



RAPPORT DU PROJET BI

année académique : 2019-2020

Réalisé par :

- MAAZOU Mohammed
- RIDOUANI AYOUB

Encadré par :

RIFFI Jamal

I. Table des matières

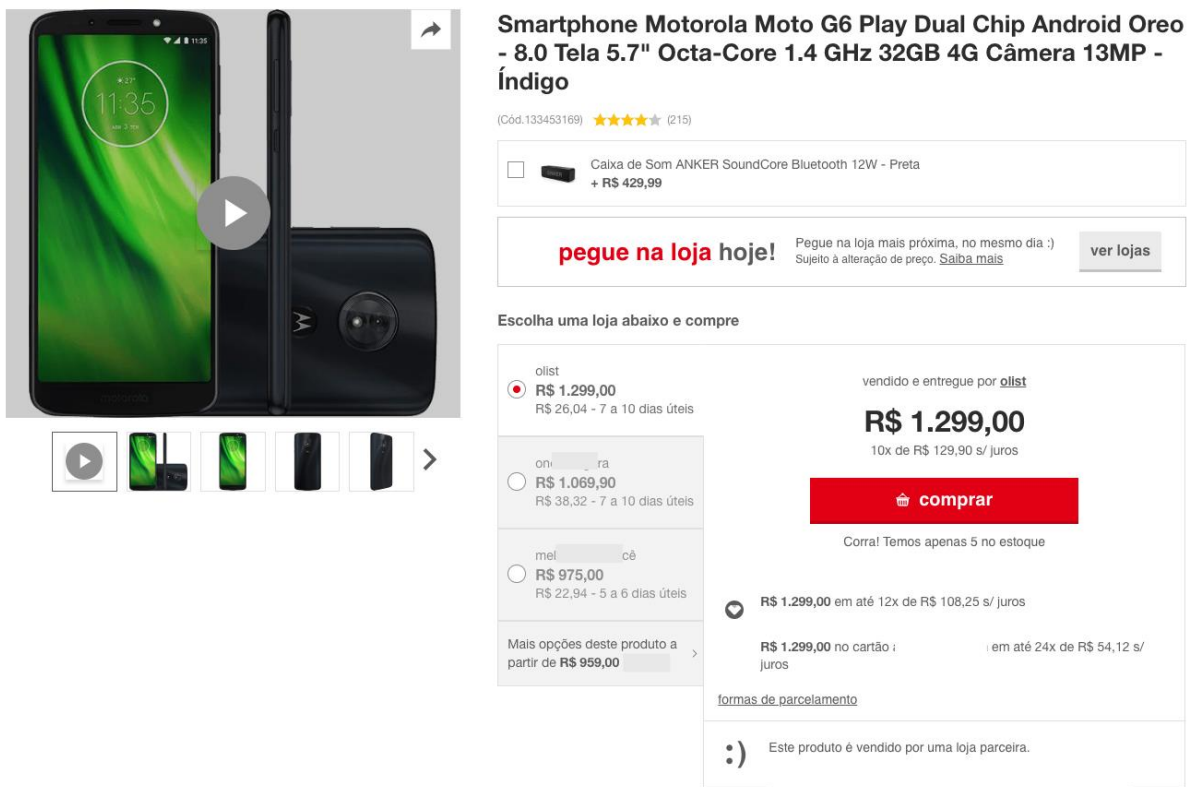
II.	Problématique :	2
A.	La description des données:	2
•	Exemple de liste de produits sur une place de marché	2
B.	Le contexte :	2
III.	Modélisation dimensionnelle	3
A.	Étude des besoins utilisateurs :	3
1.	Étude intégrée des sources de données	3
B.	Modélisation du data warehouse :	4
IV.	ETL	6
A.	Définition	6
B.	Talend open studio pour l'intégration de données	7
V.	OLAP (online analytical processing)	14
A.	Définition	14
B.	OLAP, comment ça marche ?	15
C.	Modélisation et publication de cube OLAP	15
VI.	Reporting	20
VII.	Data Mining	23

II. Problématique :

A. La description des données:

Il s'agit d'un ensemble de données publiques brésiliennes sur le commerce électronique des commandes passées sur "Olist Store". L'ensemble de données contient des informations sur 100 000 commandes de 2017 à 2019 effectuées sur plusieurs marchés au Brésil. Ses fonctionnalités permettent d'afficher une commande à partir de plusieurs dimensions: de l'état de la commande, et du prix, aux attributs du produit et enfin aux avis rédigés par les clients. et également un ensemble de données de géolocalisation.

- Exemple de liste de produits sur une place de marché



Smartphone Motorola Moto G6 Play Dual Chip Android Oreo - 8.0 Tela 5.7" Octa-Core 1.4 GHz 32GB 4G Câmera 13MP - Índigo

(Cód.133453169) ★★★★★ (215)

☐ Caixa de Som ANKER SoundCore Bluetooth 12W - Preta + R\$ 429,99

pegue na loja hoje! Pegue na loja mais próxima, no mesmo dia :) Sujeito à alteração de preço. [Saiba mais](#) [ver lojas](#)

Escolha uma loja abaixo e compre

loja	preço	prazo
olist	R\$ 1.299,00	R\$ 26,04 - 7 a 10 dias úteis
onli	R\$ 1.069,90	R\$ 38,32 - 7 a 10 dias úteis
mel	R\$ 975,00	R\$ 22,94 - 5 a 6 dias úteis

Mais opções deste produto a partir de R\$ 959,00

vendido e entregue por **olist**

R\$ 1.299,00

10x de R\$ 129,90 s/ juros

comprar

Corra! Temos apenas 5 no estoque

R\$ 1.299,00 em até 12x de R\$ 108,25 s/ juros

R\$ 1.299,00 no cartão : em até 24x de R\$ 54,12 s/ juros

[formas de parcelamento](#)

:) Este produto é vendido por uma loja parceira.

B. Le contexte :

Cet ensemble de données a été généreusement fourni par Olist, le plus grand magasin des marchés brésiliens. Olist connecte les petites entreprises de tout le Brésil à des canaux sans tracas et avec un seul contrat. Ces marchands peuvent vendre leurs produits via la boutique Olist et les expédier directement aux clients en utilisant les partenaires logistiques Olist.

Après qu'un client a acheté le produit sur Olist Store, un vendeur est notifié pour exécuter cette commande. Une fois que le client reçoit le produit ou que la date de livraison estimée est due, le client reçoit une enquête de satisfaction par e-mail où il peut donner une note pour l'expérience d'achat et écrire quelques commentaires.

III. Modélisation dimensionnelle

A. Étude des besoins utilisateurs :

La direction cherche à améliorer le nombre de ventes, et elle cherche à savoir si les employés ont une influence sur les volumes de ventes et si cela varie en fonction des jours de la semaine ou de certaines périodes de l'année. Elle voudrait également savoir dans quel endroit le nombre de vente est élevé, Elle se demande aussi si certains produits se vendent mieux à certaines dates et/ou dans certains endroits.

Requêtes organisées par dimension :

Quantité :

/ Produit (prix,avis, weight_glength_cm,height_cm ,width_cm)

/ Employé

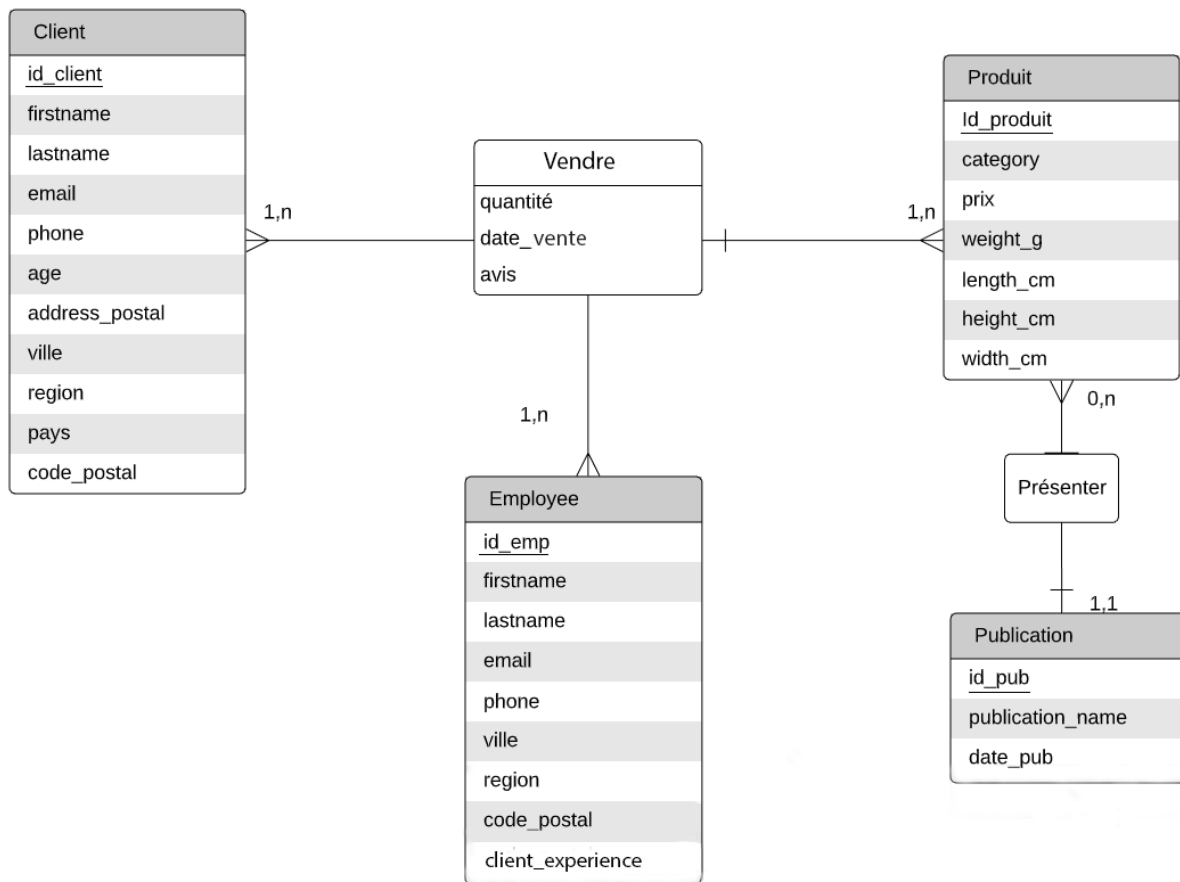
/ Lieu (ville, région, Pays)

/ Date (jds,semaine, mois, trimestre)

1. Étude intégrée des sources de données

Modèle E/A normalisé :

- **Modèle Conceptuelle de données**



• Modèle Logique de données

- ✓ Client (id_client, firstname, lastname, email, phone, age, address_postal, ville, region, pays, code_postal)
- ✓ Produit (id_produit, category, prix, weight_g, length_cm, height_cm, width_cm)
- ✓ Employee (id_emp, firstname, lastname, email, phone, ville, region, code_postal, client_experience)
- ✓ Publication (id_pub, publication_name, date_pub, id_produit)
- ✓ Vente (id_vente, #id_client, #id_produit, #id_emp, quantité, date_vente, avis)

B. Modélisation du data warehouse :

➤ les métadonnées du modèle dimensionnel :

Description de la dimension dw_lieu (DataWarehouse)

Id_lieu	NUMBER(38)	Clé primaire
Ville	VARCHAR2(50)	Nom de la ville d'achat
région	VARCHAR2(50)	Nom de la région d'achat
pays	VARCHAR2(50)	Nom du pays d'achat

Description de la dimension dw Produit (DataWarehouse)

Id_produit	NUMBER	Clé primaire
category	VARCHAR2(50)	Catégorie d'un produit
prix	NUMBER(7,2)	Prix d'un produit
Weight_g	NUMBER(7,2)	Poids d'un produit en gramme
Length_cm	NUMBER(7,2)	Longueur d'un produit en cm
Height_cm	NUMBER(7,2)	taille d'un produit en cm
Width_cm	NUMBER(7,2)	Largeur du produit en cm

Description de la dimension DW CLIENT(DataWarehouse)

Id_client	NUMBER(38)	Clé primaire
Nom_client	VARCHAR2(50)	Nom d'un client
age	NUMBER(38)	Age d'un client

Description de la dimension DW EMPLOYEE(DataWarehouse)

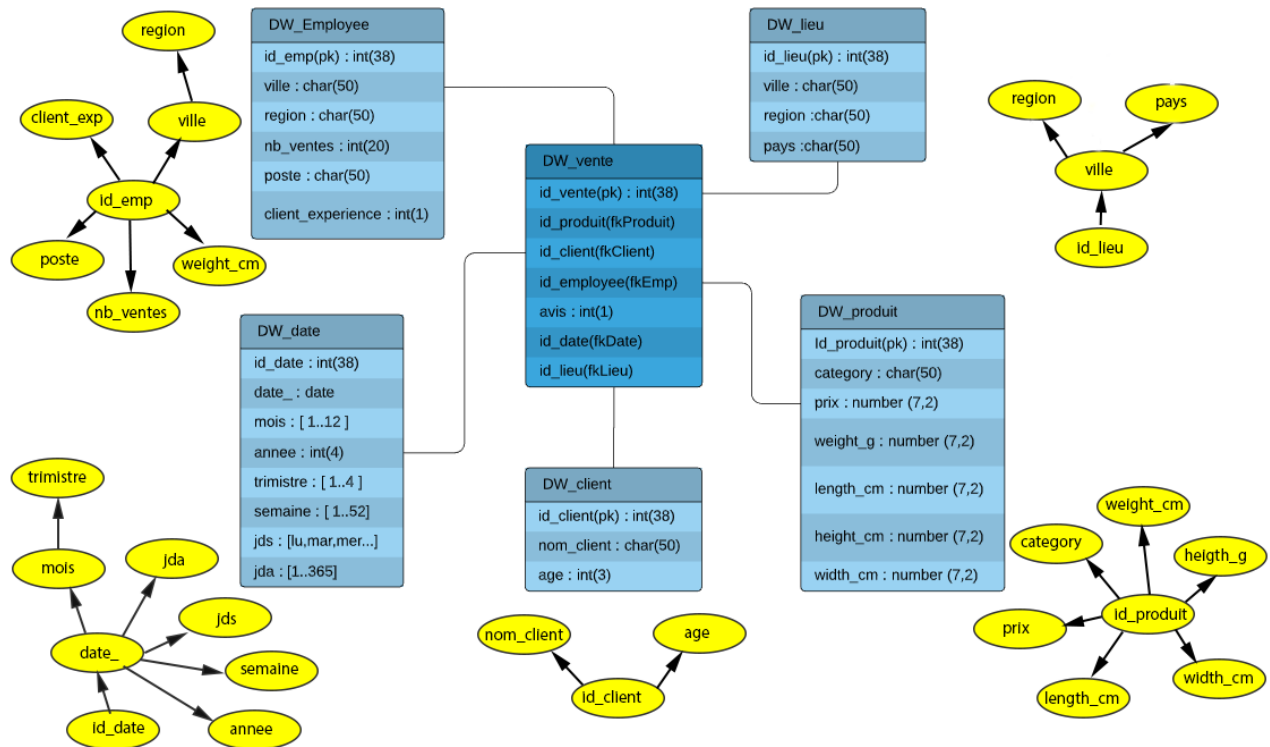
Id_emp	NUMBER(38)	Clé primaire
ville	VARCHAR2(50)	Ville d'un employé
region	VARCHAR2(50)	Région d'un employé
Nb_vente	NUMBER(38)	Nombre de vente d'un employé
Client_experience	VARCHAR2(50)	Expérience d'un employé [0..5]
poste	VARCHAR2(50)	Nom du poste d'un employé

Description de la dimension DW DATE(DataWarehouse)

Id_date	NUMBER(38)	Clé primaire
Date_	VARCHAR2(50)	Date complet d'achat
mois	NUMBER	Mois d'achat
annee	NUMBER	Année d'achat
trimestre	NUMBER	Trimestre d'achat
semaine	NUMBER	Semaine d'achat
jds	VARCHAR2(50)	Jour de la semaine d'achat
jda	NUMBER	Jour de l'année d'achat

- le Modèle dimensionnel

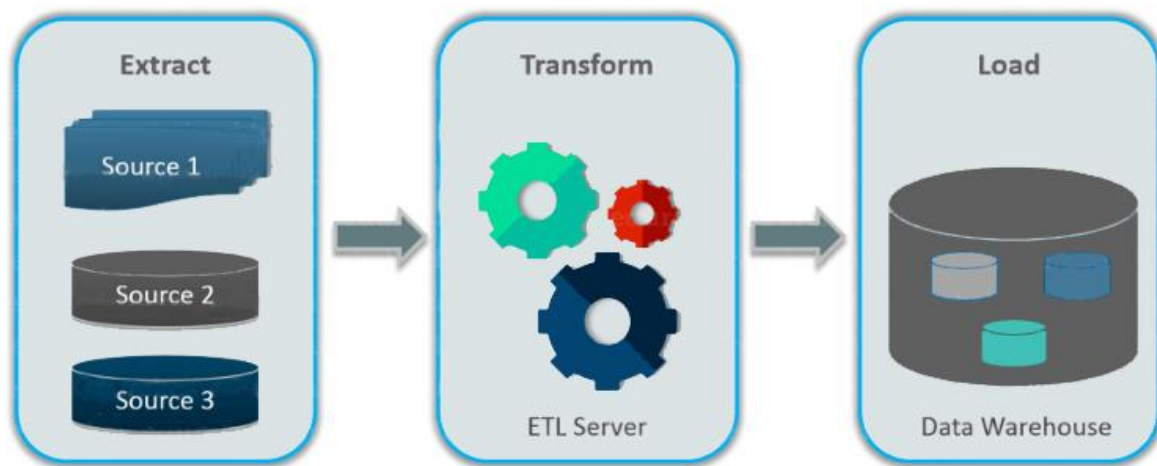
Afin de prendre en considération les besoins réels d'analyse des utilisateurs et les données disponibles et de leur qualité et selon le Diagramme Utilité / Qualité, On Obtient le Modèle dimensionnel finale suivant :



IV. ETL

A. Définition

est le processus qui consiste à rendre des données disponibles en les collectant auprès de sources multiples (dans notre cas on a importé depuis des sources externes qui sont les fichiers 'client.csv', 'employee.csv', 'produit.csv', 'publication.csv' et 'vente.csv' .) et en les soumettant à des opérations de nettoyage, de transformation et, au final, d'analytique métier. ETL signifie Extraire, Transformer et Charger :



Extraire

L'extraction de données est l'étape la plus importante d'ETL qui implique d'accéder aux données de tous les systèmes de stockage. Les systèmes de stockage peuvent être le SGBDR, les fichiers Excel, les fichiers XML, etc.

Transformer

La transformation est le prochain processus en cours. Dans cette étape, des données entières sont analysées et diverses fonctions y sont appliquées pour les transformer au format requis. Généralement, les processus utilisés pour la transformation des données sont la conversion, le filtrage, le tri, la standardisation, la suppression des doublons, la traduction et la vérification de la cohérence des différentes sources de données.

Charger

Le chargement est la dernière étape du processus ETL. Dans cette étape, les données traitées, c'est-à-dire les données extraites et transformées, sont ensuite chargées dans un référentiel de données cible qui est généralement les bases de données.

B. Talend open studio pour l'intégration de données

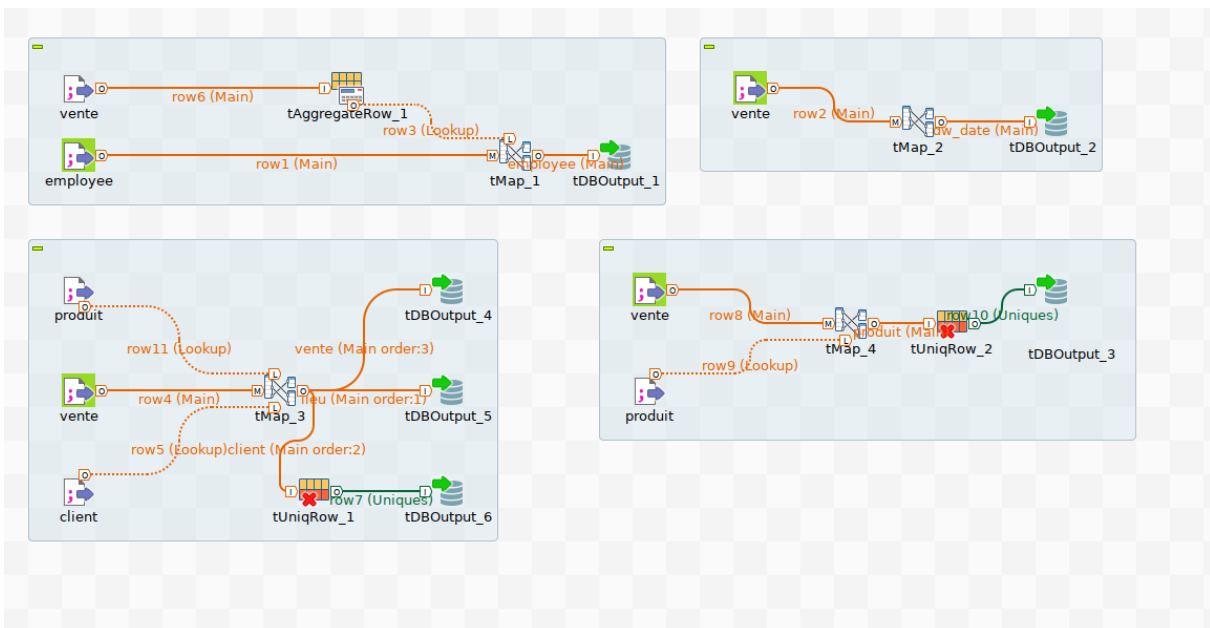
Lorsque tous ces processus sont combinés ensemble dans un *seul outil de programmation* qui peut aider à préparer les données et à gérer diverses bases de données. Ces outils ont des interfaces graphiques qui permettent d'accélérer l'ensemble du processus de mappage des tables et des colonnes entre les différentes bases de données source et cible.

Parmi ces outils il y a Talend open studio pour l'intégration de données est l'un des outils ETL d'intégration de données les plus puissants du marché, il permet de gérer facilement toutes les étapes impliquées dans le processus ETL, depuis la conception ETL initiale jusqu'à l'exécution du chargement des données ETL. Pour notre cas les étapes sont :

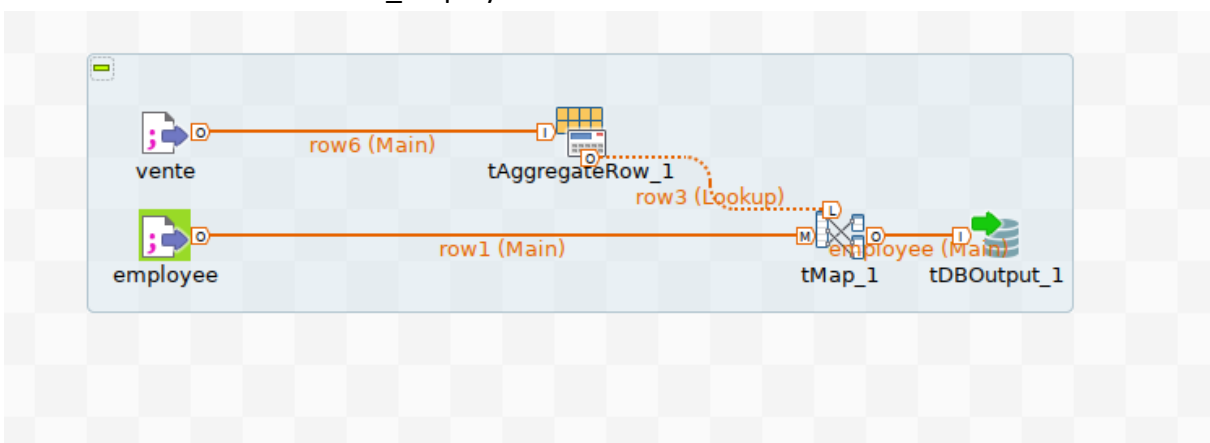
1. les fichiers qu'on a utilisés

- ▼ **Metadata**
 - ▶ **Db Connections**
 - ▼ **File delimited**
 - ▶ client 0.1
 - ▶ employee 0.1
 - ▶ produit 0.1
 - ▶ publication 0.1
 - ▶ vente 0.1
 - ▶ File positional
 - ▶ File regex
 - ▶ File xml
 - ▶ File Excel
 - ▶ File Idif
 - ▶ File Json

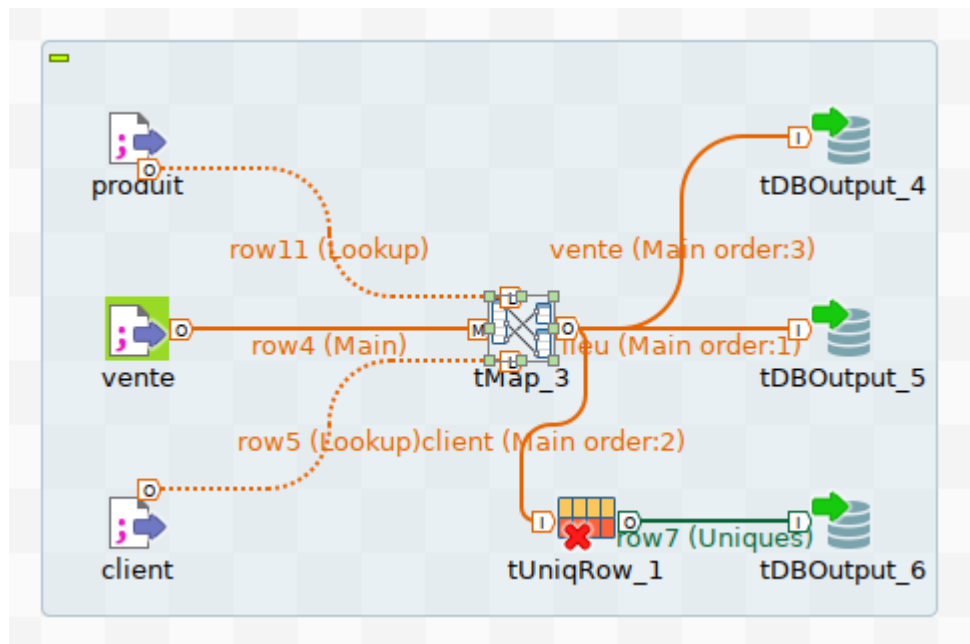
2. Création de Job (avant l'exécution) :



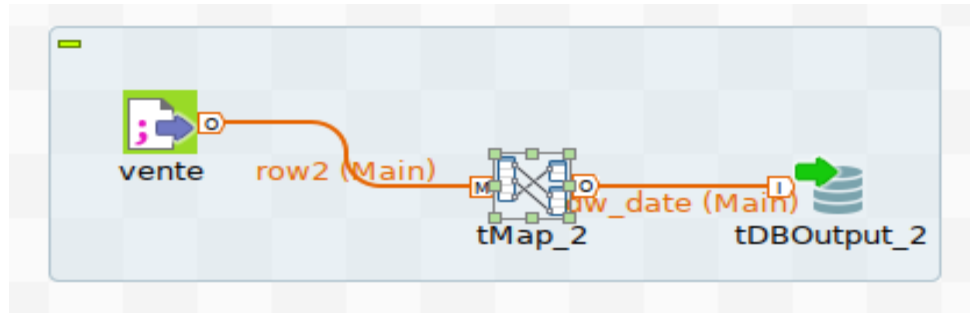
3. Pour créer la table dw_employee :



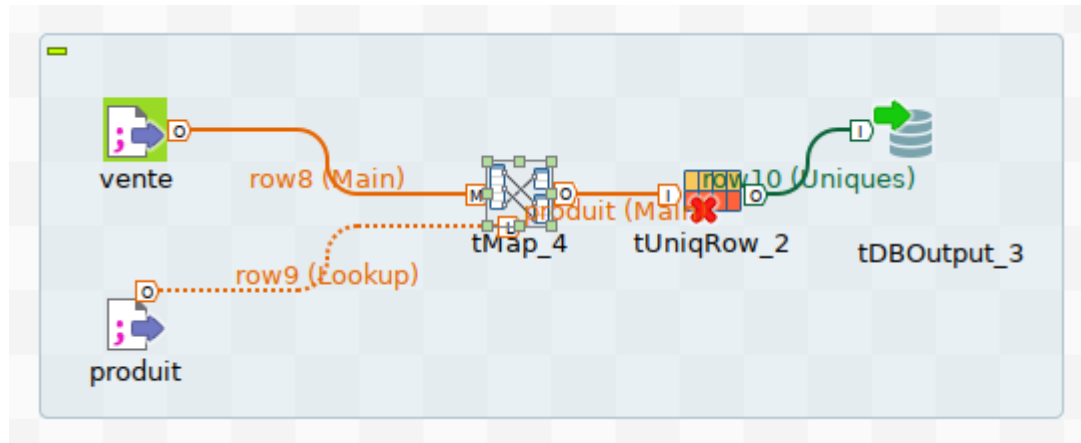
4. Pour créer les tables dw_lieu ,dw_client et dw_vente:



5. Pour créer la table dw_date:



6. Pour créer la table dw_produit:



7. creation de table dw_employee

Talend Open Studio for Data Integration - tMap - tMap_1

Find :

Var

Expression	Type	Variable

Auto map!

row1

Column
id
firstname
lastname
email
phone
ville
region
code_postal
client_experience
Designation_de_poste

row3

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Inner join
Store temp data	false

Expr. key	Column
row1.id	id_employe
	nb_vente

employee

Property	Value
Catch output reject	false
Catch lookup inner join reject	false
Schema Type	Built-In

Expression	Column
row3.id_employe	id_employe
row1.ville	ville
row1.region	region
row1.client_experience	client_experience
row1.Designation_de_poste	Designation_de_poste
row3.nb_vente	nb_vente

Schema editor

row1

Column	Key	Type	Null	Date	Patter	Length	Precisi	Defaut	Comme
id	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			5	0		
firstname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			10	0		
lastname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			10	0		
email	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			49	0		
phone	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			14	0		
ville	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			21	0		

employee

Column	Key	Type	Null	Date	Patter	Length	Precisi	Defaut	Comme
id_employe	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			5	0		
ville	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			30	0		
region	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			2	0		
client_experience	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			1	0		
Designation_de_pc	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			40	0		
nb_vente	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			3	0		

8. création des tables dw_lieu et dw_client et dw_vente

Talend Open Studio for Data Integration - tMap - tMap_3

Find :

Var

Auto map!

row4

Column
id_service
id_product
id_client
id_employe
quantite
avis
date_service

row5

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Inner join
Store temp data	false

Expr. key	Column
row4.id_client	id
	firstname
	lastname
	email
	phone
	age
	address_postal
	ville
	region
	pays
	code_postal

row11

Property	Value
Lookup Model	Load once
Match Model	Unique match

lieu

Expression	Column
row4.id_service	id_lieu
row5.address_postal	address_postal
row5.ville	ville
row5.region	region
row5.pays	pays

client

Expression	Column
row5.firstname + " " + row5.lastname	name
row5.age	age
row5.id	id

vente

Expression	Column
row4.id_service	id_vente
row4.id_product	id_product
row4.id_client	id_client
row4.id_employe	id_employe
row4.avis	avis
row4.id_service	id_date
row4.id_service	id_lieu
row4.quantite * row11.prix	CA

Schema editor

9. création de table dw_date

Talend Open Studio for Data Integration - tMap - tMap_2

Find :

Auto map!

row2

Column

- id_service
- id_product
- id_client
- id_employe
- quantite
- avis
- date_service

dw_date

Expression

- row2.id_service
- TalendDate.parseDate("dd/MM/yy")
- TalendDate.getPartOfDate("MONTH")
- TalendDate.getPartOfDate("YEAR")
- TalendDate.getPartOfDate("WEEK")
- TalendDate.getPartOfDate("WEEK")
- TalendDate.getPartOfDate("DAY_OF_WEEK")
- (TalendDate.getPartOfDate("WEEK")

Column

- id_date
- date
- mois
- annee
- trimestre
- semaine
- jds
- jda

Schema editor

row2

Column	Key	Type	Null	Date	Patter	Length	Precisi	Defai	Comme
id_service	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			5	0		
id_product	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			5	0		
id_client	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			5	0		
id_employe	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			5	0		
quantite	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			1	0		
avis	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			1	0		
date_service	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			19	0		

dw_date

Column	Key	Type	Null	Date	Patter	Length	Precisi	Defai	Comme
id_date	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			19	0		
date	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>		"dd-MM-yyyy"	19	0		
mois	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			19	0		
annee	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			19	0		
trimestre	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			19	0		
semaine	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			19	0		
jds	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			19	0		
jda	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			19	0		

10. création de table dw_produit

Talend Open Studio for Data Integration - tMap - tMap_4

Find : Var Auto map!

row8

Column
id_service
id_product
id_client
id_employe
quantite
avis
date_service

row9

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Inner join
Store temp data	false
Expr. key	Column

row8.id_product → produit_id

row9.produit_category_name → produit_category_n

row9.prix → prix

row9.produit_weight_g → produit_weight_g

row9.produit_length_cm → produit_length_cm

row9.produit_height_cm → produit_height_cm

row9.produit_width_cm → produit_width_cm

produit

Expression	Column
row8.id_product	produit_id
row9.produit_category_name	produit_category_n
row9.prix	prix
row9.produit_weight_g	produit_weight_g
row9.produit_length_cm	produit_length_cm
row9.produit_height_cm	produit_height_cm
row9.produit_width_cm	produit_width_cm

Schema editor

row9

Column	Key	Type	Null	Date	Patter	Length	Precisi	Defai	Comme
produit_id	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			2	0		
produit_category_n	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			39	0		
prix	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			6	3		
produit_weight_g	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			7	3		
produit_length_cm	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			4	2		
produit_height_cm	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			4	2		

produit

Column	Key	Type	Null	Date	Patter	Length	Precisi	Defai	Comme
produit_id	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			2	0		
produit_category_n	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			39	0		
prix	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			8	3		
produit_weight_g	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			8	3		
produit_length_cm	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			8	2		
produit_height_cm	<input type="checkbox"/>	Float	<input checked="" type="checkbox"/>			8	2		

11. configuration de tdboutput

Job(Job 0.1) Contexts(Job) Component x Run (Job Job)

tDBOutput_2(MySQL)

Basic settings

Database: MySQL Apply

Property Type: Built-In

DB Version: Mysql 8

☐ Use an existing connection

Host: "127.0.0.1" * Port: "3306" *

Database: "olist" *

Username: "root" * Password: "*****" *

Table: "dw_date" ...

Action on table: Drop table if exists and create Action on data: Insert

Schema: Built-In Edit schema Sync columns

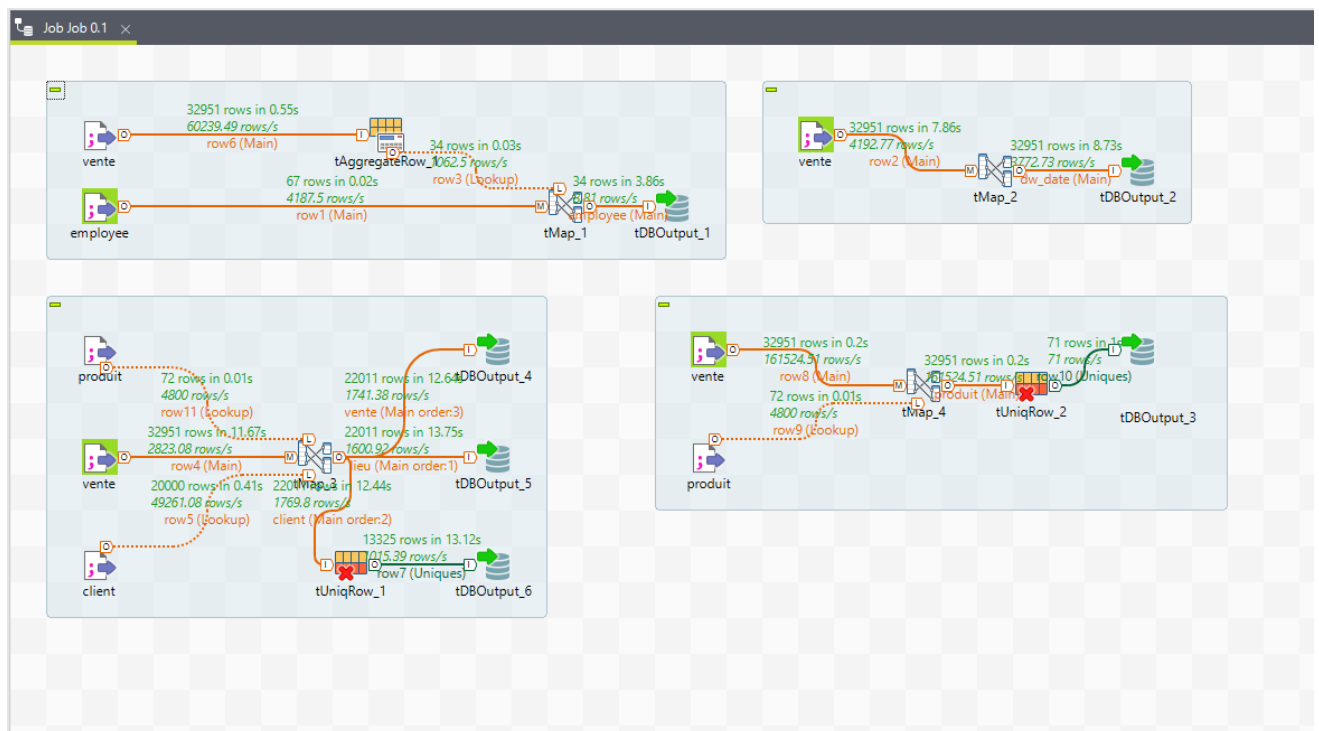
Data source

This option only applies when deploying and running in the Talend Runtime

☐ Specify a data source alias

☐ Die on error

12. job (après l'exécution)



13. les tables créées après l'exécution

Server: 127.0.0.1 » Database: olist

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> dw_client	Browse Structure Search Insert Empty Drop	13,325	InnoDB	utf8mb4_general_ci	1.5 MiB	-
<input type="checkbox"/> dw_date	Browse Structure Search Insert Empty Drop	32,951	InnoDB	utf8mb4_general_ci	2.5 MiB	-
<input type="checkbox"/> dw_employee	Browse Structure Search Insert Empty Drop	34	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> dw_lieu	Browse Structure Search Insert Empty Drop	22,011	InnoDB	utf8mb4_general_ci	2.5 MiB	-
<input type="checkbox"/> dw_produit	Browse Structure Search Insert Empty Drop	71	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> dw_vente	Browse Structure Search Insert Empty Drop	22,011	InnoDB	utf8mb4_general_ci	1.5 MiB	-
6 tables	Sum	90,403	InnoDB	utf8mb4_general_ci	8.1 MiB	0 B

Check all With selected:

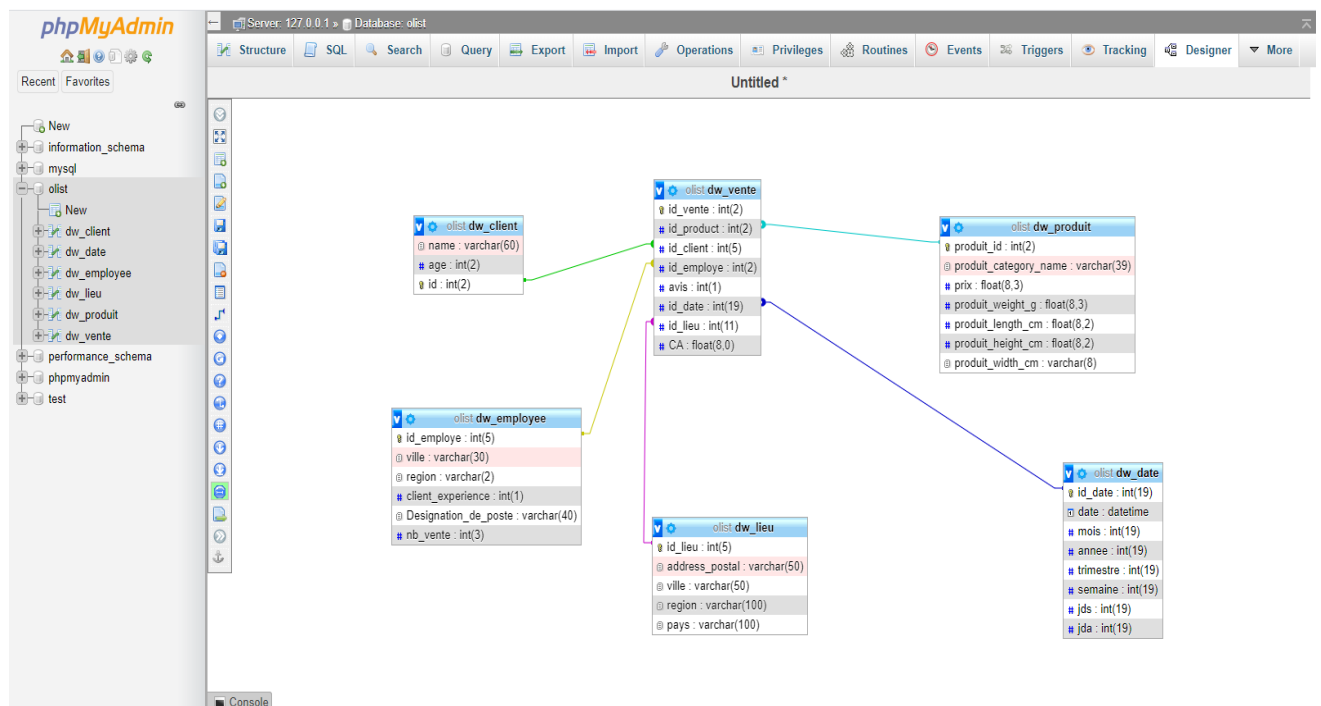
Print Data dictionary

Create table

Name: Number of columns: 4

Go

14. vue général de la base olist

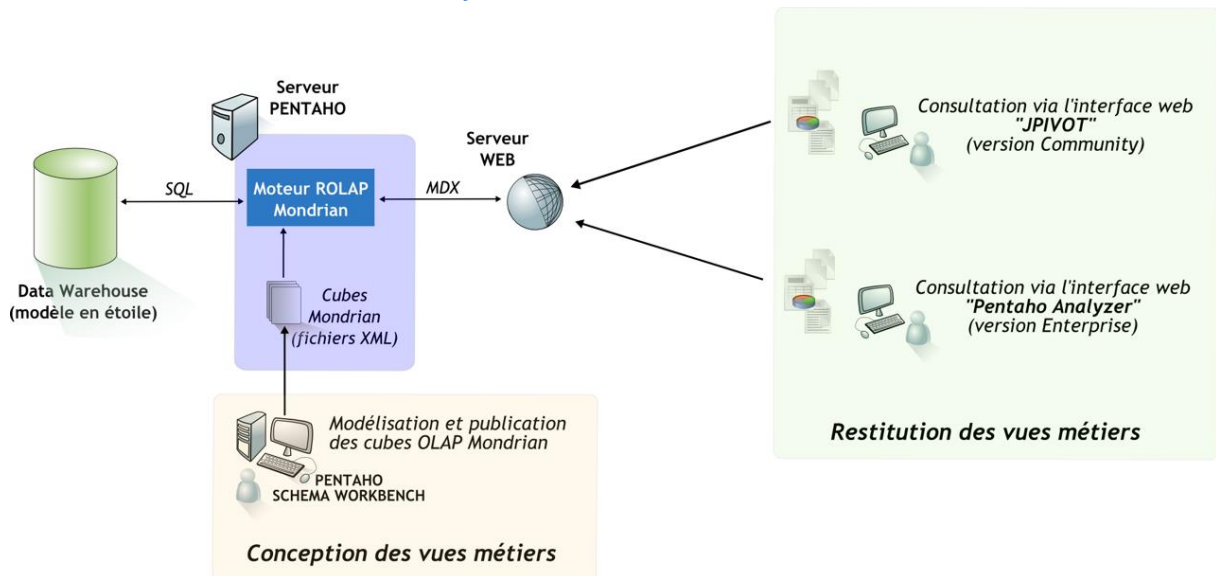


V. OLAP (online analytical processing)

A. Définition

Le OLAP, ou Online Analytical Processing, est une **technologie de traitement informatique** (computer processing). Elle permet à un utilisateur de consulter et **d'extraire facilement les données pour les comparer de différentes façons**. C'est un outil inscrit dans analysis services d'aide à la décision .Les données OLAP sont stockées sur une **base multidimensionnelle**, aussi appelées Cubes OLAP, pour faciliter ce type d'analyses. Un **serveur OLAP** est nécessaire.

B. OLAP, comment ça marche ?



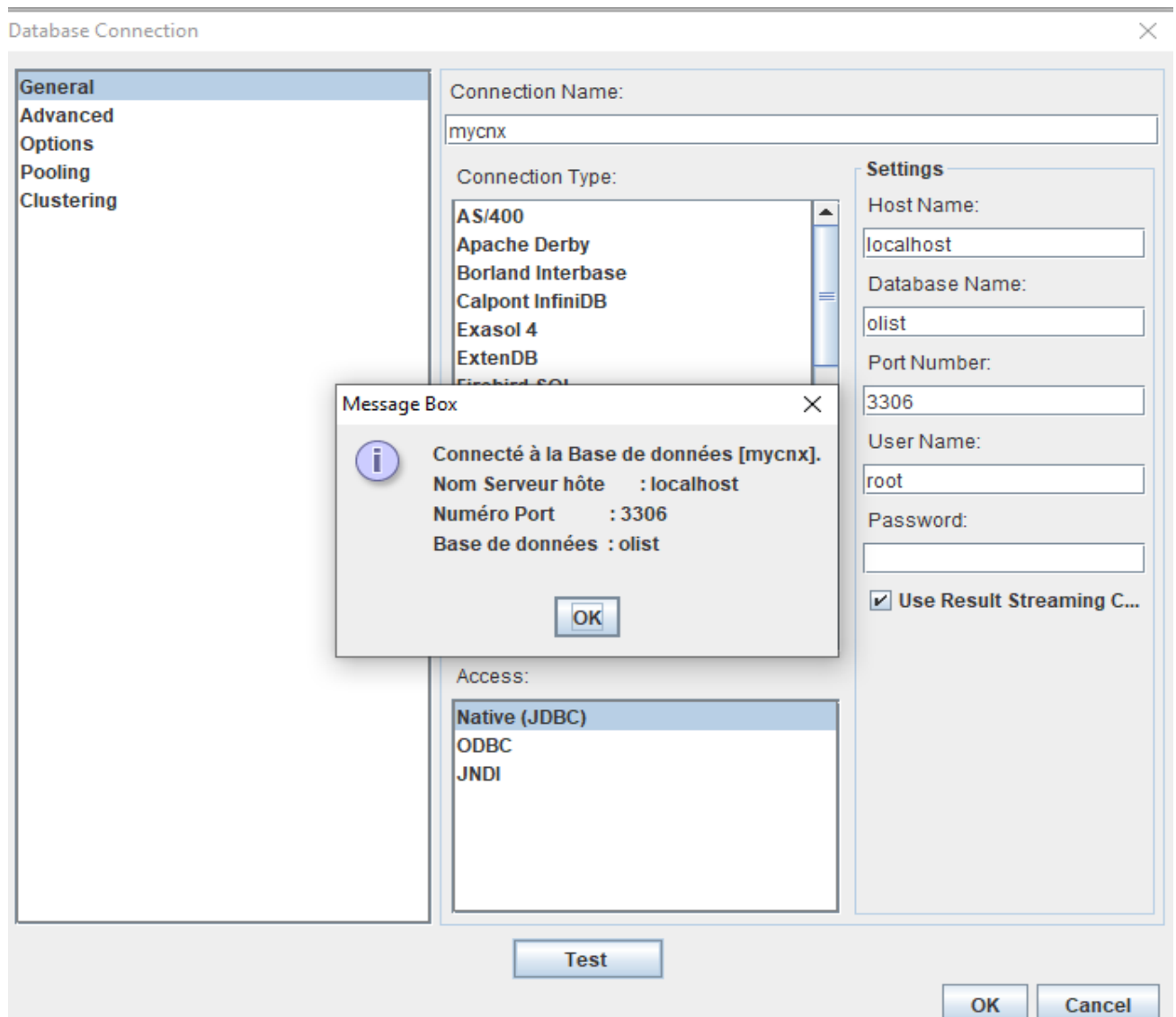
Les éléments caractéristiques de l'architecture présentée ci-dessus sont les suivants :

- **Datawarehouse** avec stockage des données au format OLAP. Les schémas en étoile et en flocons sont gérés par Mondrian, ainsi que de nombreux autres concepts propres à l'OLAP
- **Serveur web J2EE Pentaho**. Ce serveur embarque le moteur ROLAP Mondrian qui permet d'effectuer des requêtes multi-dimensionnelles (langage MDX) sur des données stockées dans un SGBD relationnel (interrogation SQL).
- Pour la partie conception, les cubes sont modélisés avec « **Pentaho Schema Workbench** », un client riche open source (en java) de modélisation et de publication de schémas Mondrian sur un serveur Pentaho
- Pour la partie **restitution**, les vues métiers sont accessibles en client léger via un navigateur web (Internet, Explorer, Firefox, Safari, Opera, ...). Pour effectuer des analyses avancées, on utilisera **JPivot**, ou bien Pivot4J qui sont des outils de requêtage puissants et stables.

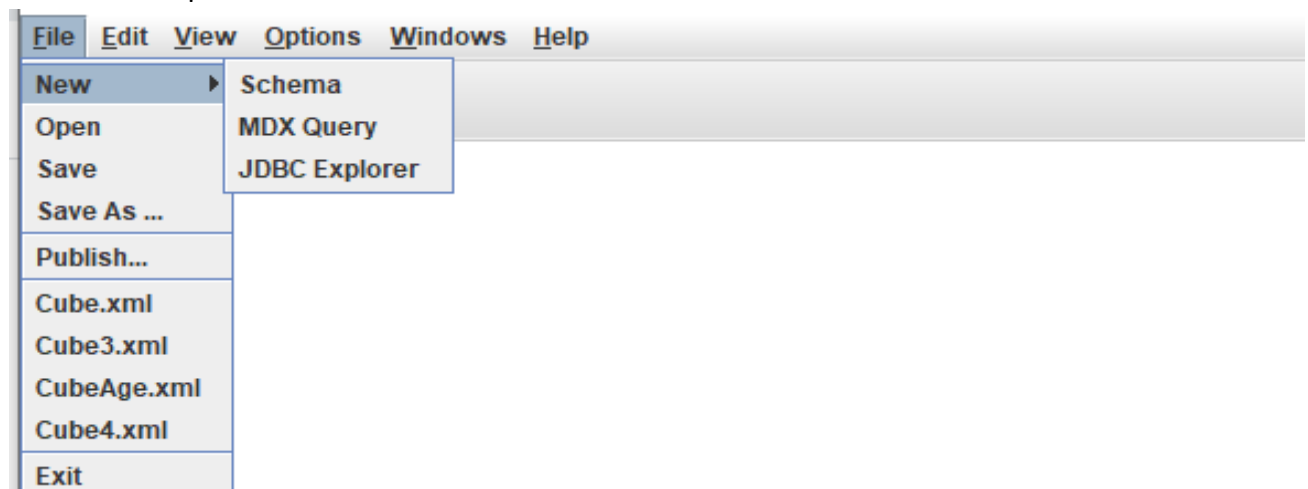
C. Modélisation et publication de cube OLAP

Avec un modèle de données physique multidimensionnel en place, nous devons créer un modèle logique qui lui correspond. Un schéma Mondrian est essentiellement un fichier XML qui effectue ce mappage, définissant ainsi une structure de base de données multidimensionnelle. Nous pouvons créer des schémas Mondrian à l'aide du **Pentaho Schema Workbench**, pour cela on procède comme suit :

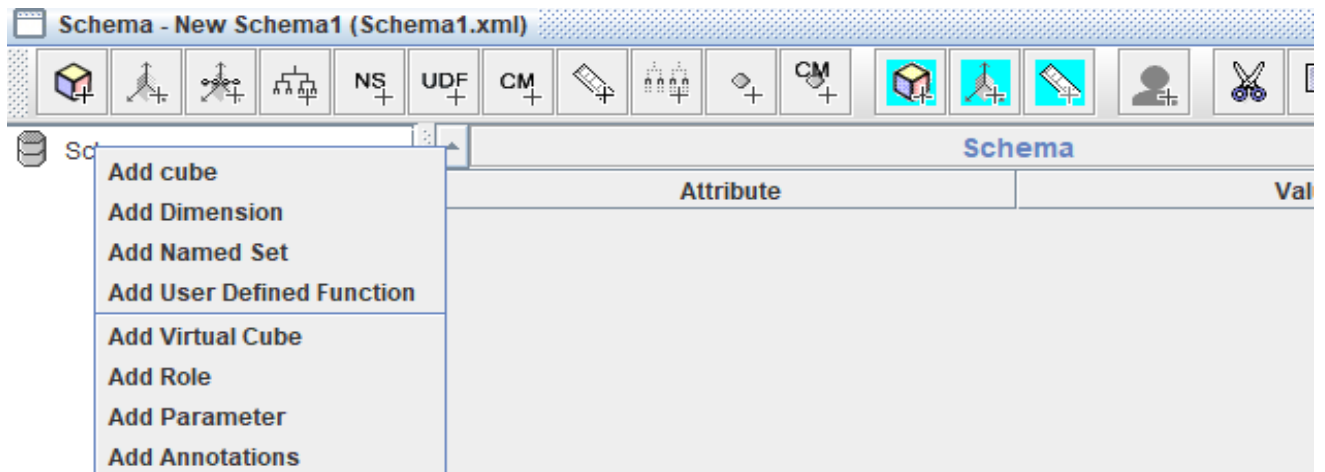
1. création de la connexion avec MySQL



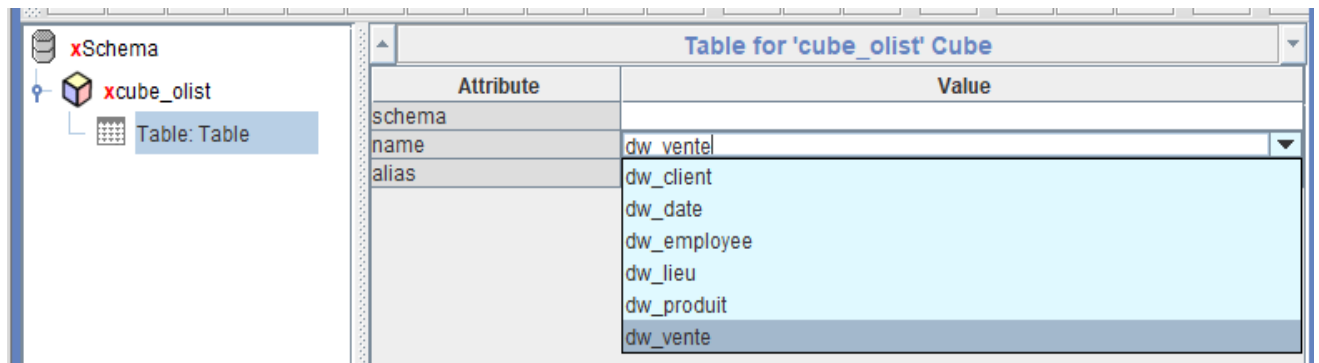
2. On clique sur **File> New> New schéma**



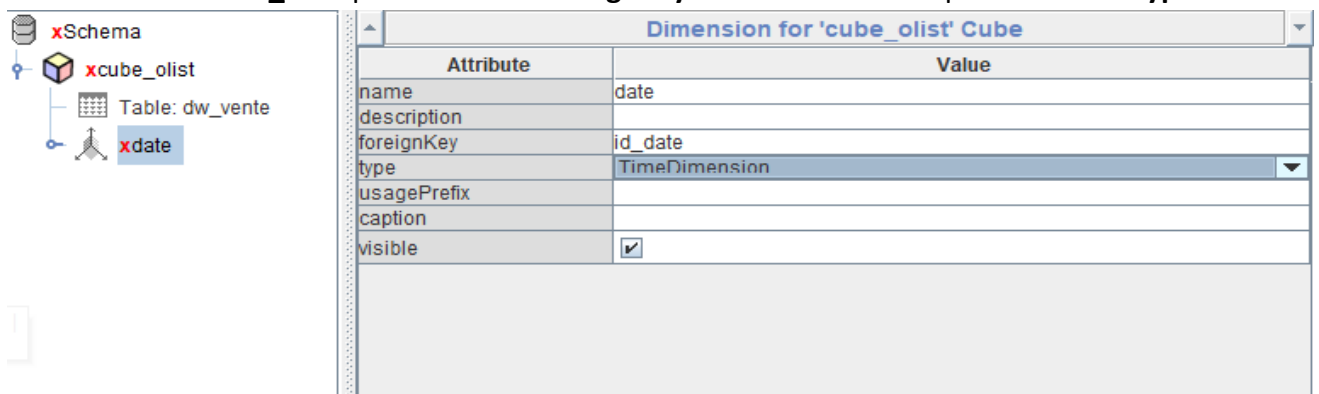
3. On fait un clic droit sur le schéma **Schema** et On fait **Add cube** on clique sur le **cube** et on écrit l'attribut **cube_olist** pour le **nom**



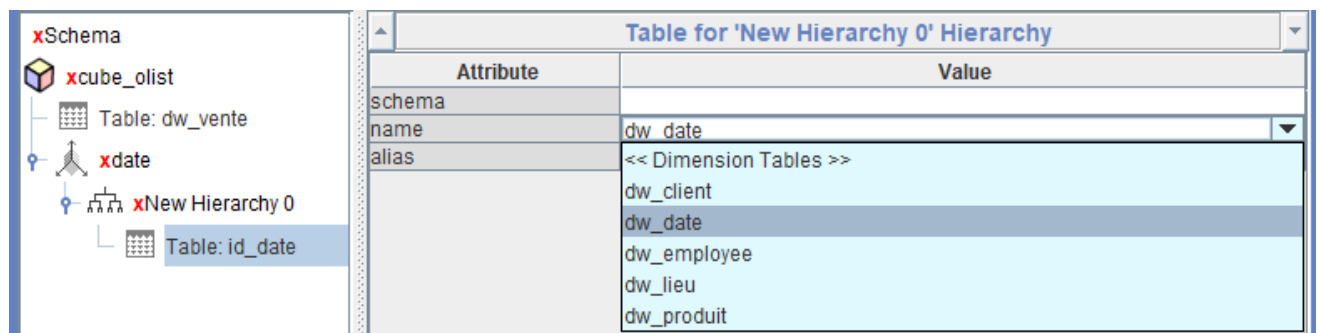
4. On fait un clic droit sur le cube **cube_olist** et On fait **Add table** (la table de fait). On clique sur **la table** et choisissez la table **dw_vente** pour l'attribut de **name**.



5. On fait un clic droit sur le cube **cube_olist** et faites **Add dimension**. On clique sur **New dimension 1** et on écrit **Date** pour l'attribut de **name**. On choisit la colonne **id_date** pour l'attribut **foreignKey** et **TimeDimension** pour l'attribut **type**.

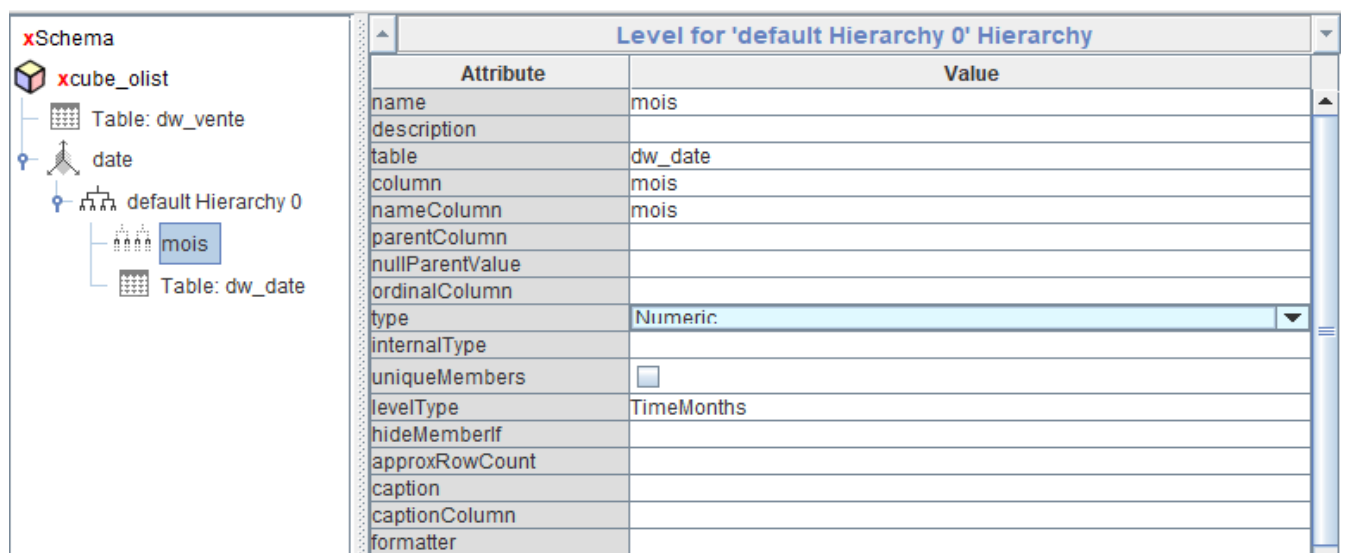


6. On fait un clic droit sur la hiérarchie **New Hierarchy 0** et on fait **Add table**. Cliquez sur **la table** et choisir la table **dw_date** pour l'attribut de **name**.



7. On fait un clic droit sur la hiérarchie **New Hierarchy 0** et on fait **Add level** . On clique sur **New Level 1** et remplissez les champs suivants:

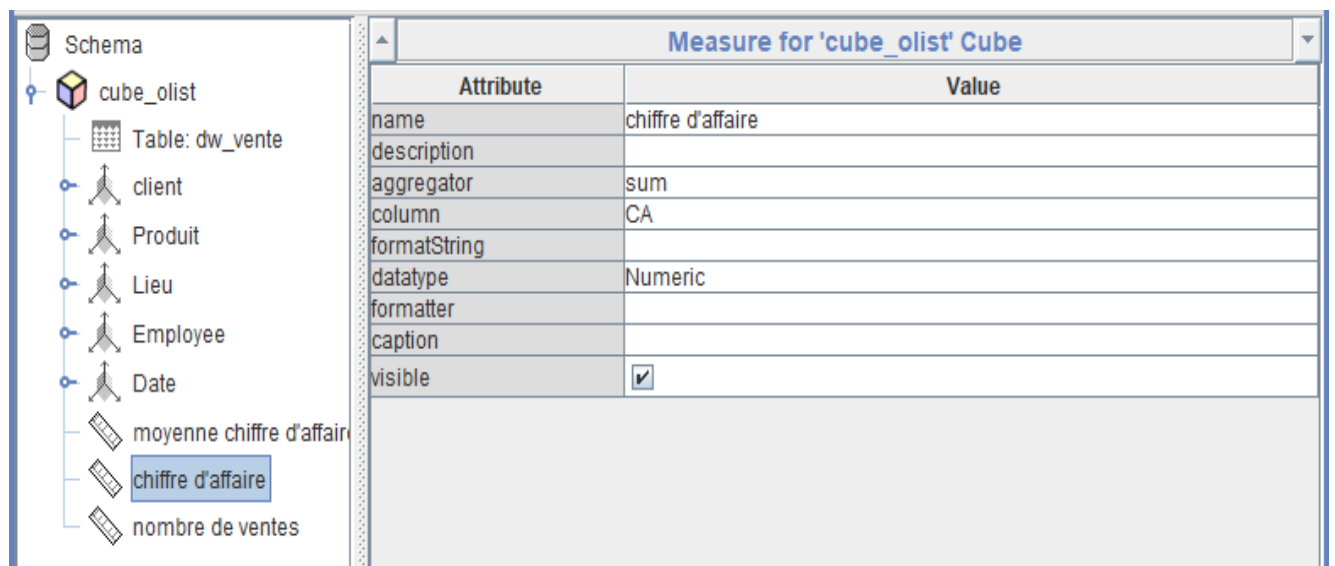
Name	mois
Table	Dw_date
Column	Mois
nameColumn	Mois
Type	Numeric
levelType	TimeMonths



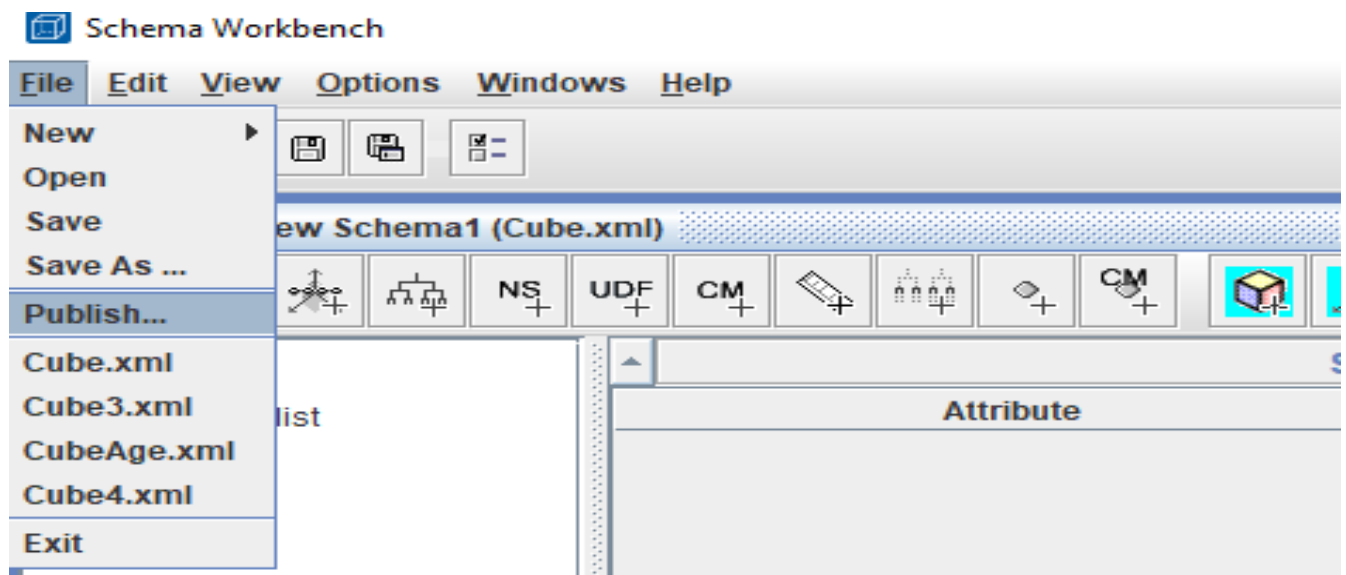
8. On répète l'étape précédente 7 pour créer des **niveaux** comme **trimestre** et **année**.
9. On répète les étapes de 5 à 8 pour créer les autres dimensions **Employée**, **Client**, **Lieu**, et **Produit**.

10. On fait un clic droit sur le cube **cube_olist** et on fait 3 fois **Add Measure**. On clique sur **New Measure 0** et **New Measure 1** et **New Measure 2** puis on remplit les champs suivants:

Name	Moyen chiffre d'affaire	Chiffre d'affaire	Nombre de ventes
Aggregator	AVG	Sum	count
column	CA	CA	Id_product



11. On Clique sur **FIILE>Publish** pour publier le cube sur pentaho server



VI. Reporting

Après la publication de cube sur Pentaho server on va le manipuler avec Pivot4J qui fournit un plugin pour le serveur Pentaho BI et qui permet la création de tables OLAP et permet à l'utilisateur d'utiliser les fonctions classiques de l'analyse multidimensionnelle pour la création des rapports



MDX :

Multi Dimensional Expression ou MDX est un langage de requête développé par Microsoft pour manipuler des informations multidimensionnelles (cubes de données). Pentaho avec sa prise en charge Pivot4J Requetes MDX. Pour utiliser MDX, commencez simplement à taper la syntaxe de la requête dans le champ de requête MDX situé en bas de la fenêtre.

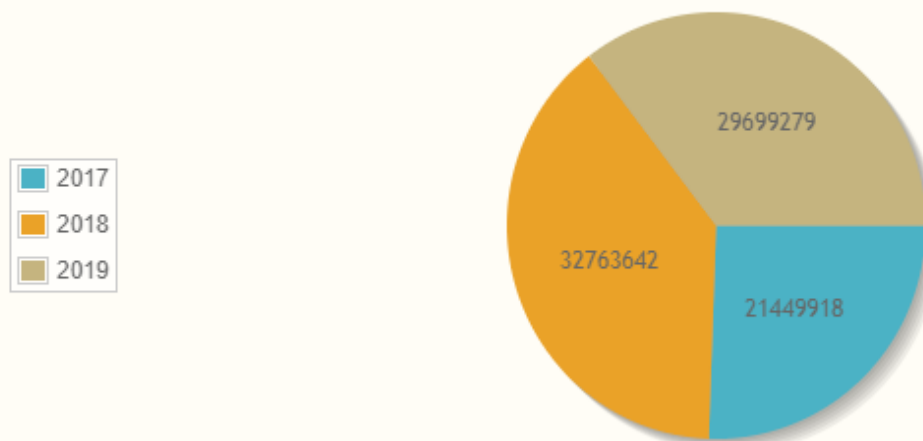
De plus, MDX dans Pentaho est dynamique. Chaque fois qu'on fait glisser et déposez des dimensions ou développez et réduisez le hiérarchique, Pivot4J fournit l'instruction MDX correspondante.

Exemples de reporting sur la base de données olist :

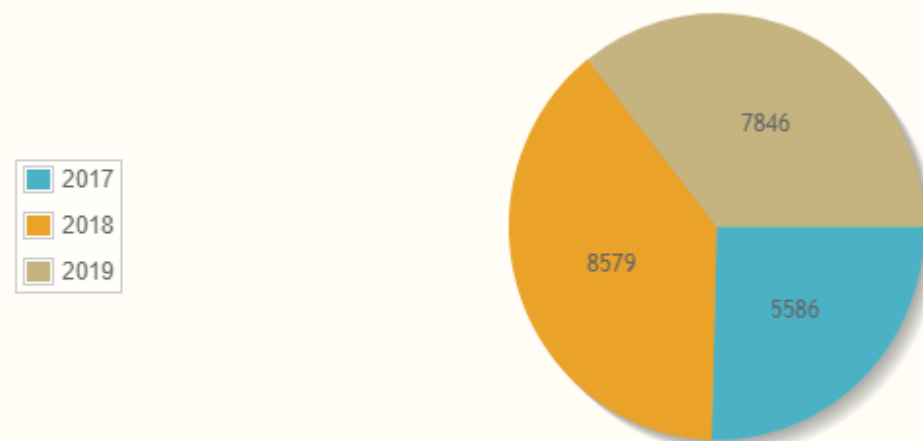
- **Les mesures en fonction de la dimension date :**

	Measures	
Date	 nombre de ventes	 chiffre d'affaire
2017	5 586	21 449 918
2018	8 579	32 763 642
2019	7 846	29 699 279
Total	22 011	83 912 839

chiffre d'affaire



nombre de ventes



- Les mesures en fonction de dimension Produit :

Produit		Mesures		
(All)	id_produit	nombre de ventes	totale de chiffre d'affaire	moyenne chiffre d'affaire
All Produits	+ 36	307	4,541,367	14,792.726
	+ 71	329	3,430,815	10,428.009
	+ 45	342	3,477,390	10,167.807
	+ 66	294	2,958,670	10,063.503
	+ 14	321	3,139,460	9,780.249
	+ 12	297	2,361,348	7,950.667
	+ 25	302	2,312,805	7,658.295
	+ 7	286	1,965,240	6,871.468
	+ 54	309	2,033,236	6,580.052
	+ 16	287	1,735,196	6,045.979
	+ 33	342	2,065,248	6,038.737
	+ 42	339	2,028,739	5,984.481
	+ 10	322	1,871,564	5,812.311

La requête SQL sous MySQL pour la vérification(id_produit=36) :

✓ Showing rows 0 - 0 (1 total, Query took 0.0449 seconds.)

```
SELECT count(*) as nbVentes,sum(CA) as CA ,avg(CA) as AVG FROM `dw_vente` WHERE id_product=36
```

☐ Show all | Number of rows: 25 ▼ Filter rows:

+ Options

nbVentes	CA	AVG
307	4541367	14792.7264

- Les mesures (nombre de vente et CA) en fonction de dimension Lieu :

Afghanistan	97	359 048
Åland Islands	100	398 687
Albania	85	299 494
Algeria	107	433 181
AlloaClackmannanshireKorea South	1	8 519
American Samoa	119	460 059
Andorra	91	296 263
Angola	60	235 968
Anguilla	70	252 521
Antarctica	67	271 954
Antigua and Barbuda	82	375 897
ApartadóAntioquiaVirgin Islands British	1	3 829
Argentina	98	359 980
Armenia	116	488 313
Aruba	76	270 850
Australia	86	324 206
Austria	108	461 683
Azerbaijan	74	296 654
Bahamas	83	344 269
Bahrain	75	259 018
Bangladesh	97	368 778
Barbados	74	238 805

- Les mesures en fonction de 2 dimensions Produit (catégorie) et Client(âge):

Produit			Client		Mesures	
(All)	id_produit	catégorie	(All)	age	nombre de ventes	chiffre d'affaire
All Produits	1	perfumery	All client.Clients	24	10	16 933
				25	3	2 870
				26	3	5 453
				27	4	3 731
				28	2	3 731
				29	2	4 018
				30	4	6 314
				31	4	6 027
				32	2	1 148
				33	12	20 377
				34	4	8 036
				35	9	8 897
				36	5	6 000

VII. Data Mining

Pour le data mining on souhaite faire la régression sur le nombre de vente en fonction du prix ,poids, longueur, taille et la largeur de produit et mois ,trimestre , et année de vente ,pour cela on a suivit les étapes suivantes :

1. Exporter les données vers un fichier excel :

Produit								Date				Mesures		
(All)	id_produit	catégorie	prix	weight_g	length_cm	height_cm	width_cm	(All)	mois	trimestre	annee	moyenne chiffre d'affaire	chiffre d'affaire	nombre de ventes
All Produits	1	perfumery	287	225	16	10	14.0	All Dates	1	1	2018	1 387,167	16 646	12
											2019	1 722	17 220	10
									2	1	2018	1 757,875	14 063	8
											2019	1 578,5	15 785	10
									3	1	2018	1 607,2	8 036	5
											2019	1 894,2	18 942	10
										2	2018	2 296	2 296	1
											2019	1 004,5	2 009	2
									4	2	2018	1 722	20 664	12
											2019	1 170,889	10 619	9
									5	2	2017	1 690,111	15 211	9
											2018	1 578,5	9 471	6
											2019	1 783	12 341	7
									6	2	2017	1 513,273	16 646	11
											2018	1 779,4	8 897	5
											2019	975,8	4 879	5
										3	2017	287	287	1
											2018	1 626,333	4 879	3
											2019			
									7	3	2017	1 243,667	11 193	9
											2018	1 757,875	14 063	8
											2019	1 804	25 256	14
									8	3	2017	1 052,333	6 314	6
											2018	1 243,667	14 924	12

2. Preprocessing data:

```
In [6]: 1 data = pd.DataFrame(data=prepr,columns=["id_produit","catégorie","prix","weight_g","length_cm","height_cm","wid
2 data
```

Out[6]:

	id_produit	catégorie	prix	weight_g	length_cm	height_cm	width_cm	mois	trimestre	annee	moyenne chiffre d'affaire	chiffre d'affaire	nombre de ventes
0	1	perfumery	287	225	16	10	14	1	1	2018	1 387,167	16 646	12
1	1	perfumery	287	225	16	10	14	1	1	2019	1 722	17 220	10
2	1	perfumery	287	225	16	10	14	2	1	2018	1 757,875	14 063	8
3	1	perfumery	287	225	16	10	14	2	1	2019	1 578,5	15 785	10
4	1	perfumery	287	225	16	10	14	3	1	2018	1 607,2	8 036	5
...
2977	71	furniture_mattress_and_upholstery	2073	1850	41	21	21	12	1	2017	0	0	0
2978	71	furniture_mattress_and_upholstery	2073	1850	41	21	21	12	1	2018	14 511	14 511	1
2979	71	furniture_mattress_and_upholstery	2073	1850	41	21	21	12	4	2017	10 986,9	109 869	10
2980	71	furniture_mattress_and_upholstery	2073	1850	41	21	21	12	4	2018	11 684,182	128 526	11
2981	71	furniture_mattress_and_upholstery	2073	1850	41	21	21	12	4	2019	18 657	37 314	2

2982 rows × 13 columns

3. Importing the dataset

Importing the dataset

```
In [8]: 1 X = np.array(data[['prix', 'weight_g', 'length_cm', 'height_cm', 'width_cm', 'mois', 'trimestre', 'annee']])
2 y = np.array(data["nombre de ventes"]).astype('int')
```

4. Feature Scaling

Feature Scaling

```
In [10]: 1 from sklearn.preprocessing import StandardScaler
2
3 sc_X = StandardScaler()
4 X_train = sc_X.fit_transform(X_train)
5 X_test = sc_X.transform(X_test)
```

5. Splitting the dataset into the Training set and Test set

Splitting the dataset into the Training set and Test set

```
In [9]: 1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

6. Result of Random Forest(regression)

7.

Random Forest

```
In [11]: 1 from sklearn.ensemble import RandomForestRegressor
2
3 clf1 = RandomForestRegressor(n_estimators = 1000, random_state = 42)
4 clf1.fit(X_train, y_train)
5
6
7 sum = 0
8 for x_p, y_p in zip(X_train, y_train):
9     sum = sum + (abs(clf1.predict([x_p]) - y_p))
10
11 print("Erreur d'apprentissage:", (sum/np.sum(y_train)*100)[0], "%")
12
13 sum = 0
14 for x_p, y_p in zip(X_test, y_test):
15     sum = sum + abs((clf1.predict([x_p]) - y_p))
16
17 print("Erreur validation: ", (sum/np.sum(y_test)*100)[0], "%")
```

Erreur d'apprentissage: 0.22511980830670952 %
Erreur validation: 1.3227972339950913 %

8. Result of SVR (regression)

SVR

```
In [12]: 1 from sklearn.svm import SVR
2
3 clf2 = SVR()
4 clf2.fit(X_train, y_train)
5
6
7 sum = 0
8 for x_p, y_p in zip(X_train, y_train):
9     sum = sum + (abs(clf2.predict([x_p]) - y_p))
10
11 print("Erreur d'apprentissage:", (sum/np.sum(y_train)*100)[0], "%")
12
13 sum = 0
14 for x_p, y_p in zip(X_test, y_test):
15     sum = sum + abs((clf2.predict([x_p]) - y_p))
16
17 print("Erreur validation: ", (sum/np.sum(y_test)*100)[0], "%")
```

Erreur d'apprentissage: 3.2756957626725787 %
Erreur validation: 3.021850797385196 %

9. Resultat de KNN

KNN

```
In [13]: 1 from sklearn.neighbors import KNeighborsClassifier
2
3 clf3 = KNeighborsClassifier(n_neighbors=1)
4 clf3.fit(X_train, y_train)
5
6
7 sum = 0
8 for x_p, y_p in zip(X_train, y_train):
9     sum = sum + (abs(clf3.predict([x_p]) - y_p))
10
11 print("Erreur d'apprentissage:", (sum/np.sum(y_train)*100)[0], "%")
12
13 sum = 0
14 for x_p, y_p in zip(X_test, y_test):
15     sum = sum + abs((clf3.predict([x_p]) - y_p))
16
17 print("Erreur validation: ", (sum/np.sum(y_test)*100)[0], "%")
```

Erreur d'apprentissage: 0.0 %
Erreur validation: 8.364934195850992 %