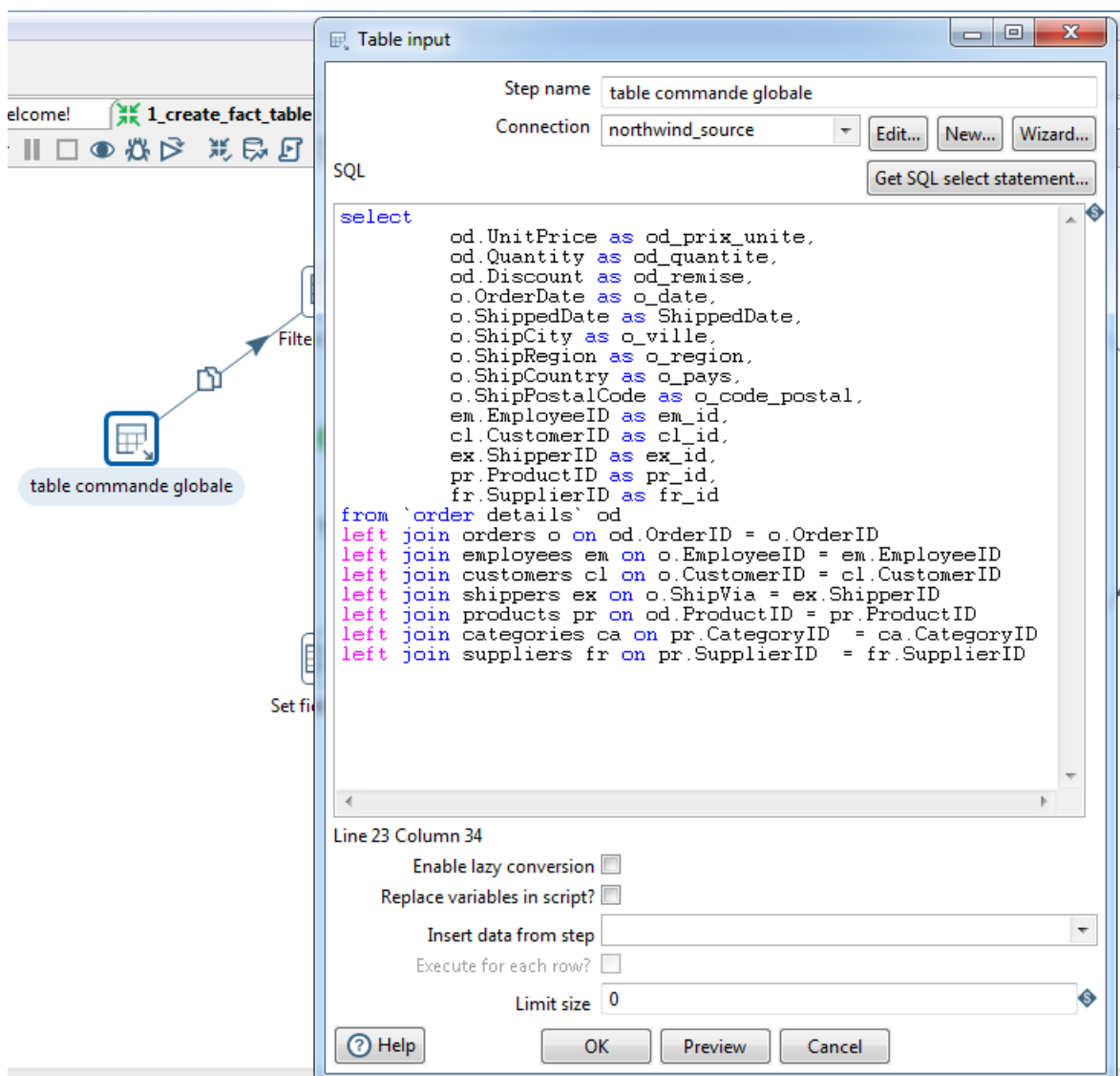


Figure 30 ETL extraction des données depuis la base source

## Etape 2 : transformation 1

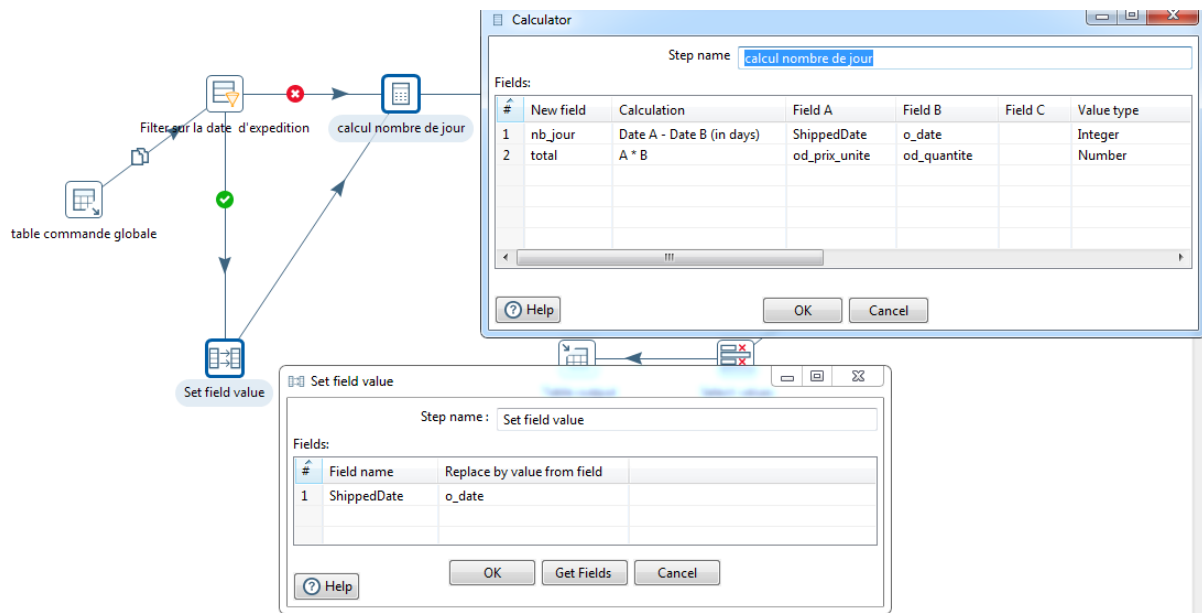


Figure 31 ETL création des variables de la table de faits

Description : calcul de la mesure « nb-jour » et la mesure « total ». La première s'obtient par la soustraction de deux colonnes date d'expédition et la date de la commande et la deuxième s'obtient par la multiplication du prix unitaire par la quantité, mais on a trouvé une difficulté, c'est que la colonne 'ShippedDate' contient des valeurs nulles cela est résolu par les remplacer par les valeurs de la colonnes date du table commande, et c'est le rôle de {filter de la date d'expédition} de détecter ces valeurs nulles.

## Etape 3 : transformation 2

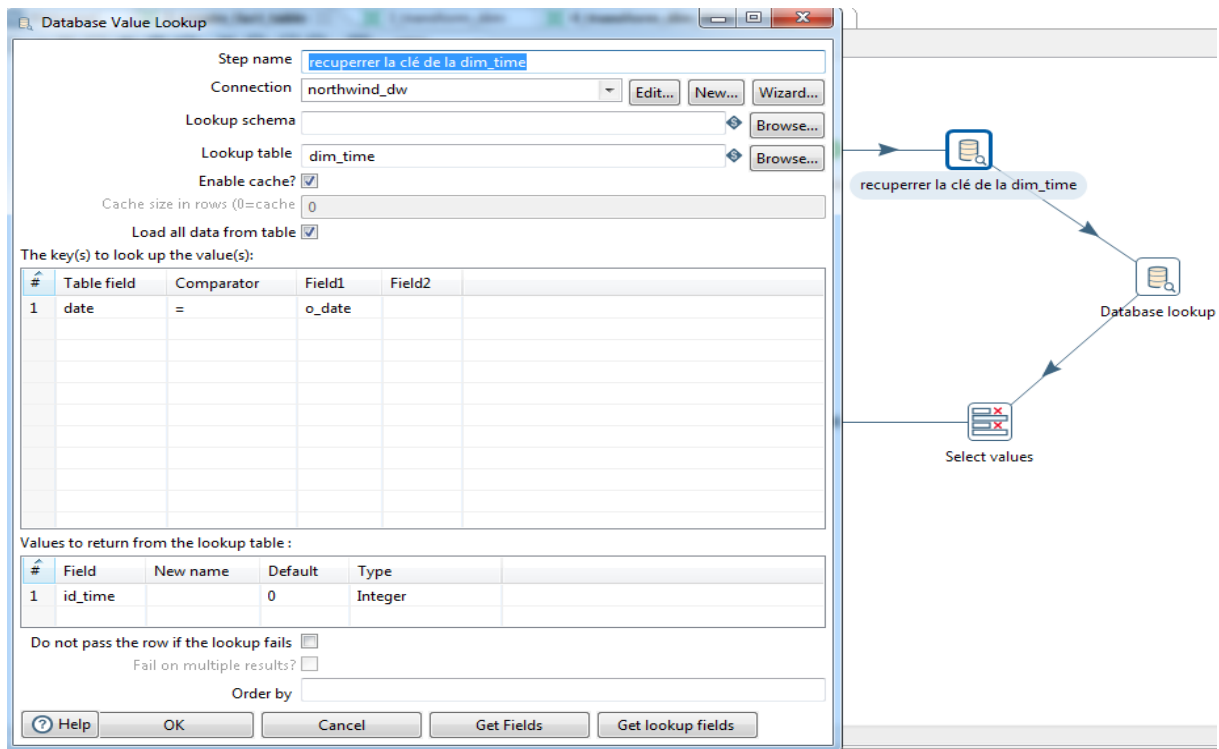


Figure 32 ETL récupération de la clé de dim\_time

Description : puisque la dimension temps est créée sur PDI alors la liaison entre cette dimension et la table de faits n'est pas encore faite. Dans cette étape on récupère la clé de la dimension de temps en pointant sur la colonne date de la table commande, celle qui appartient à la base source.

## Etape 4 : transformation 3

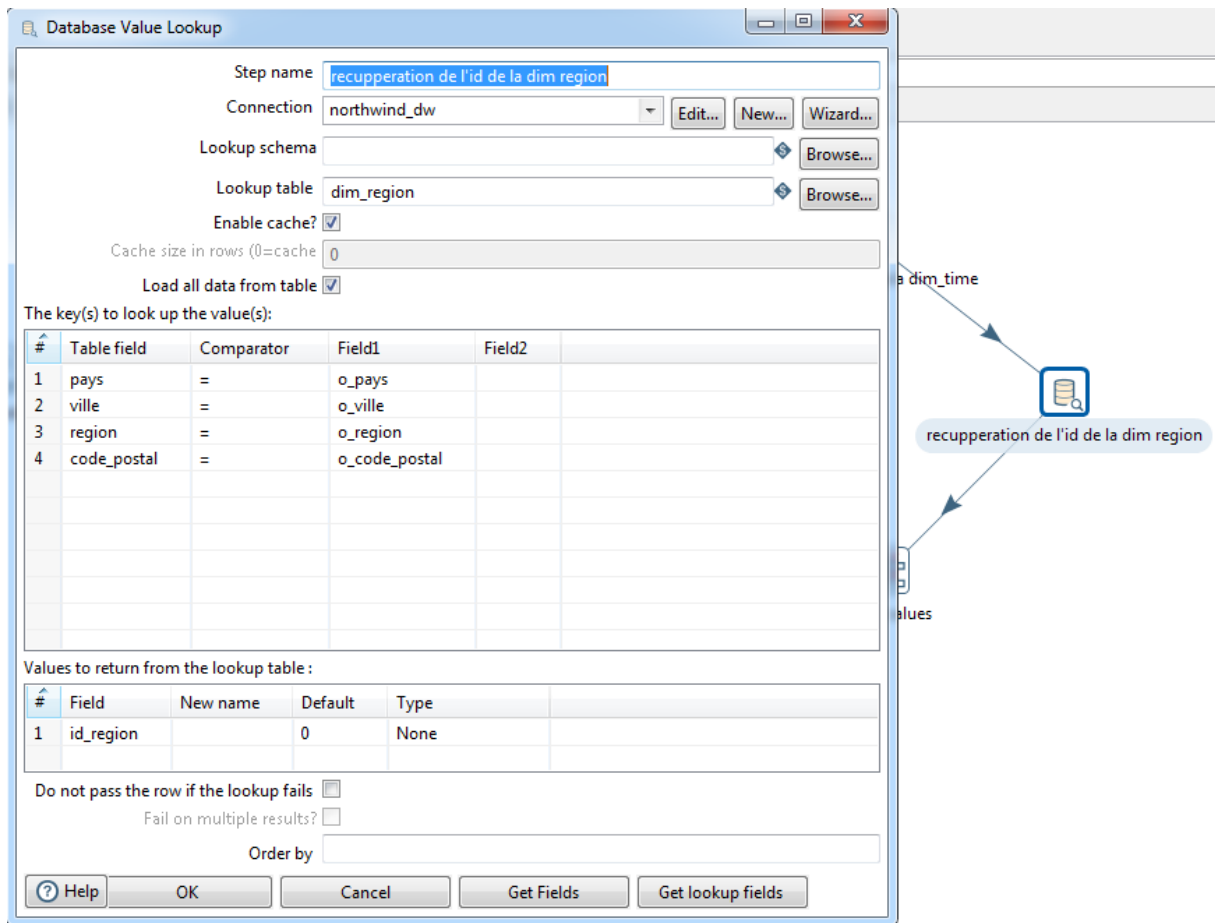


Figure 33 ETL récupération de la clé de dim\_region

Description : on récupère cette fois la clé de la dimension région et on attache cette dernière avec les données récupérées par la requête.



## Etape 5 : chargement

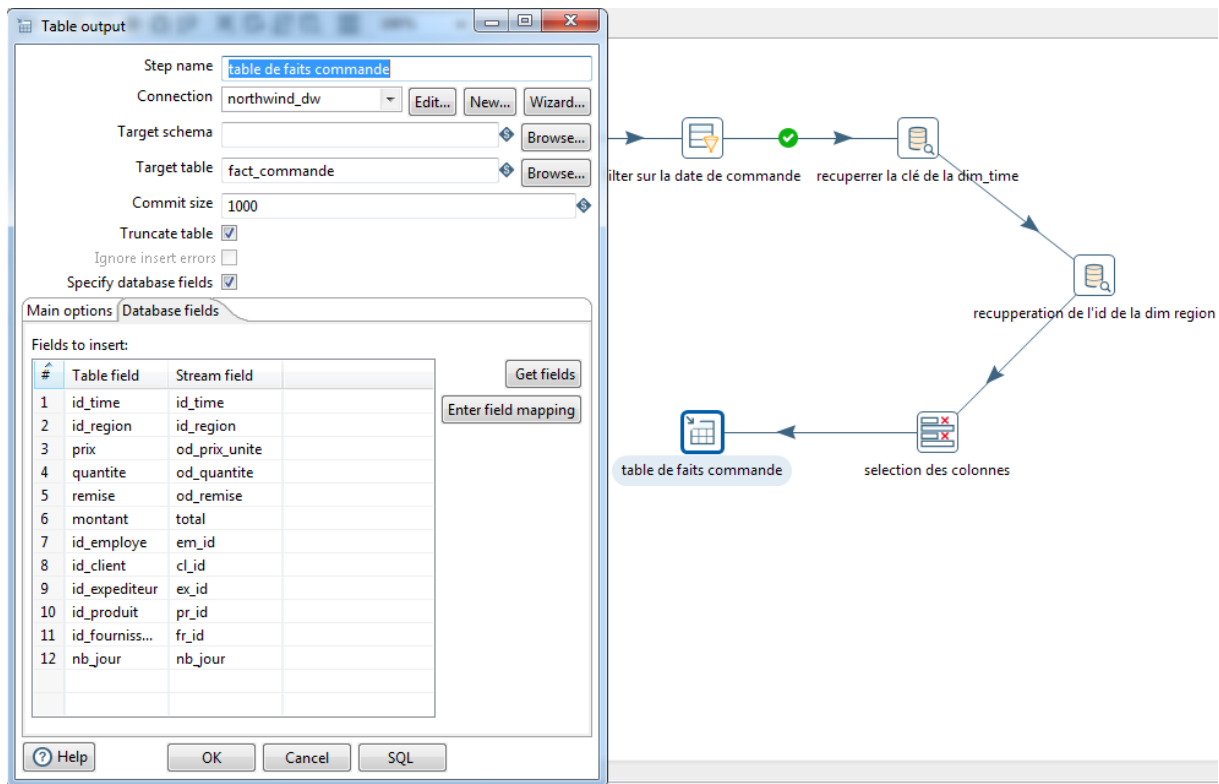


Figure 34 ETL chargement des données dans la table de faits

Description : en fin c'est le mapping des colonnes de la base des données.

# Chapitre 4

## Mondrian Schéma Design

## INTRODUCTION :

En fonction des besoins de reporting identifiés, on a défini les cubes, hiérarchies, dimensions et rôles dans un schéma Mondrian.

Le schéma Mondrian doit être représenté sous forme xml, ci-joint vous trouverez un fichier « Schema\_cube\_commandes.xml » qui contient le schéma écrit en xml.

le schéma représente la structure hiérarchique de notre datamart, il est composé d'un seul cube «commandes» qui est une collection de mesures et dimensions ayant une chose commune est la table de fait 'fact\_commande'

```
<Schema name="Schema_CMD_1">
  <Cube name="Commandes" visible="true" cache="true" enabled="true">
    <Table name="fact_commande">
    </Table>
    ...
  </Cube>
</Schema>
```

Figure 35 schéma Mondrian création de la table de faits

La table de faits est définie par l'élément <Table>.

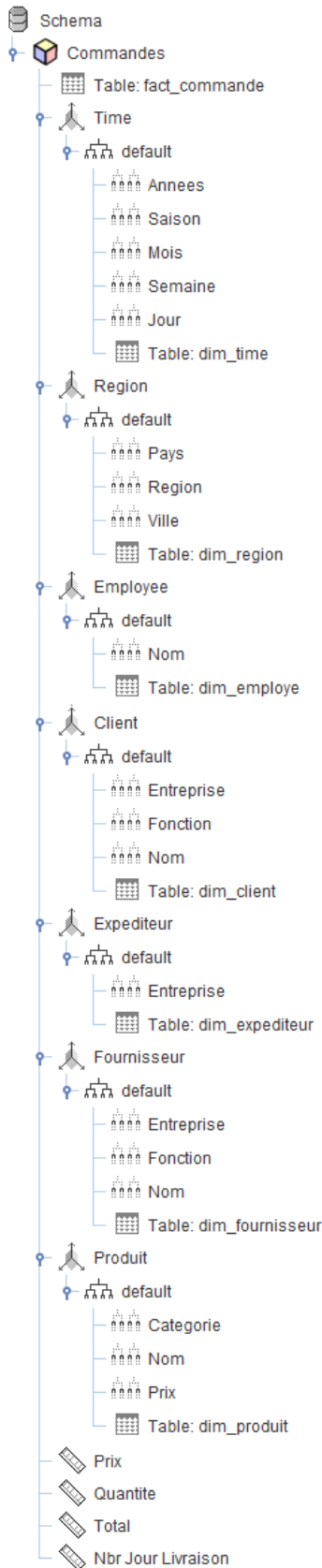


Figure 36 schéma Mondrian du data warehouse

## 4.1 Les dimensions :

**Time** : est défini par l'élément `<dimension>`, visible, contenant un attribut qui définit la clé étrangère et de nom Time

```
<Dimension type="TimeDimension" visible="true" foreignKey="id_time"
  highCardinality="false" name="Time">
  <Hierarchy visible="true" hasAll="true" primaryKey="id_time">
    <Table name="dim_time">
      </Table>
    ...
  </Hierarchy>
</Dimension>
```

Figure 37 création de la table time en XML

Cette dimension contient une hiérarchie qui définit celle de la table dim\_time, comme c'est montré ci-dessous dim\_time contient 5 niveaux.

```
<Dimension type="TimeDimension" visible="true" foreignKey="id_time" highCardinality="false" name="Time">
  <Hierarchy visible="true" hasAll="true" primaryKey="id_time">
    ...
    <Level name="Annees" visible="true" column="year" type="String" uniqueMembers="true"
      levelType="TimeYears" hideMemberIf="Never">
    </Level>

    <Level name="Saison" visible="true" column="saison" type="String" uniqueMembers="false"
      levelType="TimeQuarters" hideMemberIf="Never" captionColumn="saison_label">
    </Level>

    <Level name="Mois" visible="true" column="mois" type="String" uniqueMembers="false"
      levelType="TimeMonths" hideMemberIf="Never" captionColumn="mois_label">
    </Level>

    <Level name="Semaine" visible="true" column="semaine" type="String" uniqueMembers="false"
      levelType="TimeWeeks" hideMemberIf="Never">
    </Level>

    <Level name="Jour" visible="true" column="jour" type="String" uniqueMembers="false"
      levelType="TimeDays" hideMemberIf="Never" captionColumn="jour_label">
    </Level>
  </Hierarchy>
</Dimension>
```

Figure 38 les niveaux de la table Temps en XML

années → saison → mois → semaine → jour

**Région** : dimension qui contient une hiérarchie de 3 niveaux comme c'est mentionné dans le schéma à gauche et structuré en bas.

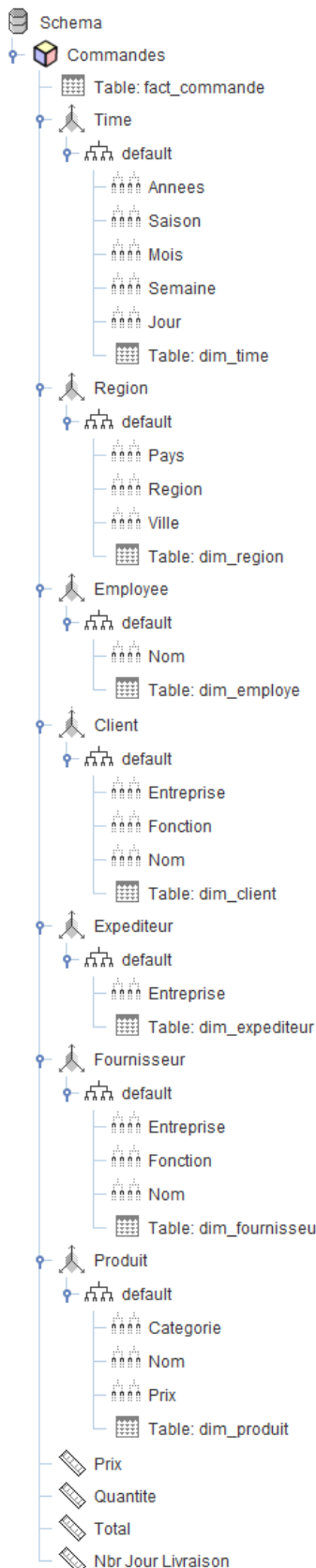
```
<Dimension type="StandardDimension" visible="true" foreignKey="id_region"
  highCardinality="false" name="Region">
  <Hierarchy visible="true" hasAll="true" primaryKey="id_region">
    <Table name="dim_region">
      </Table>

    <Level name="Pays" visible="true" column="pays" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>

    <Level name="Region" visible="true" column="region" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>

    <Level name="Ville" visible="true" column="ville" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

Figure 39 code XML de la création de la dimension Région



**Employé:** cette dimension est différente des autres, elle contient une hiérarchie parent-enfant, d'où on a un seul niveau.

```
<Dimension type="StandardDimension" visible="true"
  foreignKey="id_employe" highCardinality="false"
  name="Employee">
  <Hierarchy visible="true" hasAll="true"
    primaryKey="id_employe">
    <Table name="dim_employe">
    </Table>
    <Level name="Nom" visible="true" column="id_employe"
      nameColumn="nom_employe" parentColumn="superviseur"
      type="String" uniqueMembers="true"
      levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

Figure 40 code XML de la création de la dimension Employé

La ligne marquée en rouge qui nous intéresse, les deux attributs nameColumn et parentColumn donnent une particularité à la dimension Employé par rapport aux autres dimensions, parentColumn dans ce cas, c'est la colonne de clé étrangère qui pointe vers le superviseur d'un employé.

**Client :** contient une hiérarchie telle que

```
<Dimension type="StandardDimension" visible="true" foreignKey="id_client"
  highCardinality="false" name="Client">
  <Hierarchy visible="true" hasAll="true" primaryKey="id_client">
    <Table name="dim_client">
    </Table>
    <Level name="Entreprise" visible="true" column="nom_entreprise"
      type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    <Level name="Fonction" visible="true" column="fonction_client" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    <Level name="Nom" visible="true" column="nom_client" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

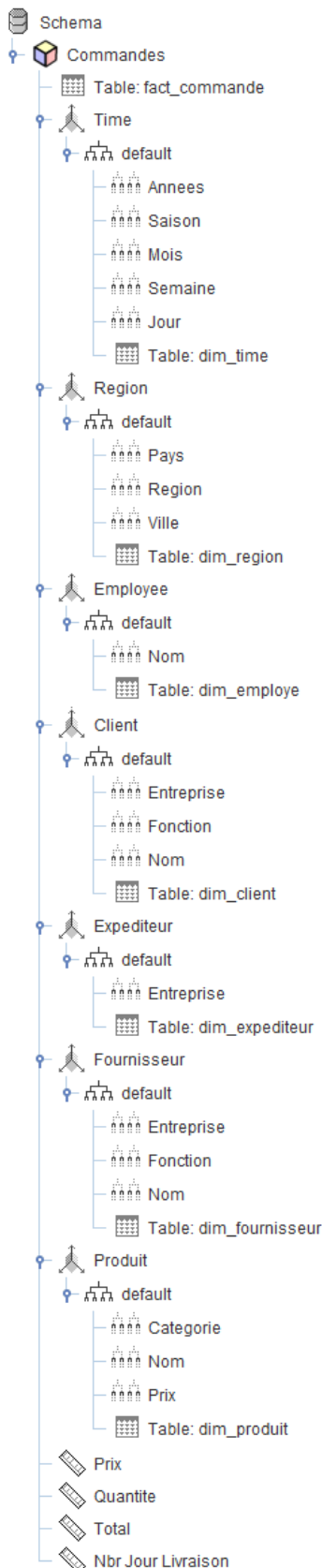
Figure 41 code XML de la création de la dimension Client

**Expéditeur :** Un seul niveau : entreprise

```
<Dimension type="StandardDimension" visible="true" foreignKey="id_expéditeur"
  highCardinality="false" name="Expéditeur">
  <Hierarchy visible="true" hasAll="true" primaryKey="id_expéditeur">
    <Table name="dim_expéditeur">
    </Table>
    <Level name="Entreprise" visible="true" column="nom_entreprise" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

Figure 42 code XML de la création de la dimension Expéditeur

3 niveaux, entreprise → fonction → nom



## Fournisseur : une hiérarchie de trois niveaux

```
<Dimension type="StandardDimension" visible="true" foreignKey="id_fournisseur"
highCardinality="false" name="Fournisseur">

  <Hierarchy visible="true" hasAll="true" primaryKey="id_fournisseur">
    <Table name="dim_fournisseur">
      </Table>

    <Level name="Entreprise" visible="true" column="nom_entreprise_fournisseur"
      type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>

    <Level name="Fonction" visible="true" column="fonction_fournisseur"
      type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>

    <Level name="Nom" visible="true" column="nom_fournisseur" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
    </Hierarchy>
  </Dimension>
```

Figure 43 code XML de la création de la dimension Fournisseur

Entreprise → fonction → nom

## Produit : elle aussi contient trois niveaux

```
<Dimension type="StandardDimension" visible="true" foreignKey="id_produit"
highCardinality="false" name="Produit">

  <Hierarchy visible="true" hasAll="true" primaryKey="id_produit">
    <Table name="dim_produit">
      </Table>

    <Level name="Categorie" visible="true" column="nom_categorie"
      type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>

    <Level name="Nom" visible="true" column="nom_produit" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>

    <Level name="Prix" visible="true" column="prix_produit" type="String"
      uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>
    </Hierarchy>
  </Dimension>
```

Figure 44 la création de la dimension Produit dans le schéma Mondrian

Catégorie → nom → prix

## 4.2 Les mesures :

**Prix** : cette mesure est déclarée par l'attributs 'measure' et est de type « somme » .

```
<Measure name="Prix" column="prix" aggregator="sum" visible="true">
</Measure>
```

Figure 45 la mesure Prix en XML

**Quantité** : calcule la somme des quantités de produits commandés.

```
<Measure name="Quantite" column="quantite" aggregator="sum" visible="true">
</Measure>
```

Figure 46 la mesure Quantité en XML

**Total** : contient le montant total de la commande.

```
<Measure name="Total" column="montant" aggregator="sum" visible="true">  
</Measure>
```

Figure 47 la mesure total en XML

**Nombre jour livraison** : le nombre moyen des jours de livraison.

```
<Measure name="Nbr Jour Livraison" column="nb_jour" aggregator="avg" visible="true">  
</Measure>
```

Figure 48 la mesure nombre de jr de livraison en XML

# Chapitre 5

## MDX Querying



## La requête 1 :

+ Quels sont les produits les plus commandés dans une région à une saison ?

❖ SELECT {[Measures].[Quantite]} ON COLUMNS,  
TopCount({[Produit].[Nom].Members}, 5.0, [Measures].[Quantite]) ON ROWS  
FROM [Commandes]  
WHERE TopCount(CrossJoin({[Region].[USA]}, {[Time].[1997].[4]}), 5.0,  
[Measures].[Quantite])

	Measures
Produit	Quantite
+ Rogede sild	150
+ Perth Pasties	120
+ Tourtire	100
+ Scottish Longbreads	91
+ Rhnbru Klosterbier	60

Figure 49 la requête décisionnelle 1

## La requête 2 :

+ Quelle est la durée moyenne d'expédition par pays ?

❖ SELECT {[Measures].[Nbr Jour Livraison]} ON COLUMNS,  
Order({[Region].[Pays].Members}, [Measures].[Nbr Jour Livraison], DESC) ON  
ROWS  
FROM [Commandes]

	Measures
Region	Nbr Jour Livraison
+ Ireland	12,236
+ Sweden	10,155
+ Belgium	9,732
+ Switzerland	8,981
+ Poland	8,812
+ Argentina	8,588
+ UK	8,437
+ France	8,19
+ Germany	8,107
+ USA	8,088
+ Portugal	8,067
+ Spain	7,796

Figure 50 la requête décisionnelle 2

**La requête 3 :**

+ Quels sont les produits les plus commandés par un client ?

❖ SELECT {[Measures].[Quantite]} ON COLUMNS,  
TopCount({[Produit].[Nom].Members}, 10.0, [Measures].[Quantite]) ON  
ROWS  
FROM [Commandes]  
WHERE [Client].[Familia Arquibaldo].[Marketing Assistant].[Aria Cruz]

	Measures
Produit	Quantite
+ Rhnbru Klosterbier	56
+ Chartreuse verte	50
+ Geitost	50
+ Tourtire	30
+ Nord-Ost Matjeshering	30
+ Guaran Fantstica	25
+ Perth Pasties	25
+ Teatime Chocolate Biscuits	18
+ Camembert Pierrot	12
+ Jack's New England Clam Chowder	12

Figure 51 la requête décisionnelle 3

**La requête 4 :**

+ Quel est le client actifs d'une année ?

❖ SELECT {[Measures].[Total]} ON COLUMNS,  
TopCount({[Client].[Nom].Members}, 1, [Measures].[Total]) ON ROWS  
FROM [Commandes]  
WHERE [Time].[1997]

	Measures
Client	Total
Horst Kloss	64 238

Figure 52 la requête décisionnelle 4

**La requête 5 :**

+ Quelle est la durée moyenne d'expédition par chaque expéditeur ?

❖ SELECT  
 {[Measures].[Nbr Jour Livraison]} ON COLUMNS,  
 Order({[Expéditeur].[Entreprise].Members}, [Measures].[Nbr Jour Livraison],  
 ASC) ON ROWS  
 FROM [Commandes]

	Measures
Expéditeur	Nbr Jour Livraison
Federal Shipping	7,415
Speedy Express	8,187
United Package	8,453

Figure 53 la requête décisionnelle 5

**La requête 6 :**

+ Quels sont les montants des commandes traitées par employé et/ou sous sa responsabilité ?

❖ SELECT {[Measures].[Total]} ON COLUMNS,  
 Hierarchize(Union({[Employee].[Andrew Fuller]}, [Employee].[Andrew  
 Fuller].Children)) ON ROWS  
 FROM [Commandes]

	Measures
Employee	Total
Andrew Fuller	1 354 458,59
Nancy Davolio	202 143,71
Janet Leverling	213 051,3
Margaret Peacock	250 187,45
Steven Buchanan	378 025,84
Laura Callahan	133 301,03

Figure 54 la requête décisionnelle 6

## La requête 7 :

- ✚ Quels sont les clients ayant commander le plus ?
- ❖ WITH MEMBER [Measures].[RAP] AS '([([Measures].[Total])) / [Measures].[Quantite])' MEMBER [Measures].[CLASS] AS 'IIf([Measures].[RAP] > 20.0), "GOOD\_CLIENT", "BAD\_CLIENT")' SELECT {[Measures].[Quantite], [Measures].[Total], [Measures].[CLASS]} ON COLUMNS, {[Client].[Nom].Members} ON ROWS FROM [Commandes] WHERE [Time].[1997]

Client	Measures		
	Quantite	Total	CLASS
Maria Anders	79	2 294	GOOD_CLIENT
Ana Trujillo	28	799,75	GOOD_CLIENT
Antonio Moreno	295	6 452,15	GOOD_CLIENT
Thomas Hardy	371	6 589	BAD_CLIENT
Victoria Ashworth	165	3 179,5	BAD_CLIENT
Christina Berglund	481	14 533,2	GOOD_CLIENT
Hanna Moos	68	1 079,8	BAD_CLIENT
Martn Sommer	60	4 035,8	GOOD_CLIENT
Frdrique Citeaux	348	8 371,8	GOOD_CLIENT

Figure 55 la requête décisionnelle 7

# Chapitre 6

## DATA Mining

### INTRODUCTION :

De nos jours, le monde de l'entreprise est caractérisé par une concurrence de plus en plus accrue. Cette accélération de la concurrence oblige les entreprises à renforcer leur marketing à travers une analyse détaillée des besoins des clients pour aboutir à de nouvelles connaissances. Ce qui montre la conservation des masses de données importantes sur les profils et les achats des clients. D'où l'utilisation du terme « **Datamining** ».

Le **datamining** est l'analyse d'un ensemble d'observations qui a pour but de trouver des relations insoupçonnées et résumer les données d'une nouvelle manière, de façon qu'elles soient plus compréhensibles et utiles pour leurs détenteurs

L'étude de Datamining que nous réalisons porte sur les données issues des ventes des produits dans l'entreprise Northwond. Cette étude est faite à l'aide du Weka.

Weka : « environnement Waikato pour l'analyse de connaissances » est une suite de logiciels d'apprentissage automatique. Écrite en Java, développée à l'université de Waikato en Nouvelle-Zélande.

Note model va nous permettre de classer les client en bon client et en client a faible potentielle en connaissant son pays d'origine et le nombre et le total de ses commandes.

## Extraction des données à partir de notre Datamart.

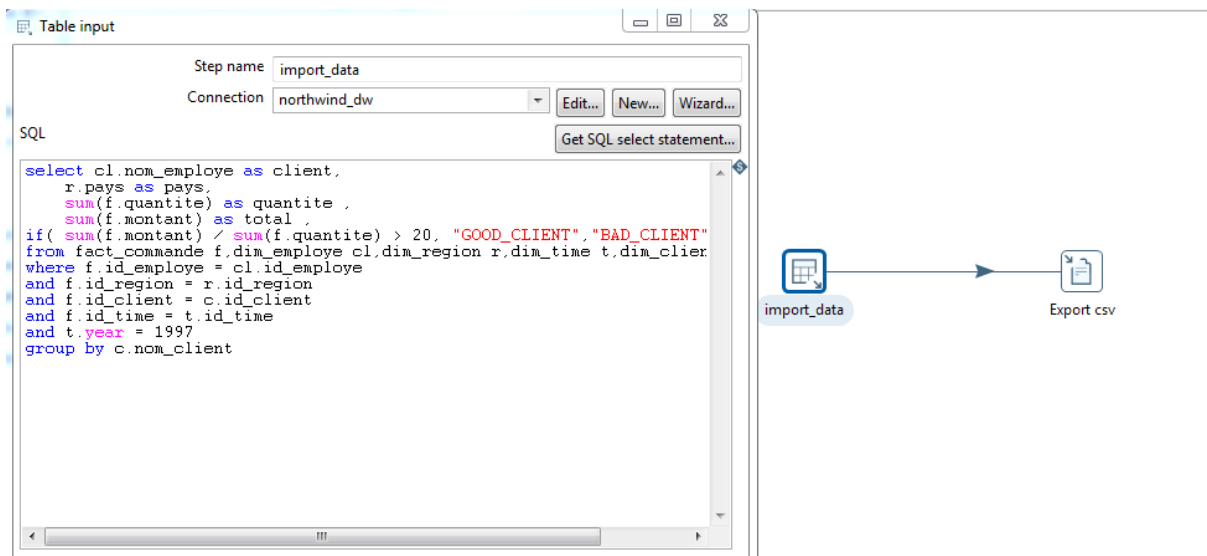


Figure 56 data mining extraction des données

- Création d'un fichier csv à partir du datamart.
- Pour déterminer la classe de notre model on a utilisé l'expression (rapport du montant et la quantité soit plus de 20 ).
- Ici on récupère les données d'une seule années (1997) afin d'utiliser les données d'une autre pour le test de notre model.

## Transfert des données du fichier csv en arff.

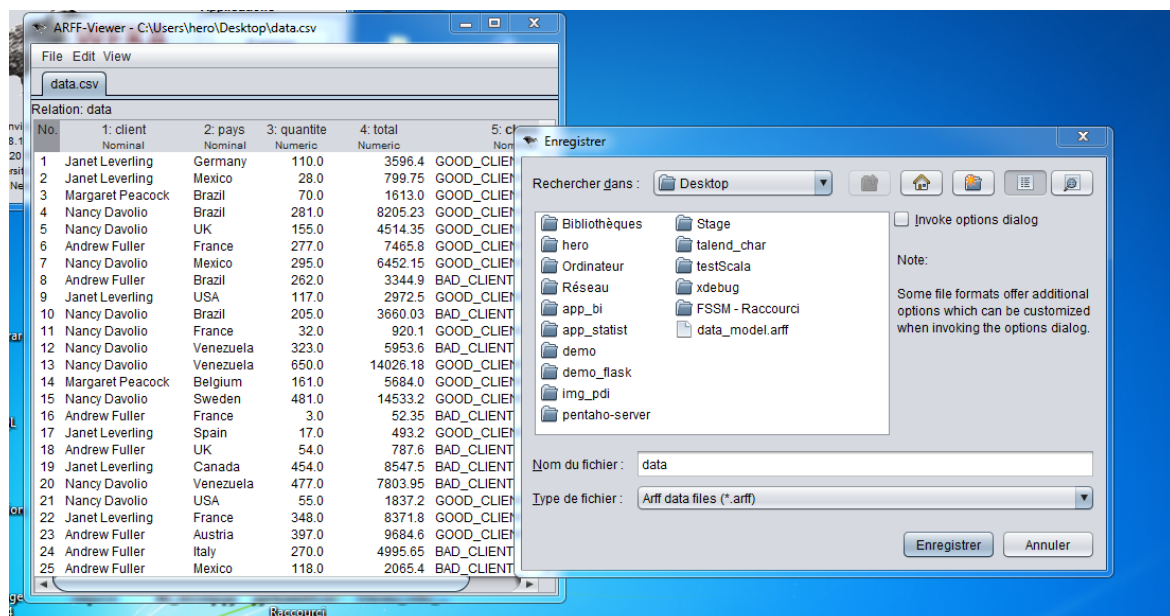


Figure 57 data mining transfert des données en arff

On a utilisé le Viewer de Weka pour transformer le fichier csv en arff

## le chargement des données dans weka

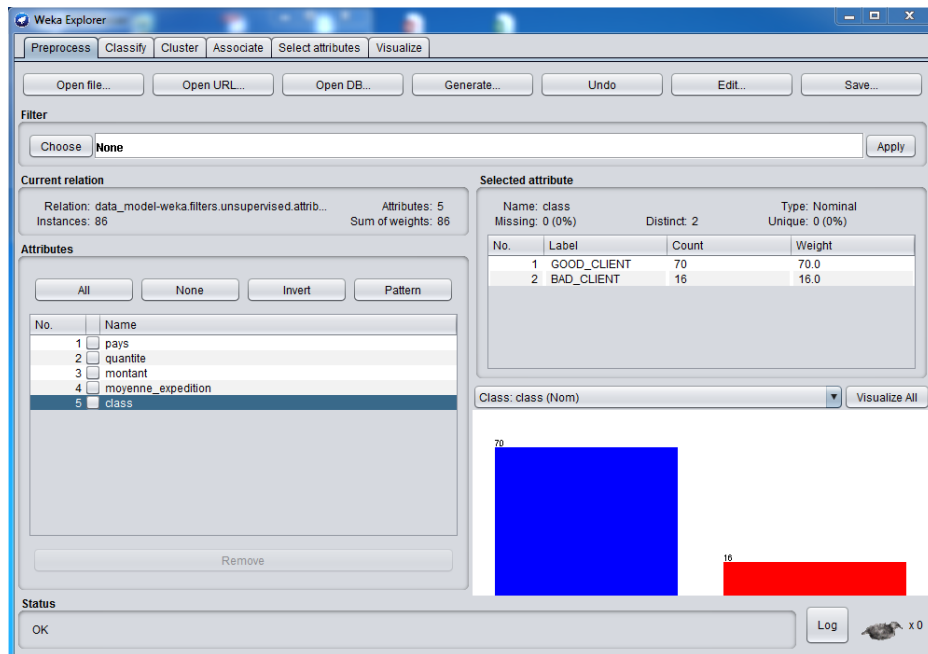


Figure 58 data mining chargement des données dans weka

## Création des modèles en utilisant l'arbre de décision .

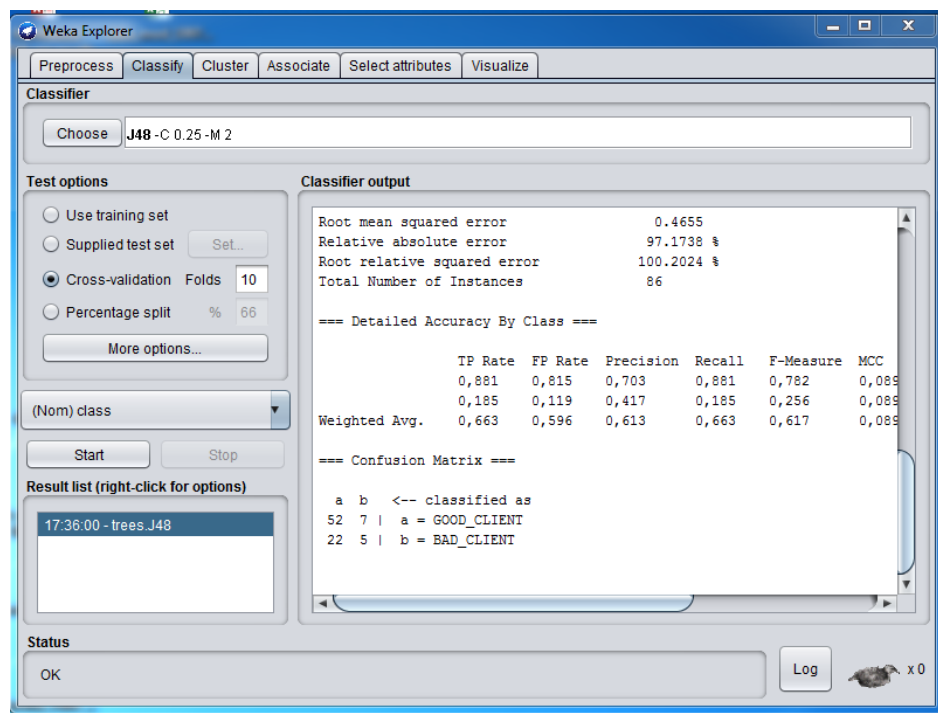


Figure 59 data mining phase de classification

Utilisation de l'algorithme **j48** (arbre de décision).



L'arbre de décision J48 est la mise en œuvre de l'algorithme ID3 développé par l'équipe du projet WEKA. R inclut ce beau travail dans le paquet RWeka.

On constate à partir de matrice de confusion de ce model est capable de détecter les bon client , par contre il a de mal à détecter les client à faible potentiel Ce nous permet avoir un taux d'erreur 33%

**Création du model en utilisant la régression logistique.**

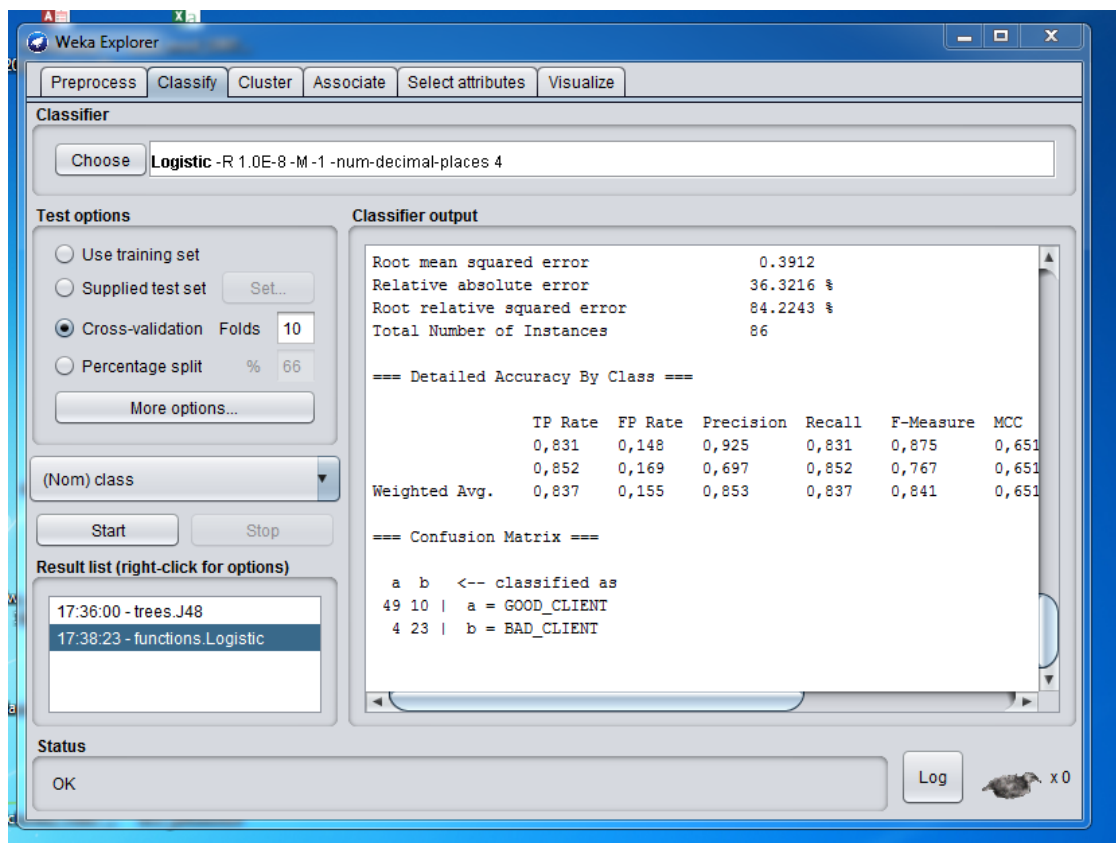


Figure 60data mining phase de classification 2

Utilisation de la régression logistique nous permet d'avoir un taux d'erreur de 16% Moins que le model de l'arbre de décision

## Utilisation du model sur des nouvelles valeurs :



Figure 61 data mining PDI

Ci-dessus les étape d'obtention du fichier csv final.

Pour plus de détails :

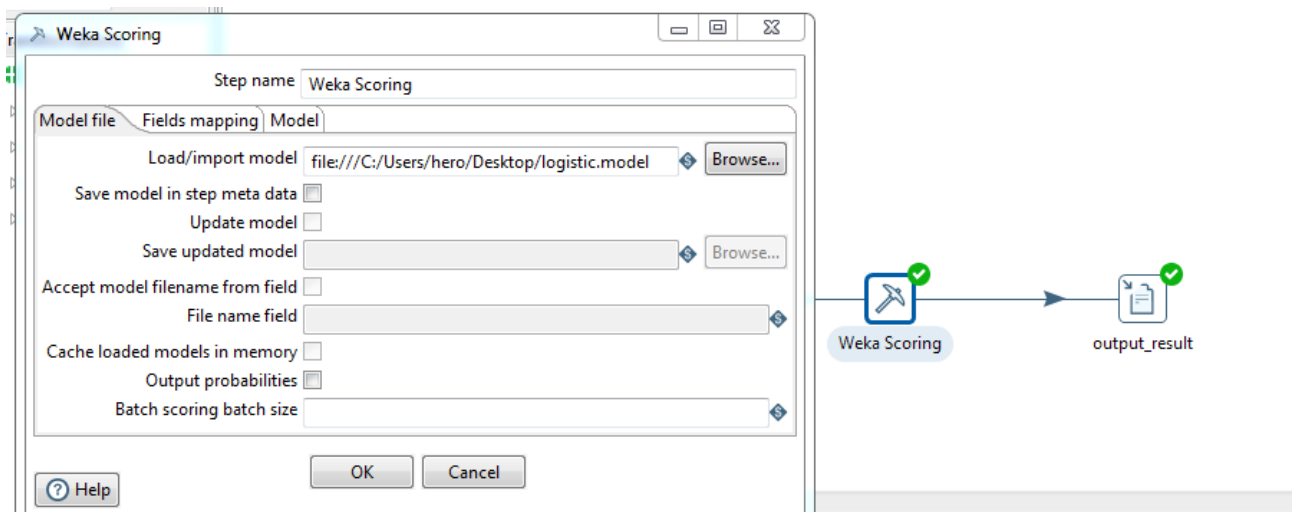


Figure 62 data mining PDI en détails

Ajout du model généré à partir de weka

# Chapitre 7

## Tableau de bord

# Tableau de bord

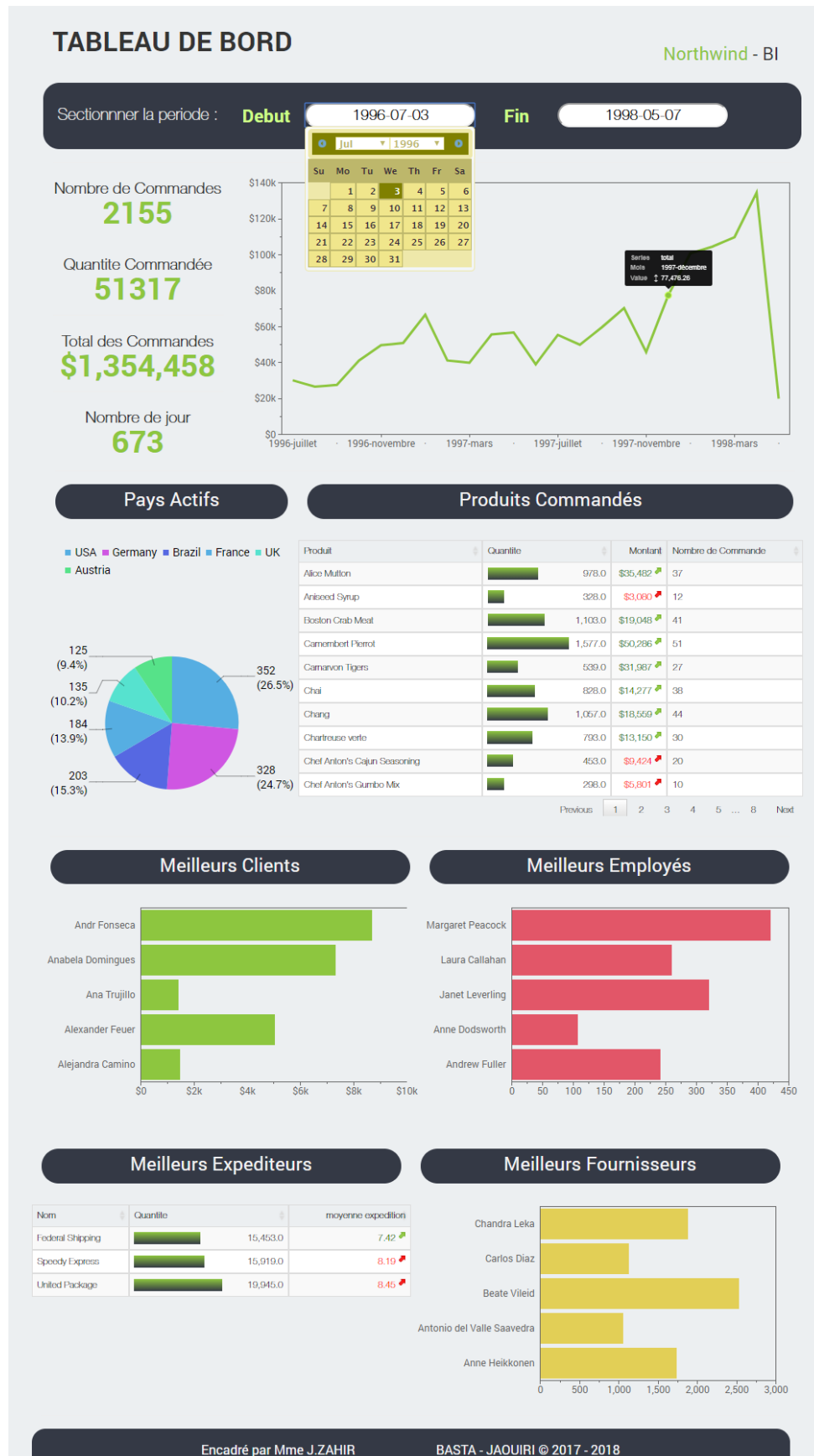


Figure 63 le tableau de bord de Northwind

Rien n'est mieux que synthétiser notre analyse des données de Northwind par un instrument de mesure de la performance qui facilite le pilotage d'une ou plusieurs activités dans le cadre d'une démarche de progrès. Il contribue à réduire l'incertitude et facilite la prise de risque inhérente à toutes décisions. C'est un instrument d'aide à la décision, le tableau de bord aide les décideurs de l'entreprise de visualiser les différentes statistiques, les comparaisons dans le temps des ventes et le suivi des client et le personnel.

Nous commençons par expliquer la première partie du Dashboard.



Figure 64 tableau de bord de Northwind partie 1

Dans cette partie on choisit la période des ventes qu'on veut apparaitre, on a comme résultat les valeurs des mesures de notre cube et un diagramme qui compare les montant des commandes pendant la période demandée.

Le tableau de bord contient aussi :

- Un diagramme circulaire : les six premiers pays ayant le effectuer les commande dans cette période.
- Un tableau statistique : les statistiques de tous les produits par rapport à leurs ventes dans cette période.

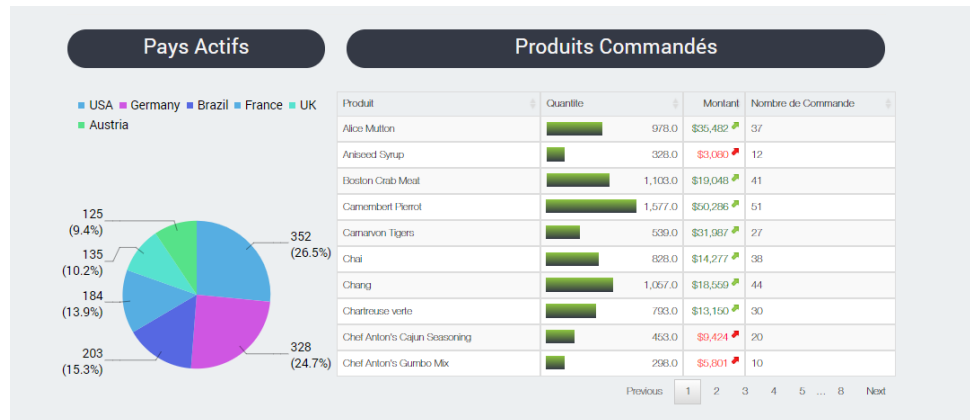


Figure 65 tableau de bord partie 2

- Deux histogrammes, un pour classier les client et l'autre pour les employés.

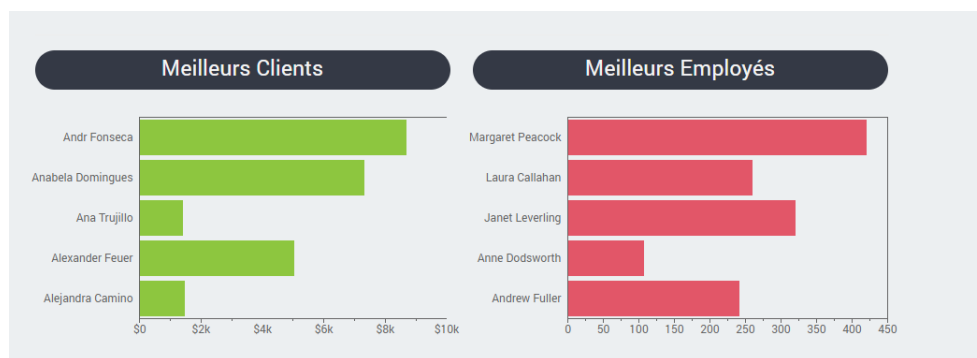


Figure 66 tableau de bord partie 3

- Un tableau statistique pour classier les expéditeurs.
- Un histogramme qui affiche les valeurs des meilleurs fournisseurs.

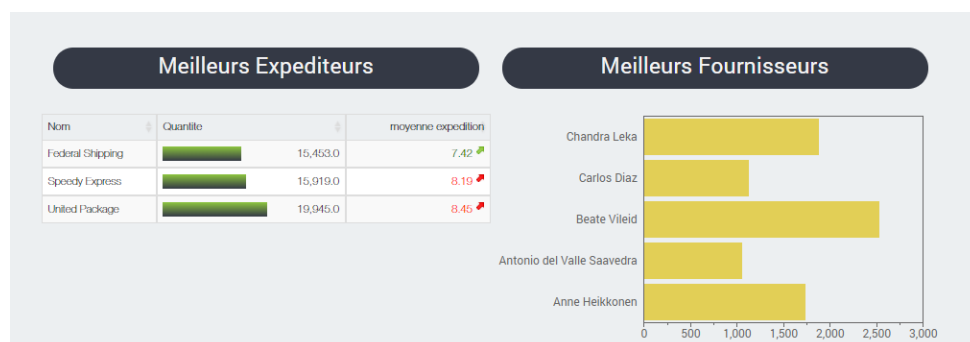


Figure 67 tableau de bord partie 4

- Le cours de Mme J. Zahir en business intelligence.
- <https://northwinddatabase.codeplex.com/>.
- <https://mondrian.pentaho.com/documentation/schema.php>.
- <https://www.webucator.com/tutorial/learn-sql/simple-selects/introduction-the-northwind-database-reading.cfm>.
- [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwiM9p\\_D173YAhUGOhQKHQ78D1IQFggguMAI&url=https%3A%2F%2Fwww.piloter.org%2Fmesurer%2Ftableau\\_de\\_bord%2Fprincipe-tableau-de-bord.htm&usg=AOvVaw3X1yJDpP-ZcdZ6HSosbSqH](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwiM9p_D173YAhUGOhQKHQ78D1IQFggguMAI&url=https%3A%2F%2Fwww.piloter.org%2Fmesurer%2Ftableau_de_bord%2Fprincipe-tableau-de-bord.htm&usg=AOvVaw3X1yJDpP-ZcdZ6HSosbSqH).