

**UNIVERSITÉ MOHAMMED V**  
Faculté des Sciences de Rabat



Université Mohammed V  
Faculté des Sciences  
Rabat

Master Ingénierie de Données et  
Développement Logiciel

*Business Intelligence*

## PROJET DE FIN MODULE

Mise en place d'un système décisionnel pour  
l'analyse des données

Réalisé par  
Hamza TALAGHZI

Encadré par  
Houssaine ZIYATI

## Table des matières

---

<b>1</b>	<b>Problématique</b>	<b>6</b>
<b>2</b>	<b>Conception de l'entrepôt de données Omrane</b>	<b>7</b>
<b>3</b>	<b>Préparation de l'environnement</b>	<b>10</b>
3.1	Pentaho Data Integration . . . . .	10
3.1.1	Fiche technique : . . . . .	10
3.1.2	Étapes d'installation : . . . . .	11
3.2	HeidiSQL . . . . .	11
3.2.1	Fiche technique : . . . . .	12
3.2.2	Étapes d'installation : . . . . .	12
3.3	Schema Workbench . . . . .	12
3.3.1	Fiche technique : . . . . .	13
3.3.2	Étapes d'installation : . . . . .	13
3.4	Pentaho Server . . . . .	13
3.4.1	Fiche technique : . . . . .	13
3.4.2	Étapes d'installation : . . . . .	13
3.5	Jenkins . . . . .	14
3.5.1	Fiche technique : . . . . .	14
3.5.2	Étapes d'installation : . . . . .	14
<b>4</b>	<b>Création de la base de données cible</b>	<b>16</b>
<b>5</b>	<b>Manipulation des données avec l'outil PDI</b>	<b>17</b>
5.1	Créer une connexion JDBC dans PDI . . . . .	17
5.2	Extraire les données Excel . . . . .	18
5.3	Sélectionner le flux de données . . . . .	20
5.4	Trier les clés, supprimer les doublons . . . . .	21
5.5	Générer / calculer des nouvelles champs . . . . .	23
5.6	Appliquer une fonction d'agrégation aux mesures . . . . .	26
5.7	Ajouter les dimensions à évolution lente pour gérer les mises à jour . . . . .	27
5.8	Ordonnancer le flux . . . . .	32
5.9	Lancer la transformation . . . . .	33
<b>6</b>	<b>Représentation multidimensionnelle avec Schema Workbench</b>	<b>35</b>
6.1	Création de la connexion . . . . .	35
6.2	OLAP Cube . . . . .	35
6.2.1	Dimensions . . . . .	36
6.2.2	Mesures . . . . .	37
<b>7</b>	<b>Déploiement de la solution sur Pentaho server</b>	<b>38</b>
7.1	Création de la connexion sur Pentaho Server . . . . .	38
7.2	Visualisation sur Pentaho Server . . . . .	39

7.2.1	Méthode typique pour créer un graphique sur un tableau de bord CDE : . . . .	39
7.2.2	Application 1 : Graphique à secteurs (Pie chart) sur un tableau de board CDE : . . . .	40
7.2.3	Application 2 : Répartition géographique sur une carte CDE : . . . . .	41
7.2.4	Application 3 : Évolution du chiffre d'affaires en fonction des coûts . . . . .	46
7.2.5	Application 4 : Chiffre d'affaires, coûts d'exploitation, et superficie par province . . . . .	47
<b>8</b>	<b>Automatisation des tâches avec Jenkins</b>	<b>48</b>
8.1	Configuration Jenkins . . . . .	48
8.2	Création d'une tâche (job) régulière . . . . .	48
8.3	Épreuve de la build . . . . .	51
<b>A</b>	<b>Annexes</b>	<b>52</b>
A.1	Extraction de latitude et longitude à l'aide d'API de géocodage externes, et de PDI . . . . .	52
A.1.1	Conception de l'algorithme . . . . .	53
A.1.2	Réalisation de la solution . . . . .	53
A.1.3	Automatisation de la tâche . . . . .	56

## Table des figures

1	Extrait du fichier Excel contenant les données en question . . . . .	6
2	Dimensions de l'entrepôt de données . . . . .	7
3	Hierarchie de dimensions . . . . .	7
4	Table de faits . . . . .	8
5	Schéma en étoile . . . . .	9
6	Pentaho Data Integration (pdi) . . . . .	10
7	Assistant d'installation JDK 8.0u241 . . . . .	11
8	HeidiSQL : Interface d'identification et de gestion de session . . . . .	12
9	Assistant d'installation Jenkins . . . . .	15
10	HeidiSQL : Création d'une nouvelle base de données . . . . .	16
11	Page de téléchargement du pilote MySQL . . . . .	17
12	Configuration de la Connexion à la base de données . . . . .	18
13	Étape <b>input_Qualified.xls</b> . . . . .	20
14	Étape <b>Select Values</b> . . . . .	20
15	Étape <b>select Id, province, commune, type_commune</b> . . . . .	21
16	Étape <b>sort commune, province</b> . . . . .	22
17	Étape <b>Sort rows</b> . . . . .	23
18	Création des deux champs Latitude, et Longitude . . . . .	24
19	Test du script de l'étape <b>extract month, quarter, semester, year</b> . . . . .	25
20	Création de la constante <b>all_projects</b> avec la valeur <b>all</b> . . . . .	25
21	Étape <b>Group by</b> . . . . .	27
22	Étape <b>dim date</b> . . . . .	28
23	Étape <b>dim reg</b> . . . . .	30
24	Étape <b>dim projet</b> . . . . .	31
25	Étape <b>Block until "dim reg" finish</b> . . . . .	32
26	Résultat de l'exécution de la transformation créée sur PDI . . . . .	33
27	Table de la dimension Date . . . . .	33
28	Table de la dimension Projet . . . . .	33
29	Table de la dimension Operation . . . . .	34
30	Table de la dimension Region . . . . .	34
31	Table de faits . . . . .	34
32	Configuration de la Connexion à la base de données . . . . .	35
33	OLAP Cube . . . . .	36
34	Création d'une nouvelle connexion sur Pentaho Server . . . . .	38
35	Importation du schéma <b>omrane.xml</b> . . . . .	39
36	Application d'un modèle dans le tableau de bord . . . . .	40
37	Répartition du chiffre d'affaire par type de commune . . . . .	41
38	Carte de la répartition géographique des coûts, des CA, et des superficies par province . . . . .	45
39	Exemple : Répartition des coûts, des CA, et des superficies dans la province de BOULEMANE . . . . .	46
40	Évolution du chiffre d'affaires en fonction des coûts pour la période de 1901 à 2012 . . . . .	47
41	CA, coûts, et superficies par province . . . . .	47

42	CA, coûts, et superficies par province . . . . .	48
43	Étape <b>REST client</b> . . . . .	55
44	Étape <b>JSON input</b> . . . . .	55

# 1 PROBLÉMATIQUE

Une société de construction voulait déployer un entrepôt de données pour avoir la capacité de suivre ses travaux, sachant que toutes les informations sont stockées sous forme d'un fichier Excel dont le schéma est indiqué dans la figure suivante :

Id	nature_projet	nature_operation	province	commune	type_commune	cout_operation	Chiffre_D_affaires	superficie	date_debut
2977	Amenagement Foncier	Lotissement Habitat	FES - ZOUAGHA - MOULAY YACOUB	ZOUAGHA	Urbaine	1.300429e+01	16.374142	5.97040e+00	1996-05-01
2978	Amenagement Foncier	Lotissement Habitat	SEFROU	SEFROU	Urbaine	5.060994e+00	0.304000	2.38000e+00	1995-08-01
2979	Amenagement Foncier	Lotissement Habitat	FES - MEDINA	FES - MEDINA	Urbaine	5.429669e+00	34.108025	8.00000e-01	1996-11-15
2980	Amenagement Foncier	Lotissement Habitat	FES - MEDINA	FES - MEDINA	Urbaine	2.352609e+01	50.450850	3.91100e+00	2010-10-05
2981	Amenagement Foncier	Lotissement Habitat	FES - MEDINA	FES - MEDINA	Urbaine	2.951934e-01	0.209117	3.40000e-01	1996-07-17
2982	Amenagement Foncier	Lotissement Habitat	FES - MEDINA	FES - MEDINA	Urbaine	1.846407e+01	4.269400	4.08000e+00	2006-09-20
2983	Amenagement Foncier	Lotissement Habitat	TAZA	TAZA AL OULYA	Urbaine	1.247453e+02	35.099347	2.91067e+01	2001-12-25
2984	Amenagement Foncier	Lotissement Habitat	SEFROU	SEFROU	Urbaine	1.373695e+01	43.774627	2.42340e+00	2006-03-03
2985	Mise a Niveau Urbaine	Restruturation QHNR	SEFROU	SEFROU	Urbaine	2.861486e+00	1.060000	8.10000e+00	1994-07-01
2986	Mise a Niveau Urbaine	Restruturation QHNR	SEFROU	SEFROU	Urbaine	5.463026e+00	1.310000	9.40000e+00	1995-12-01
2987	Construction	Logements	TAZA	TAZA AL JADIDA	Urbaine	4.201269e+01	65.119967	4.25000e-01	1995-12-26
2988	Construction	Logements	FES - ZOUAGHA - MOULAY YACOUB	ZOUAGHA	Urbaine	2.224247e+01	29.513510	5.84800e-01	1995-08-25
2989	Mise a Niveau Urbaine	Restruturation QHNR	FES - MEDINA	FES - MEDINA	Urbaine	1.925527e+00	0.000000	1.06200e+00	2003-10-01

FIGURE 1 – Extrait du fichier Excel contenant les données en question

La direction a besoin d'un système décisionnel pour analyser quelques mesures pertinentes, par exemple répondre aux questions suivantes :

1. Le chiffre d'affaire par projet, région et type de région.
2. La superficie par projet, région et semestre.

## 2 CONCEPTION DE L'ENTROPÔT DE DONNÉES OMRANE

- **Sujet** : Gestion des projets.

- **Dimensions** :

Une dimension est un ensemble de données structurées correspondant à un axe d'analyse sur un événement mesurable. Dans ce contexte, les événements sont appelés «*faits*». Les dimensions catégorisent et décrivent les faits et les mesures de l'entrepôt de données pour permettre des réponses significatives aux questions économiques.

Notre entrepôt de données peut être construit à l'aide du modèle de données dimensionnel composés de dimensions suivants : DATE, OPERATION, REGION, et PROJET.

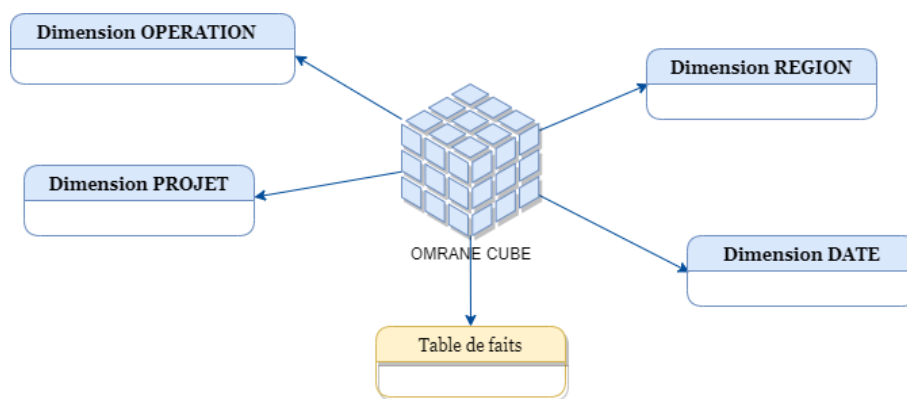


FIGURE 2 – Dimensions de l'entrepôt de données

- **Hierarchie** :

Une hiérarchie de dimension est une structure logique utilisant des niveaux triés pour organiser et agréger des données. (Voir Fig. 3)

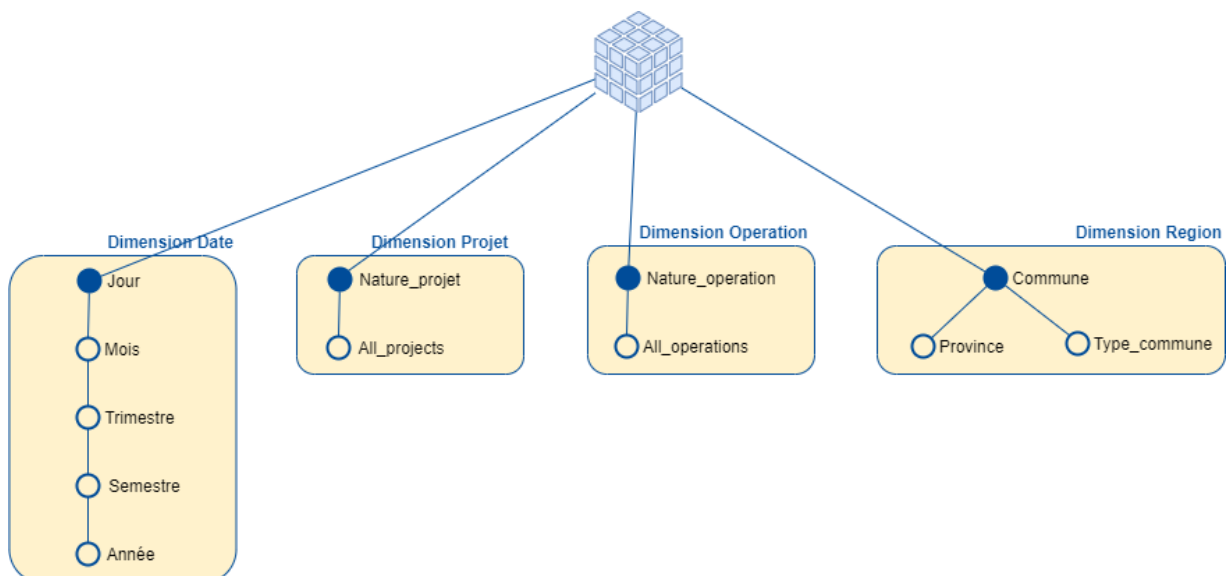


FIGURE 3 – Hiérarchie de dimensions

- **Table de faits :**

Une table de fait est une table qui contient les données observables (les faits) que l'on possède sur un sujet et que l'on veut étudier, selon divers axes d'analyse (les dimensions). Une table de faits contient les valeurs numériques de ce qu'on désire mesurer, ainsi les clés associées aux dimensions. "Il s'agit des clés étrangères dans la table de faits" [1]. (Voir Fig. 4)

Table de faits	
FK1	date_debut:Date
FK2	id_projet:Integer
FK3	nature_operation:String
FK4	commune:String
chiffre_d_affaires:Double	
cout_operation:Double	
superficie:Double	

FIGURE 4 – Table de faits

- **Modélisation du data warehouse :**

Il existe principalement deux types de base de modèle dimensionnel :

- **Le schéma en étoile** : Dans un schéma en étoile, une table centrale de faits contenant les faits à analyser référence les tables de dimensions par des clefs étrangères. Chaque dimension est décrite par une seule table (feuille de l'arbre de tables) dont les attributs représentent les diverses granularités possibles. (Voir Fig. 5)
- **Le schéma en flocons de neige** : Dans un schéma en flocon, cette même table de faits, référence les tables de dimensions de premier niveau, au même titre que le schéma en étoile. La différence réside dans le fait que les dimensions sont décrites par une succession de tables (à l'aide de clefs étrangères) représentant la granularité de l'information. Ce schéma évite les redondances d'information mais nécessite des jointures lors des agrégats de ces dimensions.



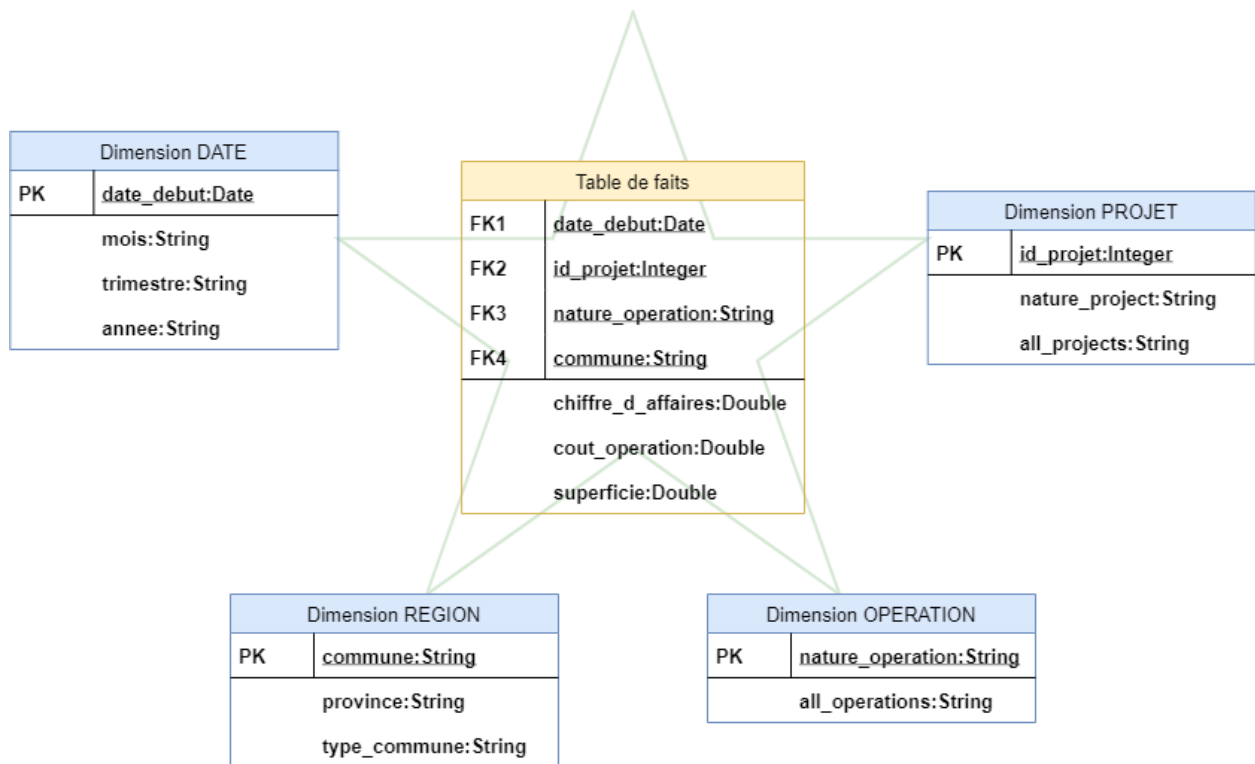


FIGURE 5 – Schéma en étoile

### 3 PRÉPARATION DE L'ENVIRONNEMENT

Pour pouvoir déployer notre solution d'entrepôt de données, nous aurons besoin de 4 outils :

- **Pentaho Data Integration (pdi)** pour la manipulation de données et la création de dimensions et la table de faits.
- **HeidiSQL** pour la création de la base de données cible.
- **Schema Workbench (psw)** pour l'analyse de données OLAP.
- **Pentaho Server** pour le déploiement et la publication de rapports personnalisés.
- **Jenkins** pour l'automatisation et l'exécution des transformations.

#### 3.1 PENTAHO DATA INTEGRATION

Pentaho Data Integration (PDI), longtemps connu sous le nom de Kettle, est un ETL open source qui permet de concevoir et d'exécuter des opérations de manipulation et de transformation de données.

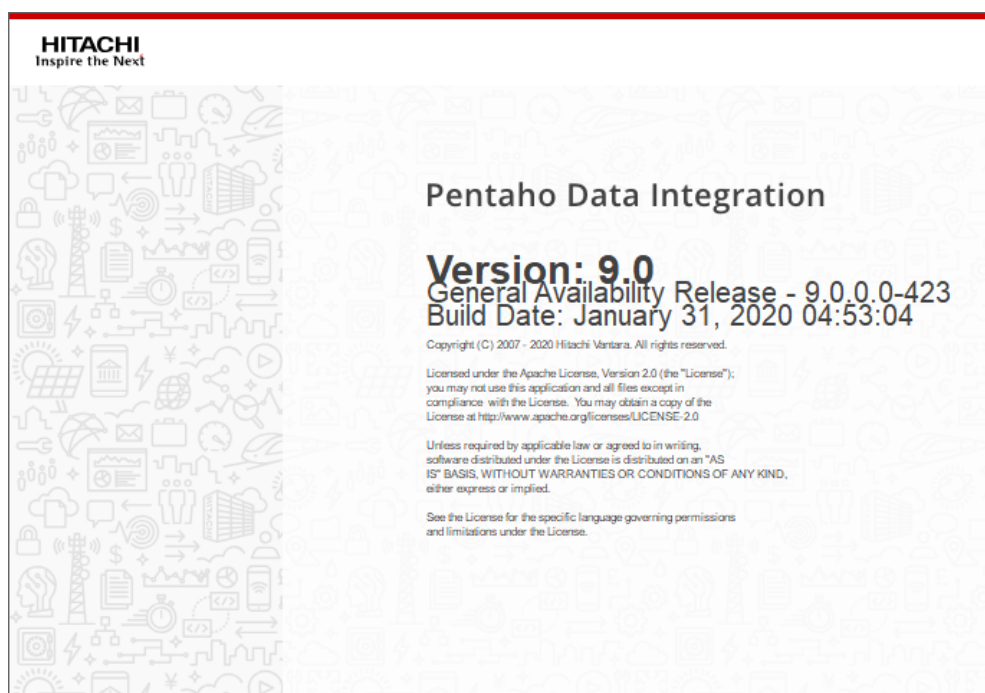


FIGURE 6 – Pentaho Data Integration (pdi)

##### 3.1.1 FICHE TECHNIQUE :

- Version étudiée : 9.0.0.0-423
- Date de réalisation : 31/01/2020
- Distribuée par : Hitachi Vantara
- Site web : <https://www.hitachivantara.com/en-us/products/data-management-analytics/pentaho-platform/pentaho-data-integration.html>
- Licence : GNU Library or Lesser General Public License version 2.0 (LGPLv2), GNU General Public License version 2.0 (GPLv2), Mozilla Public License 1.1 (MPL 1.1)

### 3.1.2 ÉTAPES D'INSTALLATION :

1. Pentaho Data Integration se décline en deux variétés :
  - Community Edition (CE) - Version gratuite pour les développeurs.  
Vous pouvez la télécharger depuis [Sourceforge.net](https://sourceforge.net).
  - Enterprise Edition (EE) - Version payante pour une utilisation en entreprise.  
Vous pouvez télécharger une [version d'évaluation](#) de 30 jours.
2. **Conditions préalables** : PDI 9.0 nécessite l'installation d'un Java Development Kit (JDK) version 8.0 ou supérieure (la version choisie ici est 8.0u241)

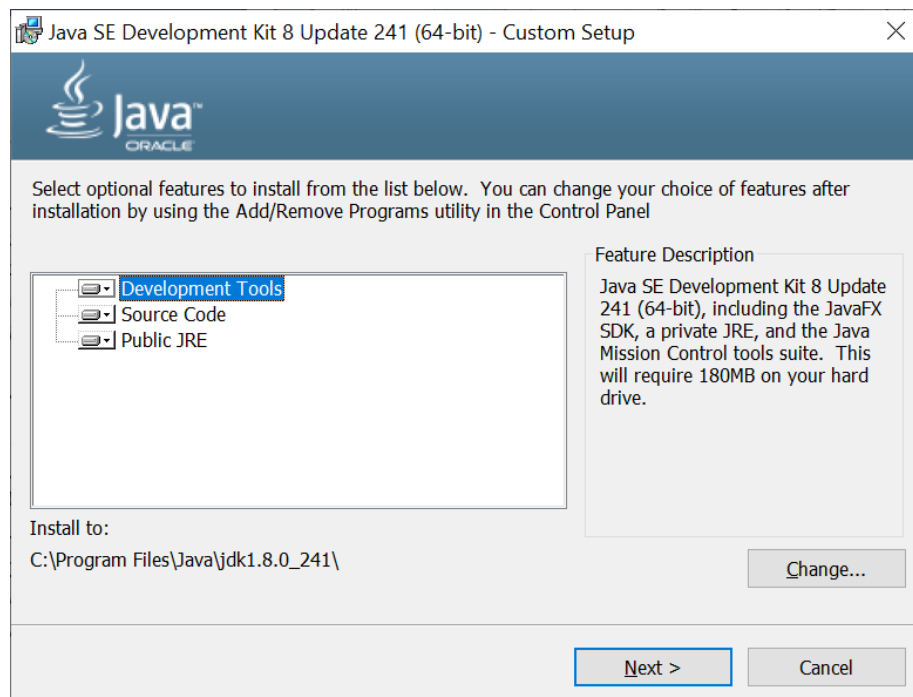


FIGURE 7 – Assistant d'installation JDK 8.0u241

3. **PDI ne nécessite pas d'installation.** Décompressez simplement le fichier zip dans un dossier de votre choix.
4. **Exécution** : PDI est livré avec une interface graphique appelée Spoon, en plus des scripts de ligne de commande (Kitchen, Pan) pour exécuter des transformations et des travaux, ainsi que d'autres utilitaires.

## 3.2 HEIDISQL

**HeidiSQL** est un outil d'administration gratuit et open-source pour *MySQL*, *Microsoft SQL Server*, *PostgreSQL* et *SQLite*.

Il permet de parcourir et de modifier des données, de créer et de modifier des tables, des vues, des procédures, des déclencheurs et des événements planifiés.

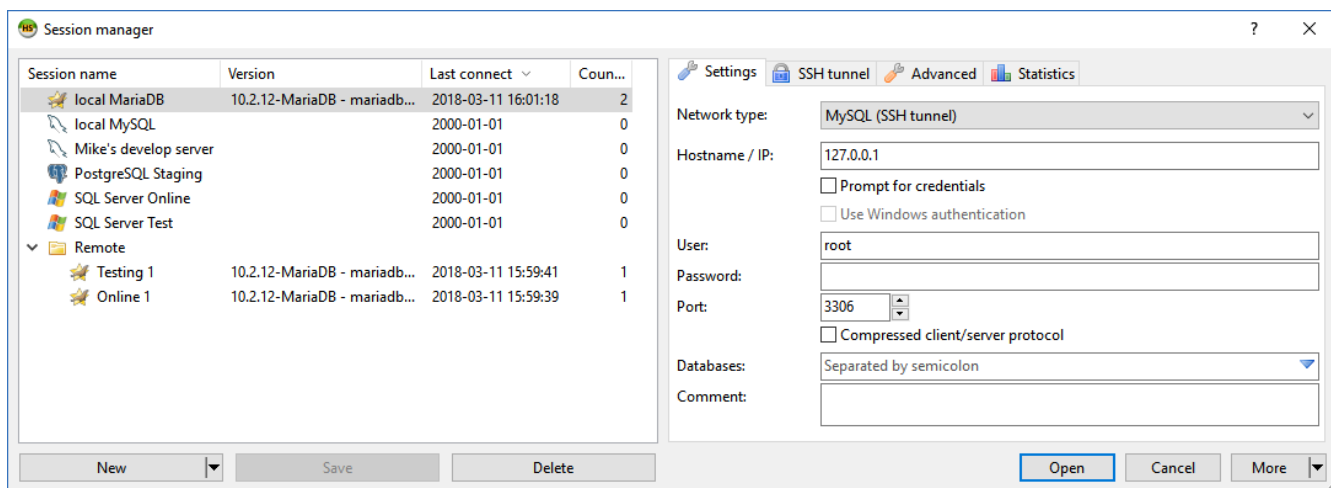


FIGURE 8 – HeidiSQL : Interface d'identification et de gestion de session

### 3.2.1 FICHE TECHNIQUE :

- Version stable : 11.0
- Date de réalisation : 2020-03-17
- Écrit en : Delphi
- Site web : <https://www.heidisql.com/>
- Github repository : <https://github.com/HeidiSQL/HeidiSQL>
- Licence : GPL (GNU GENERAL PUBLIC LICENSE)

### 3.2.2 ÉTAPES D'INSTALLATION :

1. HeidiSQL est disponible de différentes manières :
  - Programme d'installation, [32/64 bits](#)
  - Application Microsoft Store, [32 bits](#)
  - Version portable (zippée) : [32 bits](#), [64 bits](#)

Dans notre cas, nous allons utiliser la version portable 64 bits
2. **Compatibilité du système d'exploitation** : fonctionne correctement sur Windows 7, 8 et 10. (L'exécution de HeidiSQL sur Wine est actuellement assez instable)
3. **HeidiSQL ne nécessite pas d'installation**. Décompressez simplement le fichier zip dans un dossier de votre choix.
4. **Exécution** : Lancer directement le fichier `heidisql.exe` pour l'ouvrir.

## 3.3 SCHEMA WORKBENCH

Le **Schema workbench**, du projet **Pentaho** d'analyse de données OLAP «*Mondrian*», est une interface graphique permettant de produire des schemas Mondrian au format XML et de les publier dans une solution Pentaho.

#### Remarque :

Le Schema Workbench n'est pas indispensable car on peut aussi écrire le XML manuellement, mais il est bien pratique.

### 3.3.1 FICHE TECHNIQUE :

- Version étudiée : 3.14.0.0-12
- Date de réalisation : 22/05/2017
- Site web : <https://community.hitachivantara.com/s/article/mondrian>
- Licence : Eclipse Public License (EPL)

### 3.3.2 ÉTAPES D'INSTALLATION :

1. Le logiciel Pentaho Schema Workbench (PSW) est disponible pour le téléchargement depuis [Sourceforge.net](https://sourceforge.net).
2. **Conditions préalables** : psw 3.14 nécessite l'installation d'un Java Development Kit (JDK) version 1.5 ou supérieur (la version choisie ici est 8.0u241)
3. **PSW ne nécessite pas d'installation**. Décompressez simplement le fichier zip dans un dossier de votre choix.
4. **Exécution** : Lancer directement le fichier `workbench.bat` pour l'ouvrir.

## 3.4 PENTAHO SERVER

Pentaho Server est un serveur d'analyse et de génération de rapports. Il permet aux utilisateurs d'accéder à des rapports riches et interactifs, explorer les données visuelles et découvrir rapidement les patterns pour prendre les meilleures décisions plus rapidement.

### 3.4.1 FICHE TECHNIQUE :

- Version étudiée : 9.0.0.0-423
- Date de réalisation : 31/01/2020
- Distribuée par : Hitachi Vantara
- Site web : <https://www.pentaho.com>
- Licence : Pentaho Community Edition (CE) : Apache license version 2.0;

### 3.4.2 ÉTAPES D'INSTALLATION :

1. Le logiciel **Pentaho Server** est disponible pour le téléchargement depuis [Sourceforge.net](https://sourceforge.net).
2. **Pentaho Server ne nécessite pas d'installation**. Décompressez simplement le fichier zip dans un dossier de votre choix.  
Cette méthode installe le Pentaho Server, les plugins BA (tels que Analyzer et Interactive Reporting) et les plugins DI (tels que Big Data et Marketplace).
3. **Exécution** : Lancer directement le fichier `start-pentaho.bat` pour l'ouvrir.

4. Depuis un poste de travail, ouvrez un navigateur Web et entrez cette URL :  
<http://localhost:8080/pentaho> pour accéder à la console utilisateur Pentaho (PUC).


## 3.5 JENKINS

Jenkins est un serveur d'automatisation gratuit et open source. Il permet d'automatiser les tranches du développement logiciel liées à la création, aux tests et au déploiement.

### 3.5.1 FICHE TECHNIQUE :

- Version étudiée : 2.222.3
- Date de réalisation : 24/04/2020
- Écrit en : Java
- Site web : <https://www.jenkins.io/>
- Licence : MIT License

### 3.5.2 ÉTAPES D'INSTALLATION :

1. **Configuration matérielle minimale** : 256 Mo de RAM / 1 Go d'espace disque [2].
2. **Logiciels requis** :
  - **Java** :
    - Environnements d'exécution Java 8, les versions 32 bits et 64 bits sont prises en charge.
    - Depuis Jenkins 2.164 et 2.164.1, les environnements d'exécution Java 11 sont pris en charge.
    - Les versions antérieures de Java, Java 9, Java 10 et Java 12 ne sont pas prises en charge [3].
  - **Navigateur Web** : voir la page de [compatibilité du navigateur Web](#)
3. Accédez à <https://www.jenkins.io/download/> et sélectionnez la plate-forme. Dans notre cas, Windows.
4. Décompressez le package téléchargé. Double-cliquez sur jenkins.msi décompressé. Vous pouvez également utiliser un WAR (application Web ARchive) mais ce n'est pas recommandé.
5. Dans l'assistant d'installation Jenkins, cliquez sur .

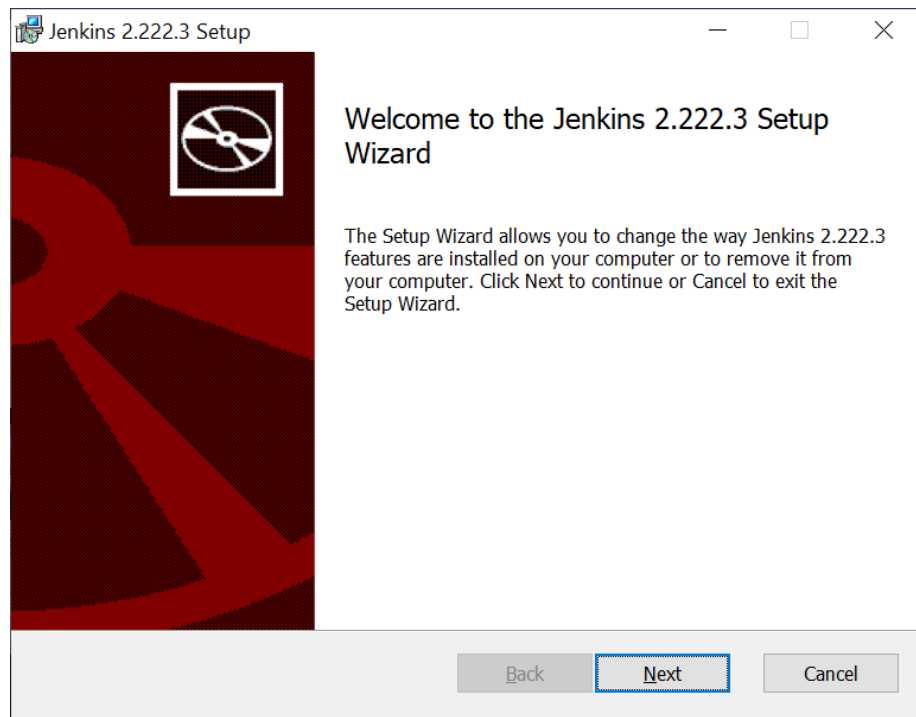


FIGURE 9 – Assistant d'installation Jenkins

6. Choisissez l'emplacement où vous souhaitez installer l'instance Jenkins (l'emplacement par défaut est `C:\ProgramFiles(x86)\Jenkins`), puis cliquez sur le bouton **Next**.
7. Cliquez sur le bouton **Install**.
8. Une fois l'installation terminée, cliquez sur **Finish**.

## 4 CRÉATION DE LA BASE DE DONNÉES CIBLE

1. Allez dans le logiciel **HeidiSQL**, et remplissez les champs d'identification :
  - **Network type** : MariaDB or MySQL (TCP/IP)
  - **Hostname/IP** : localhost
  - **User** : root
  - **Password** :
  - **Port** : 3306et cliquez sur **Open**
2. Dès que vous êtes connecté au serveur MySQL, vous pouvez créer une nouvelle base de données :
  - Allez sur l'onglet **Query**, et tapez la requête SQL permettant de créer une nouvelle base de donnée :

```
CREATE DATABASE 'omrane_data_marts'
```

- Sinon, avec le bouton droit de la souris, cliquez sur le «noeud d'arbre» tout en haut à gauche; Dans son menu contextuel, cliquez sur **Create New** → **Database**.

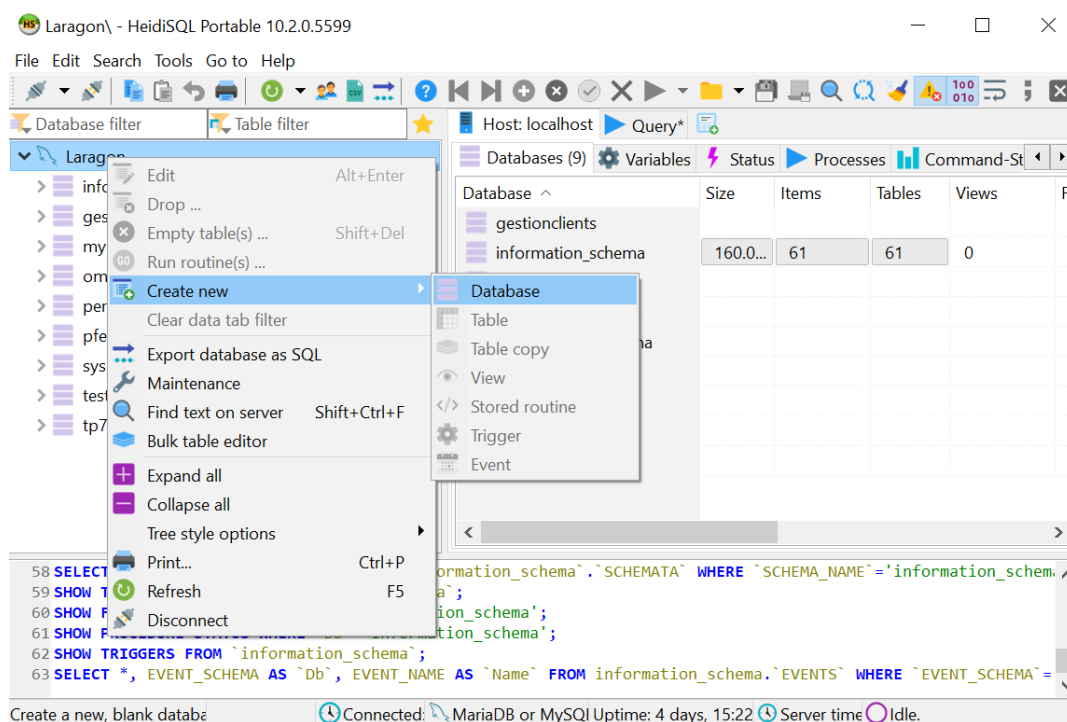


FIGURE 10 – HeidiSQL : Création d'une nouvelle base de données



## 5 MANIPULATION DES DONNÉES AVEC L'OUTIL PDI

### 5.1 CRÉER UNE CONNEXION JDBC DANS PDI

Avant d'établir la connexion avec la base de données nouvellement créée, nous devons installer le pilote MySQL pour JDBC :

1. Allez dans le logiciel PDI, dans le menu en haut cliquez sur **Tools** → **MarketPlace** et recherchez **PDI MySQL Plugin**
2. Installez-le (cela installera automatiquement le pilote manquant dans le répertoire : `data-integration/plugins/databases/pdi-mysql-plugin/lib`)
3. Redémarrez PDI  
[4]

Sinon, si vous voulez installer une version spécifique du connecteur, vous pouvez suivre cette méthode :

1. Allez sur <https://dev.mysql.com/downloads/connector/j/>

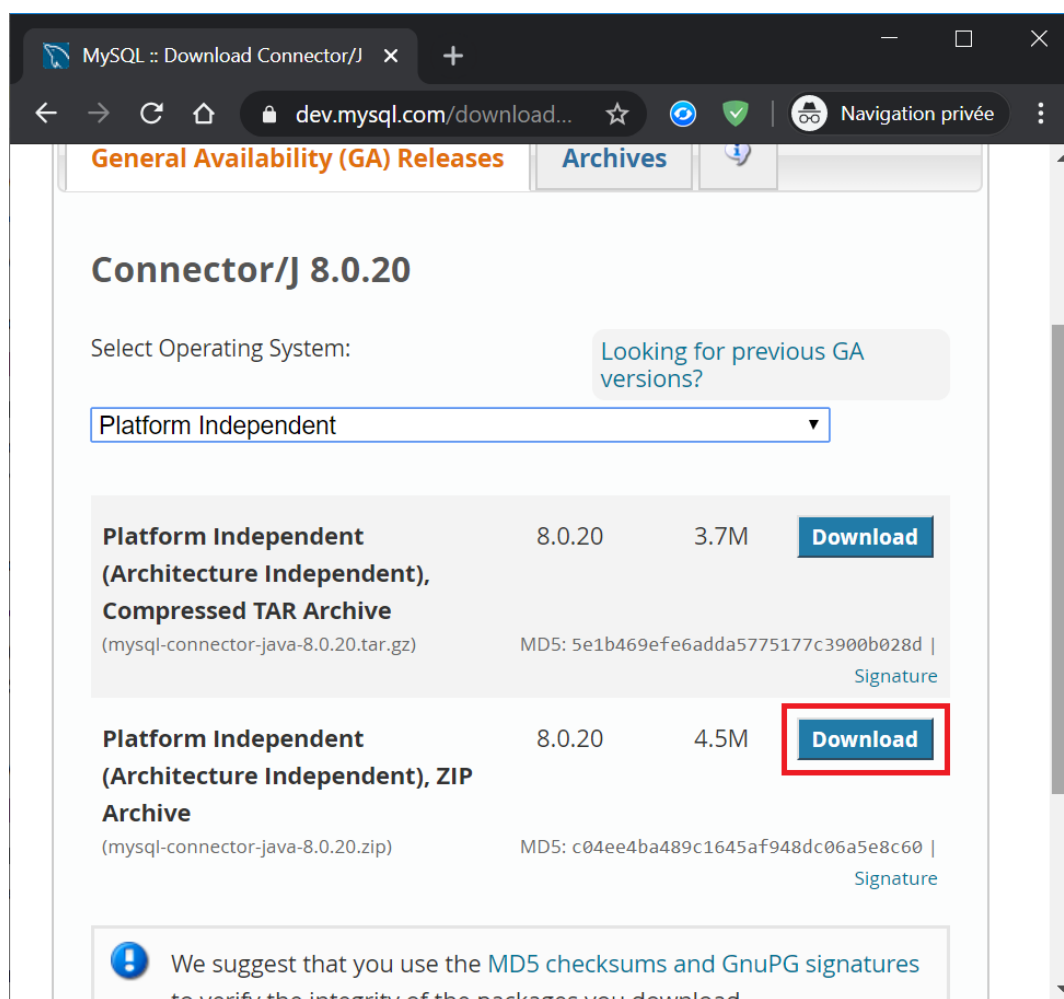


FIGURE 11 – Page de téléchargement du pilote MySQL

2. Choisissez **plateforme indépendante** dans la liste déroulante comme OS, et cliquez sur **Download**

3. Décompressez le fichier zip (dans notre cas c'est mysql-connector-java-8.0.20)
4. Copiez le fichier .jar (mysql-connector-java-8.0.20.jar) et collez-le dans le dossier `data-integration/lib`

Maintenant que le connecteur MySQL pour JDBC est installé. Allez dans PDI, cliquez sur **File** → **New** → **Database Connection...**, et remplissez la configuration comme indiqué dans la figure ci-dessous :

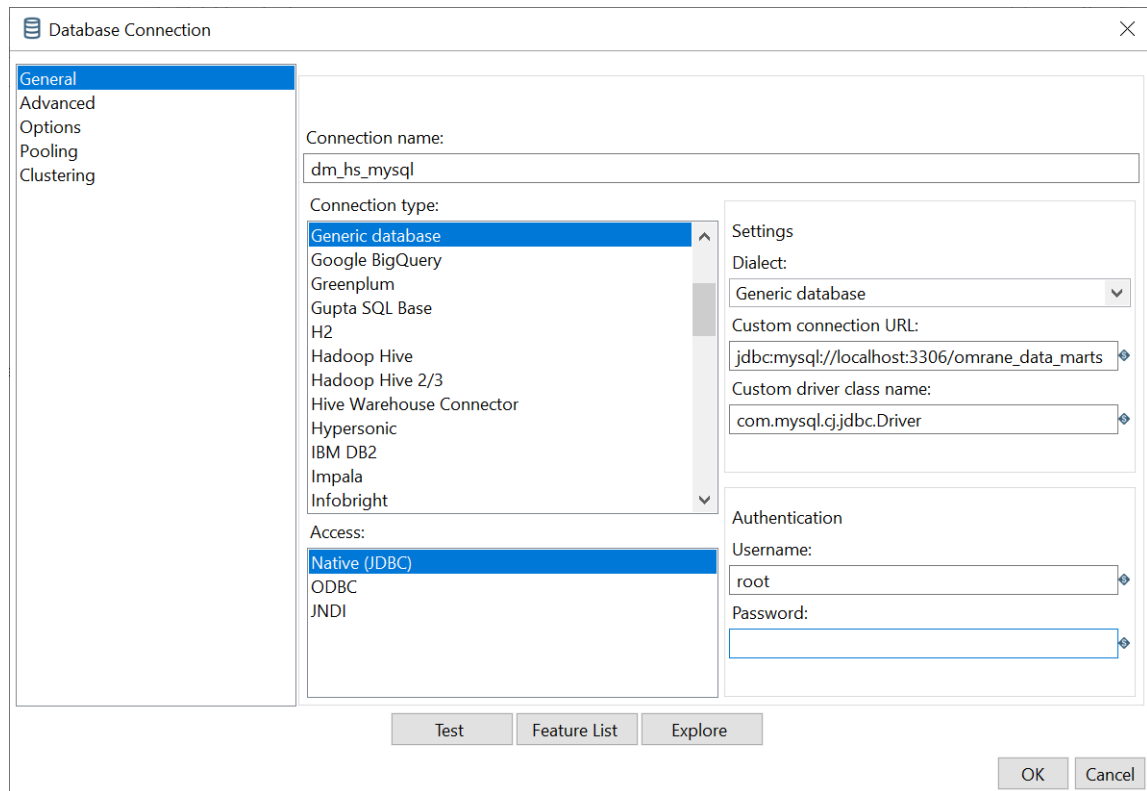
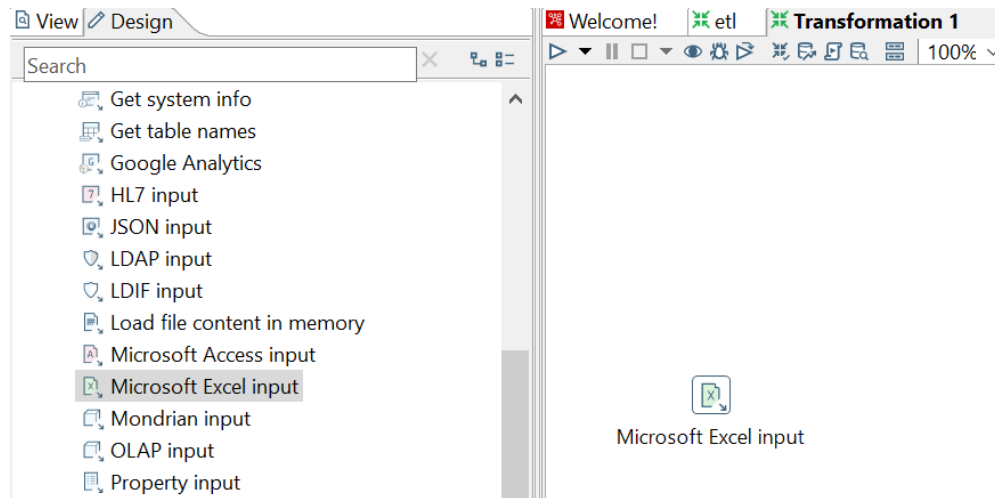


FIGURE 12 – Configuration de la Connexion à la base de données

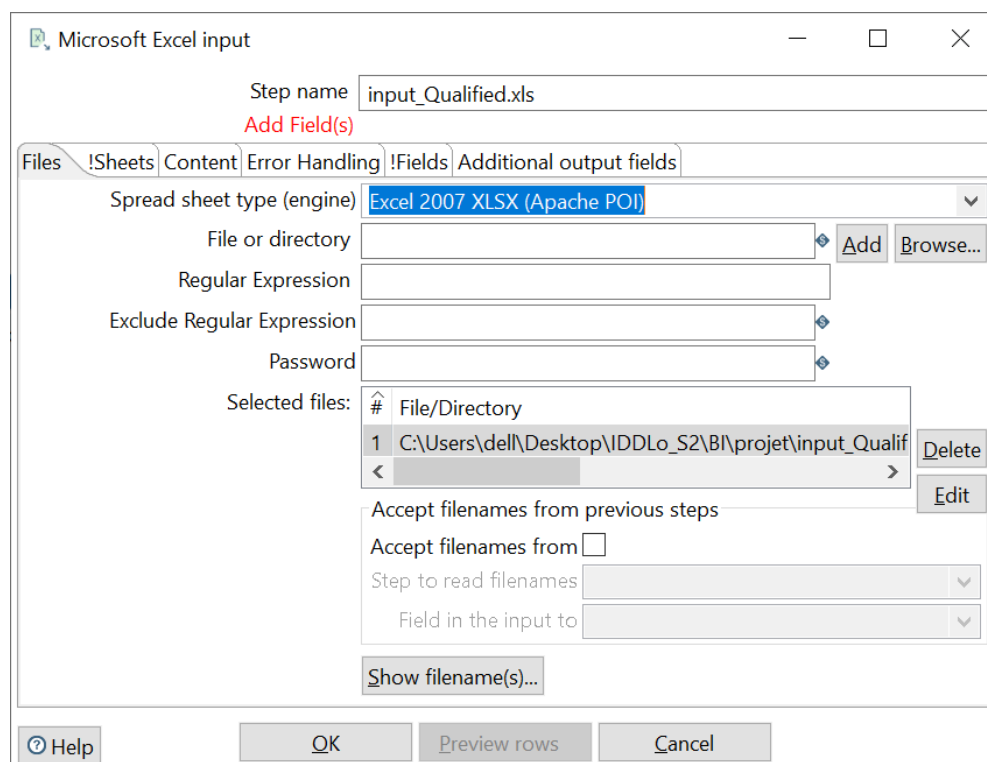
## 5.2 EXTRAIRE LES DONNÉES EXCEL

Vu qu'il n'y a qu'une seule source et que le volume de données traitées est assez faible, nous n'utiliserons pas de `staging tables` comme moyen de traitement temporaire, et nous satisferons par le processus ETL classique pour la création et l'alimentation des tables de dimensions ainsi que la table de faits.

1. Pour commencer, allez d'abord dans le logiciel **Pentaho Data Integration**, cliquez sur **File** → **New** → **Transformation** ou appuyez sur **ctrl** + **N**.
2. Cliquez sur l'onglet **design** → **Input**, puis glissez l'étape **Microsoft Excel Input** dans l'espace de travail.



3. Double-cliquez sur l'étape **Microsoft Excel Input**
  - (a) Dans l'onglet **!Files**, cliquez sur **Browse** et sélectionnez votre fichier Excel depuis l'explorateur de fichiers.
  - (b) Cliquez ensuite sur **Add**
  - (c) Dans la liste déroulante **Spread sheet type (engine)** choisissez le mode de compatibilité Excel en tant que **Excel 2007 XLSX (Apache POI)**



- (d) Allez maintenant sur l'onglet **!Fields** et cliquez sur **Get fields from header row...**
- (e) Finalement, veillez que les noms des champs ne contiennent pas d'espace blanc en début; milieu; fin, et cliquez sur **OK** quand vous aurez terminé.

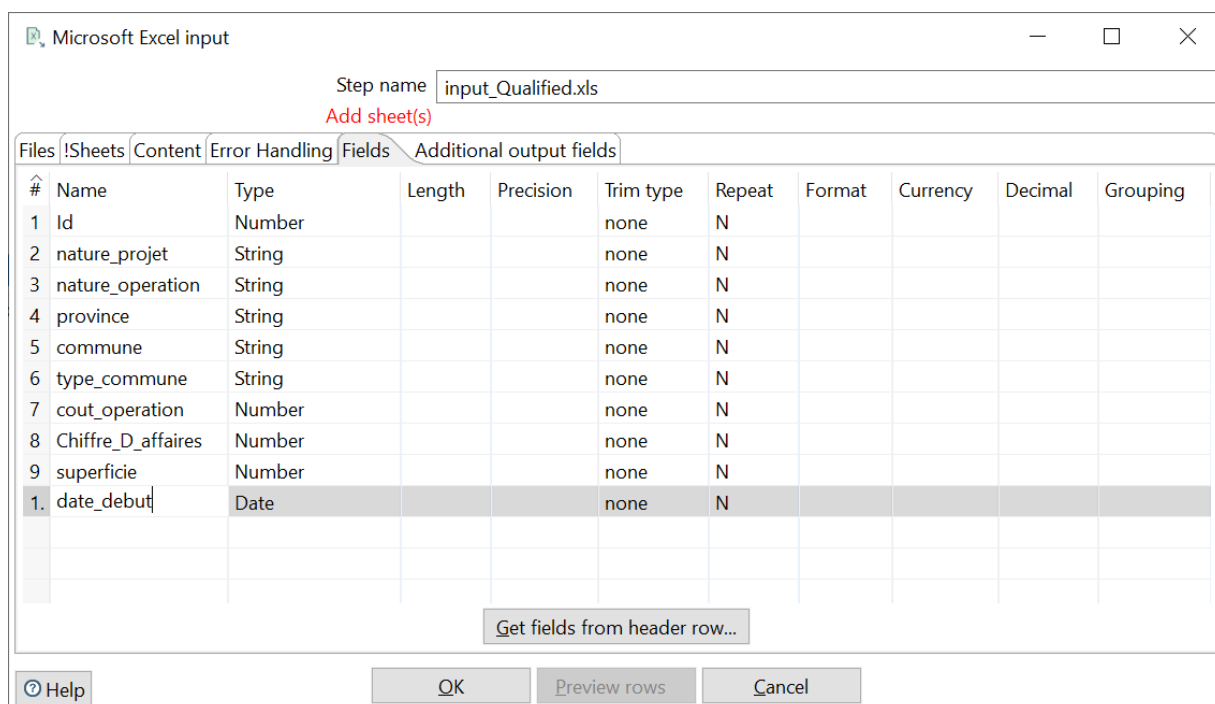


FIGURE 13 – Étape **input\_Qualified.xls**

### 5.3 SÉLECTIONNER LE FLUX DE DONNÉES

1. Cliquez sur l'onglet **design** → **Transform**, puis glissez l'étape **Select values** dans l'espace de travail.
2. Dupliquez cette étape 4 fois.

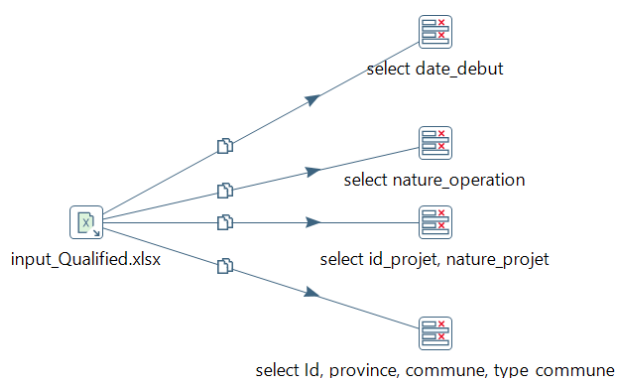


FIGURE 14 – Étape **Select Values**

3. Pour chacune de ces 4 étapes, sélectionnez uniquement le flux de données qui vous intéressent. (Exemple : Voir la figure ci-dessous)

Step name:

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	Id			
2	province			
3	commune			
4	type_commune			

Get fields to select  
Edit Mapping

Include unspecified fields, ordered by ☐

Help OK Cancel

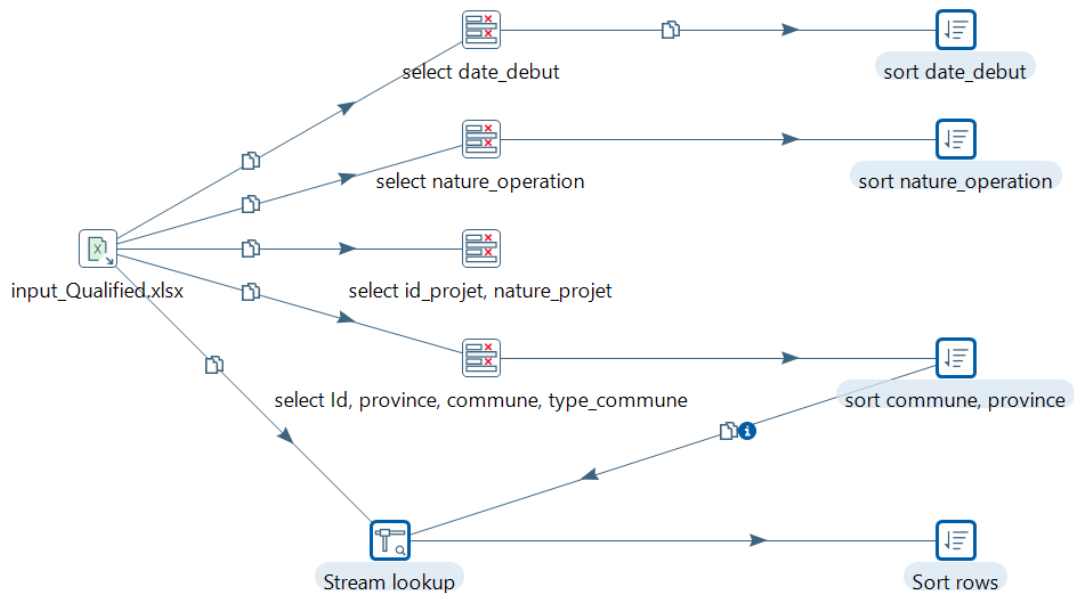
FIGURE 15 – Étape **select Id, province, commune, type\_commune**

## 5.4 TRIER LES CLÉS, SUPPRIMER LES DOUBLONS

1. Cliquez sur l'onglet **design** → **Transform**, puis glissez l'étape **Sort rows** dans l'espace de travail.
2. Dupliquez cette étape 3 fois (pour les 3 flux provenant de **select date\_debut**, **select nature\_operation**, **select Id, province, commune, type\_commune**.  
**Remarque** : Comme les valeurs du champ Id du fichier source sont déjà triées et uniques, nous n'utiliserons pas l'étape **Sort rows** pour le flux provenant de **Select Id, nature\_projet**, car elle est inutile dans ce cas.
3. Ajoutez l'étape **Stream lookup** pour pouvoir extraire et joindre le champ **id\_commune** dans le flux principal provenant de **input\_Qualified.xlsx**.

Les données provenant de l'étape Source (**sort commune, province**) sont d'abord lues en mémoire et sont ensuite utilisées pour rechercher les données du flux principal.

4. Ensuite, ajoutez l'étape **Sort rows** pour le flux provenant de **Stream lookup**



5. Pour chacune de ces 3 étapes (**sort commune, province**, **sort date\_debut**, **sort nature\_operation**) :
- Cliquez sur **Get Fields** puis enlevez ces champs et ne laissez que les champs déterminant la clé primaire.
  - Cochez la case **Only pass unique rows ?** pour supprimer les groupes de clés doublons.
- (Exemple : Voir la figure 16)

Sort rows

Step name:

Sort directory:

TMP-file prefix:

Sort size (rows in memory):

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☒

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	province	Y	N	N	0	N
2	commune	Y	N	N	0	N

FIGURE 16 – Étape **sort commune, province**

- Concernant le flux provenant de **Stream lookup**, sélectionnez les clés associé aux dimensions. (Voir la figure ci-dessous)

Step name:

Sort directory:

TMP-file prefix:

Sort size (rows in memory):

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields :

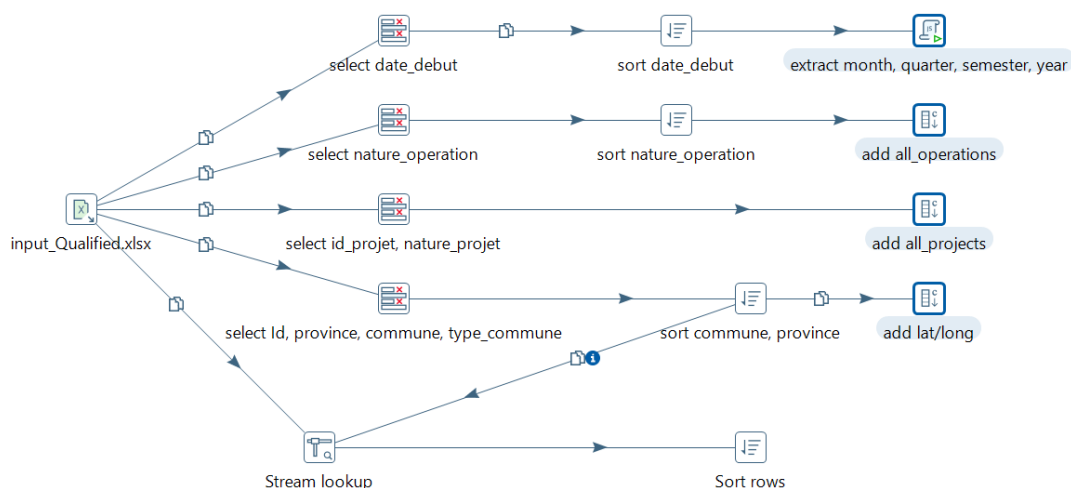
#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	Id	Y	N	N	0	N
2	nature_operation	Y	N	N	0	N
3	commune	Y	N	N	0	N
4	date_debut	Y	N	N	0	N

FIGURE 17 – Étape **Sort rows**

## 5.5 GÉNÉRER / CALCULER DES NOUVELLES CHAMPS

Dans cette phase, nous ajouterons des champs supplémentaires aux flux provenant de :

- **Sort date\_debut**
- **Sort nature\_operation**,
- **select Id, nature\_projet**,
- **Sort commune, province**.



- Ajouter deux champs Latitude, et Longitude au flux de données provenant de l'étape **sort commune, province** :
  - Cliquez sur l'onglet **design** → **Transform**, puis glissez l'étape **Add constants** dans l'espace de travail.

(b) Créer les deux champs Latitude, et Longitude avec la valeur *null*.

Step name:

Fields :

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	latitude	Number								N
2	longitude	Number								N

Buttons:

FIGURE 18 – Création des deux champs Latitude, et Longitude

(c) Cliquez sur  pour enregistrer.

2. Créer les niveaux mois, trimestre, semestre, et annee à partir du champ **date\_debut** :

(a) Cliquez sur l'onglet  → , puis glissez l'étape **Modified JavaScript value** dans l'espace de travail.

L'étape **Modified JavaScript value** fournit une interface utilisateur pour créer des expressions JavaScript que vous pouvez l'utiliser pour générer un flux de données additionnel.

(b) Utilisez le script ci-dessous pour créer ces quatre champs de mois, trimestre, semestre, et annee :

```
var mois = date2str(date_debut, "MM/YYYY");
var monthNumber = date_debut.getMonth();
var annee = date2str(date_debut, "YYYY");
var trimestre = Math.floor((monthNumber + 3) / 3) + "/" + annee;
var semestre = (monthNumber < 6) ? "1/" + annee : "2/" + annee;
```



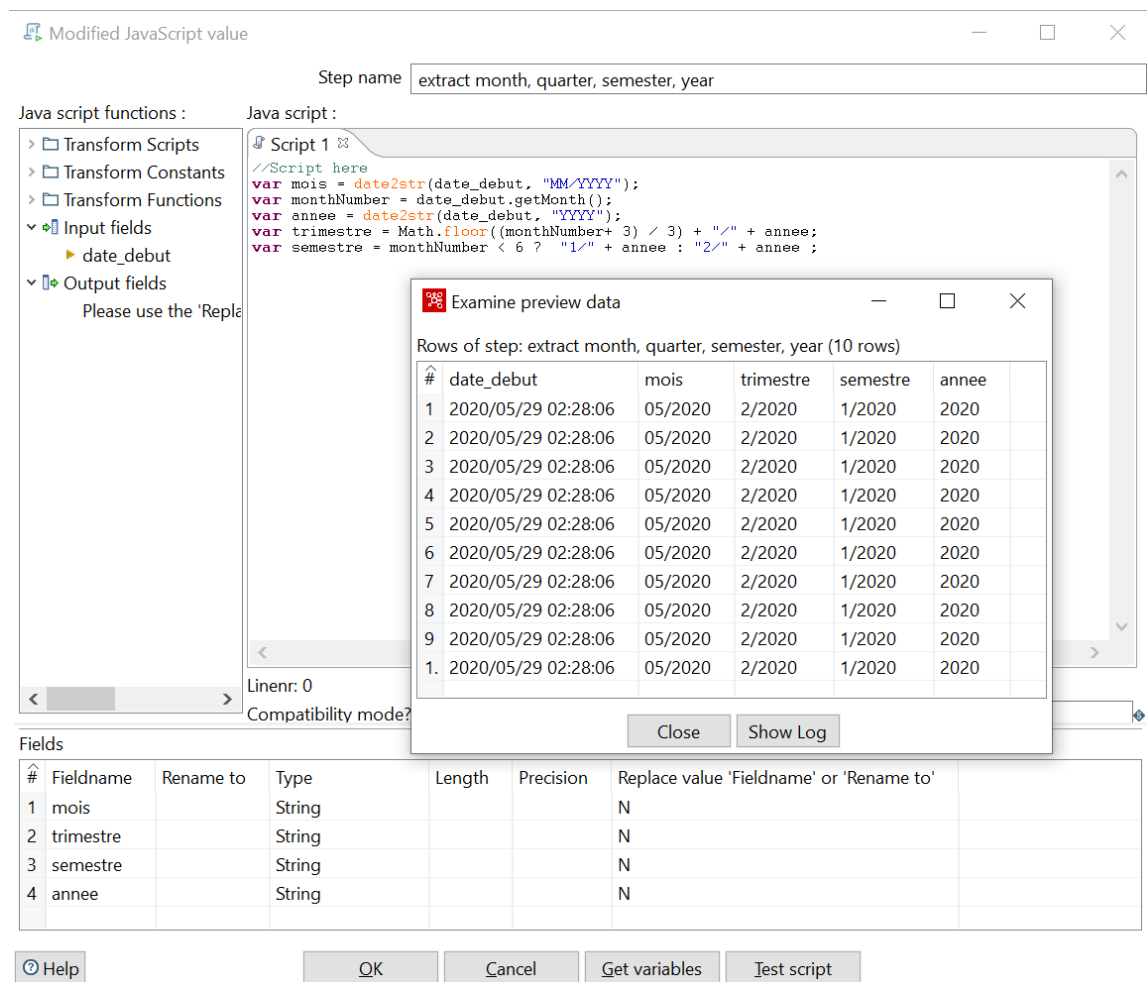


FIGURE 19 – Test du script de l'étape **extract month, quarter, semester, year**

3. Ajouter deux champs **all\_projects**, et **all\_operations** au flux de données provenant respectivement des étapes **select id\_projet**, **nature\_projet** et **sort nature\_operation**:
  - (a) Cliquez sur l'onglet **design** → **Transform**, puis glissez l'étape **Add constants** dans l'espace de travail.
  - (b) Dupliquez cette étape 2 fois (Pour les 2 flux de données provenant de **select id\_projet**, **nature\_projet** et **sort nature\_operation**)

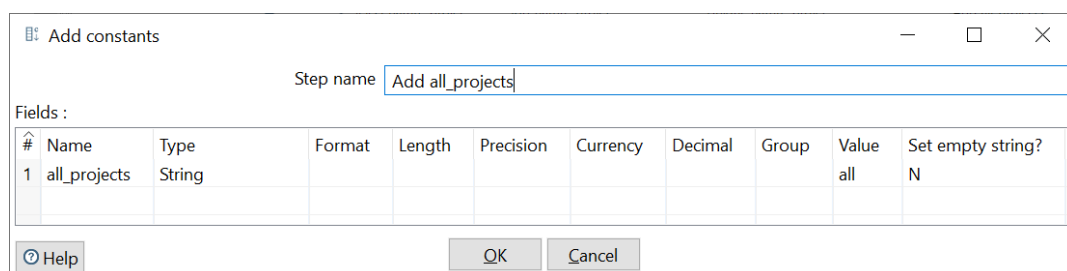
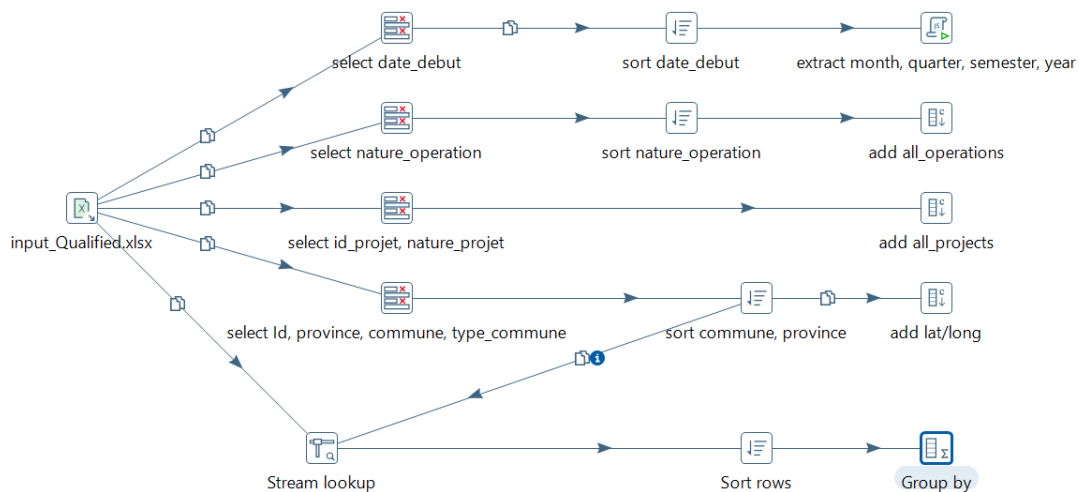


FIGURE 20 – Création de la constante **all\_projects** avec la valeur **all**

- (c) Cliquez sur **OK** pour enregistrer.

## 5.6 APPLIQUER UNE FONCTION D'AGRÉGATION AUX MESURES

1. Cliquez sur l'onglet **design** → **Statistics**, puis glissez l'étape **Group by** dans l'espace de travail.



2. Double-cliquez sur l'étape **Group by** et remplissez les champs qui composent le groupe, ainsi que les agrégats. Cette étape regroupera les clés en appliquant une fonction d'agrégation (COUNT, MAX, MIN, SUM, AVG, ...) sur les mesures de nos choix. (Voir Fig. 21)

Step name:

Include all rows? ☐

Temporary files directory:

TMP-file prefix:

Add line number, restart in each group ☐

Line number field name:

Always give back a result row ☐

The fields that make up the group:

#	Group field
1	Id
2	date_debut
3	nature_operation
4	id_commune

Aggregates :

#	Name	Subject	Type	Value
1	cout_operation	cout_operation	Sum	
2	Chiffre_D_affaires	Chiffre_D_affaires	Sum	
3	superficie	superficie	Sum	

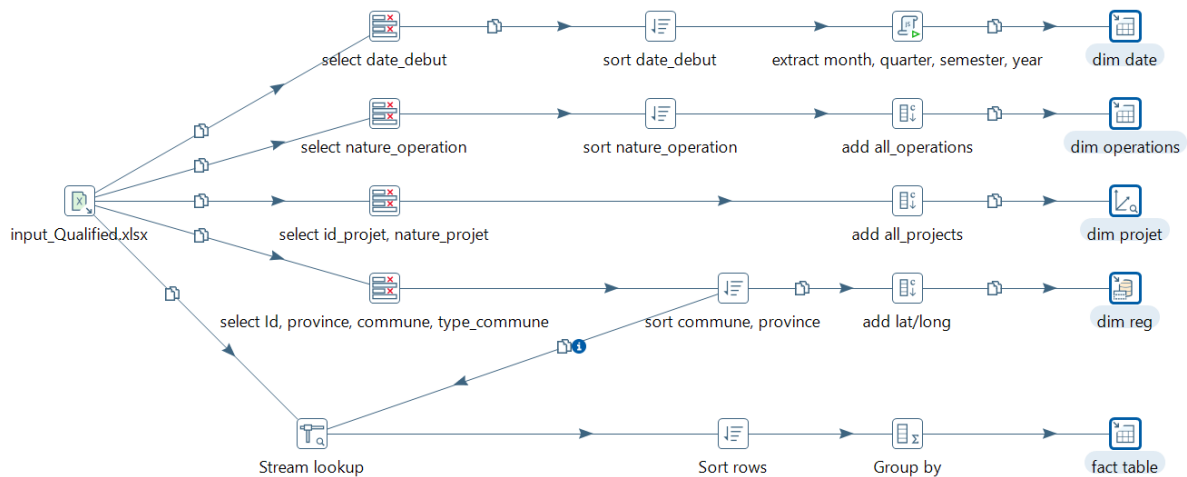
FIGURE 21 – Étape **Group by**

Notez que si le flux de données entrantes n'est pas trié sur les clés spécifiées, les résultats de sortie peuvent être incorrects.

## 5.7 AJOUTER LES DIMENSIONS À ÉVOLUTION LENTE POUR GÉRER LES MISES À JOUR

Au cours de cette phase, nous utiliserons deux stratégies d'historisation :

- **SCD 1** : Écraser l'ancienne valeur avec la nouvelle;
- **SCD 2** : Ajouter une ligne dans la table de dimension pour la nouvelle valeur;



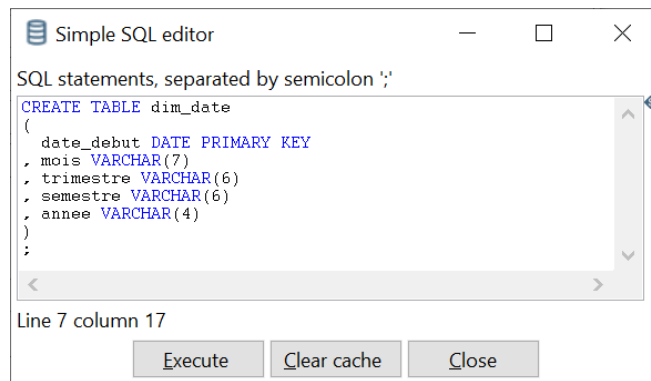
1. Mise en place de la première solution **SCD 1** :

- (a) Cliquez sur l'onglet **design** → **Output**, puis glissez l'étape **Table output** dans l'espace de travail.
- (b) Dupliquez cette étape 3 fois pour le flux de données provenant de **extract month, quarter, semester, year**, **add all\_operations**, et **Group by**.
- (c) Pour chacune de ces 3 **Table output** :
  - i. Entrez le nom de la connexion et celui de la table cible.
  - ii. Cochez **Truncate table** et **Specify database fields**
  - iii. Cliquez sur **Get fields** (Exemple : Voir la figure 22)

#	Table field	Stream field
1	date_debut	date_debut
2	mois	mois
3	annee	annee
4	trimestre	trimestre

FIGURE 22 – Étape **dim date**

- iv. Cliquez ensuite sur **SQL**, et vérifiez la syntaxe de la requête; vous pouvez également ajouter des clés sur la table (Voir la figure ci-dessous)



- v. Cliquez sur **Execute** pour lancer le script de création de la table dans la base de données cible.
- (d) Cliquez sur l'onglet **design** → **Output**, puis glissez l'étape **Insert / Update** dans l'espace de travail :
- Entrez le nom de la connexion et celui de la table cible.
  - Entrez les clés à rechercher les valeurs.
  - Cliquez sur **Get fields** et sélectionnez **Y** pour les champs à mettre à jour si la clé à rechercher est trouvée dans le flux. (Exemple : Voir la figure 23)

Step name: dim reg

Connection: mysql\_hs

Target schema:

Target table: dim\_reg

Commit size: 100

Don't perform any updates: ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2	Get fields
1	id_commune	=	Id		

Update fields:

#	Table field	Stream field	Update	Get update fields
1	id_commune	Id	N	
2	province	province	Y	
3	commune	commune	Y	
4	type_commune	type_commune	Y	
5	latitude	latitude	N	
6	longitude	longitude	N	

Buttons: Help, OK, Cancel, SQL

FIGURE 23 – Étape **dim reg**

- iv. Répétez la même procédure que celle décrite aux points 1(c)iv. et 1(c)v.
2. Mise en place de la deuxième solution **SCD 2** :
- Cliquez sur l'onglet **design** → **Data Warehouse**, puis glissez l'étape **Dimension lookup/update** dans l'espace de travail :
    - (a) Entrez le nom de la connexion et celui de la table cible.
    - (b) Entrez les clés à rechercher dans la dimension. (Voir la figure 24)
    - (c) Entrez le nom du "Technical key".
    - (d) Cliquez sur l'onglet **Fields** puis sur **Get fields** et sélectionnez pour chaque champ le type d'évolution à effectuer.

Dimension lookup/update

Step name:

Update the dimension? ☒

Connection:

Target schema:

Target table:

Commit size:

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all):

Keys Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	id_projet	id_projet

Technical key field:

Creation of technical key

☐ Use table maximum + 1

☐ Use sequence

☒ Use auto increment field

Version field:

Stream Datefield:

Date range start field:  Min. year:

Use an alternative start date? ☒

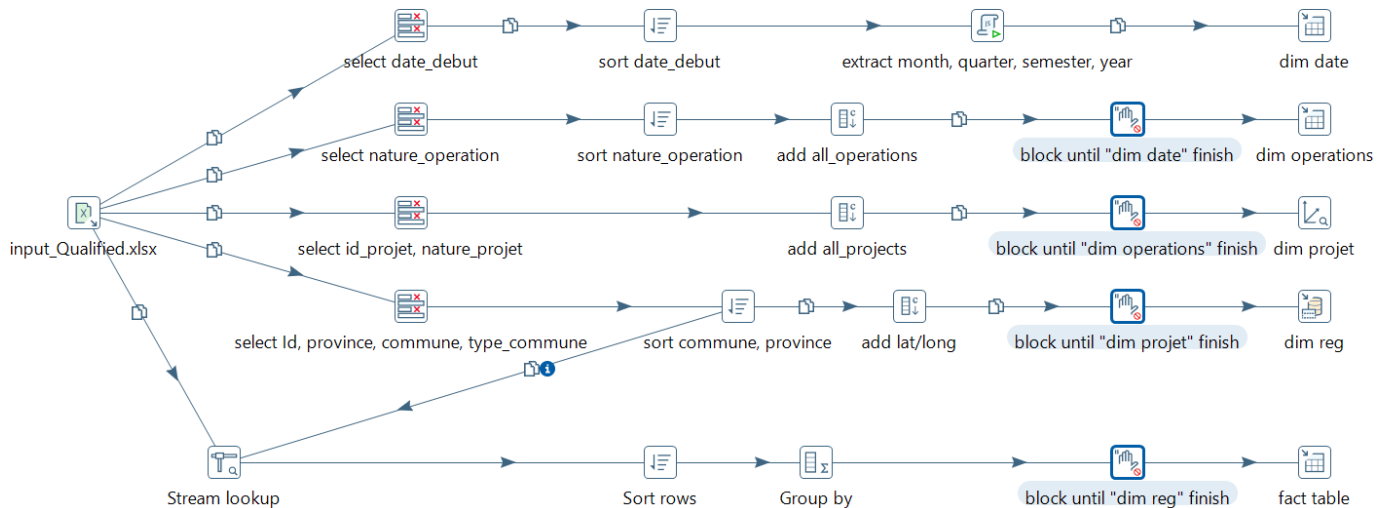
Table date range end:  Max. year:

FIGURE 24 – Étape **dim projet**

(e) Répétez la même procédure que celle décrite aux points 1(c)iv. et 1(c)v.

## 5.8 ORDONNANCER LE FLUX

1. Cliquez sur l'onglet **design** → **Flow**, puis glissez l'étape **Block this step until steps finish** dans l'espace de travail.
2. Dupliquez cette étape 4 fois.



3. Pour chacune de ces 4 étapes, ajoutez les étapes à surveiller (Exemple : Voir la figure ci-dessous)

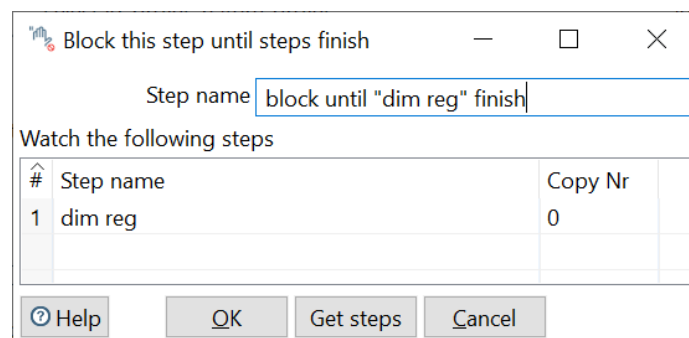


FIGURE 25 – Étape **Block until "dim reg" finish**



# 5.9 LANCER LA TRANSFORMATION

1. Cliquez sur **Exécuter** ou appuyez sur **F9** pour démarrer la transformation.

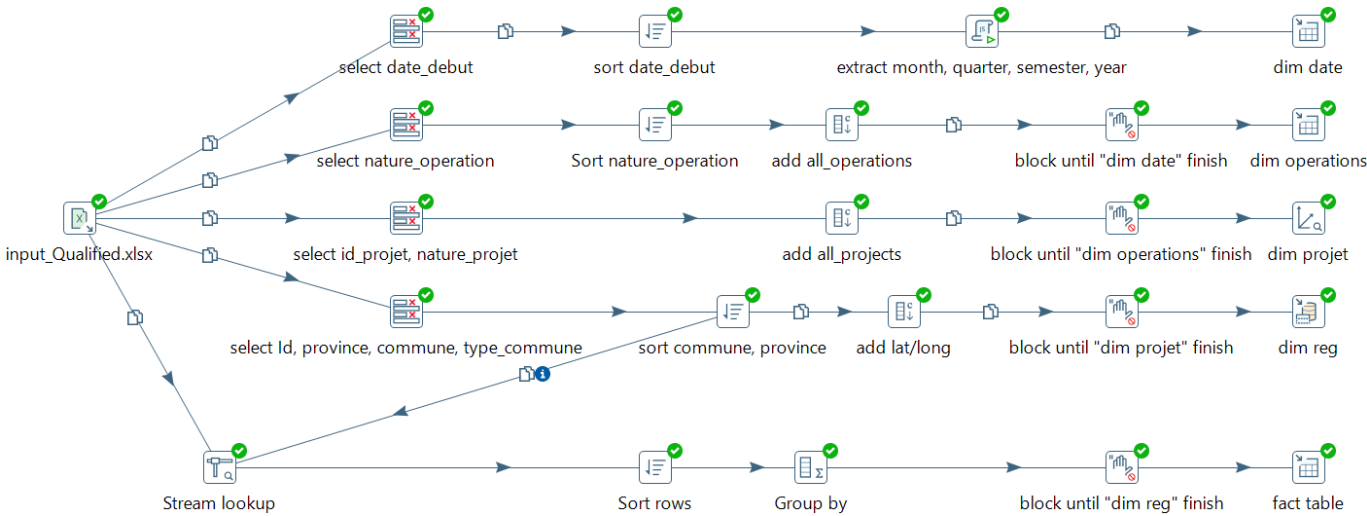


FIGURE 26 – Résultat de l'exécution de la transformation créée sur PDI

2. Les tables de dimension et la table de faits sont maintenant mises à jour avec les données.

omrane\_data\_marts.dim\_date: 1,308 rows total

🔑 date_debut	mois	annee	trimestre
1901-01-01	01/1901	1901	1/1901
1973-09-15	09/1973	1973	3/1973
1974-01-01	01/1974	1974	1/1974
1974-04-19	04/1974	1974	2/1974
1974-08-15	08/1974	1974	3/1974
1975-02-28	02/1975	1975	1/1975
1975-04-05	04/1975	1975	2/1975
1975-05-01	05/1975	1975	2/1975
1975-05-10	05/1975	1975	2/1975
1975-07-17	07/1975	1975	3/1975

FIGURE 27 – Table de la dimension Date

omrane\_data\_marts.dim\_projet: 2,992 rows total (approximately), limited to 1,000

Next

Show all

🔑 tk_projet	version	date_from	date_to	🔑 id_projet	nature_projet	all_projects
183	1	1899-12-31 23:00:00	2038-01-01 00:00:00	182	Construction	all
184	1	1899-12-31 23:00:00	2038-01-01 00:00:00	183	Construction	all
185	1	1899-12-31 23:00:00	2038-01-01 00:00:00	184	Construction	all
186	1	1899-12-31 23:00:00	2038-01-01 00:00:00	185	Construction	all
187	1	1899-12-31 23:00:00	2038-01-01 00:00:00	186	Construction	all
188	1	1899-12-31 23:00:00	2038-01-01 00:00:00	187	Amenagement Foncier	all
189	1	1899-12-31 23:00:00	2038-01-01 00:00:00	188	Amenagement Foncier	all
190	1	1899-12-31 23:00:00	2038-01-01 00:00:00	189	Construction	all
191	1	1899-12-31 23:00:00	2038-01-01 00:00:00	190	Construction	all

FIGURE 28 – Table de la dimension Projet

omrane\_data\_marts.dim\_operations: 11 rows total

nature_operation	all_operations
Autre	all
Commerces	all
Equipements	all
Logements	all
Lotissement Activite	all
Lotissement Habitat	all
Rehabilitation	all
Restructuration QHNR	all
Traitement menacant ruine	all
ZAP	all
ZUN - Ville nouvelle	all

FIGURE 29 – Table de la dimension Operation

omrane\_data\_marts.dim\_reg: 374 rows total (approximately)

Next

Show all

A Z ↓ id_commune	commune	province	type_commune	A Z ↓ latitude	longitude
1	TIZNIT	TIZNIT	Urbaine	(NULL)	(NULL)
2	TARSOUAT	TIZNIT	Rurale	(NULL)	(NULL)
4	REGGADA	TIZNIT	Rurale	(NULL)	(NULL)
5	SIDI IFNI	TIZNIT	Urbaine	(NULL)	(NULL)
6	OURZAZATE	OUARZAZATE	Urbaine	(NULL)	(NULL)
7	TINGHIR	OUARZAZATE	Urbaine	(NULL)	(NULL)
8	BOUIZAKARNE	GUELMIM	Urbaine	(NULL)	(NULL)
10	AGADIR	AGADIR	Urbaine	(NULL)	(NULL)
20	AIT IAAZA	TAROUDANT	Urbaine	(NULL)	(NULL)
22	TAROUDANT	TAROUDANT	Urbaine	(NULL)	(NULL)
23	OULAD TEIMA	TAROUDANT	Urbaine	(NULL)	(NULL)
26	AL FAID	TAROUDANT	Rurale	(NULL)	(NULL)

FIGURE 30 – Table de la dimension Region

omrane\_data\_marts.fact\_table: 2,991 rows total (approximately), limited to 1,000

Next

Show all

id_commune	date_debut	id_projet	nature_operation	cout_operation	Chiffre_D_affaires	superficie
1	1998-02-21	1	Logements	45.46747294	10.204	0.4561
2	1999-11-24	2	Lotissement Habitat	0.70397788	1.127525	0.489475
1	2002-05-01	3	Logements	1.19616497	9.962399	0.442004
4	2006-01-01	4	Lotissement Habitat	9.570037	12.0169	2.1237
5	2003-02-01	5	Logements	3.5792041	0.57	0.74131
6	2005-10-05	6	Lotissement Habitat	136.38171985	152.71615	20.03
7	2005-12-07	7	Lotissement Habitat	133.61247475	92.31805	102.376
8	2009-02-28	8	Lotissement Habitat	45.97500729	92.31805	22.1439
8	2003-10-01	9	Lotissement Habitat	0.222439	0	23.9634
10	2006-11-04	10	Lotissement Habitat	600.53554664	7,485	19.819483
10	2006-11-04	11	Lotissement Habitat	0.3636	0	71
10	2007-05-30	12	Logements	38.1520227	41.85	0.6663

FIGURE 31 – Table de faits

## 6 REPRÉSENTATION MULTIDIMENSIONNELLE AVEC SCHEMA WORKBENCH

### 6.1 CRÉATION DE LA CONNEXION

1. Copiez le fichier .jar du pilote (mysql-connector-java-8.0.20.jar) précédemment téléchargé et collez-le dans le dossier `schema-workbench/lib`
2. Allez dans le logiciel Shema Workbench, cliquez sur **Options** → **Connection...** et remplissez la configuration comme indiqué dans la figure ci-dessous :

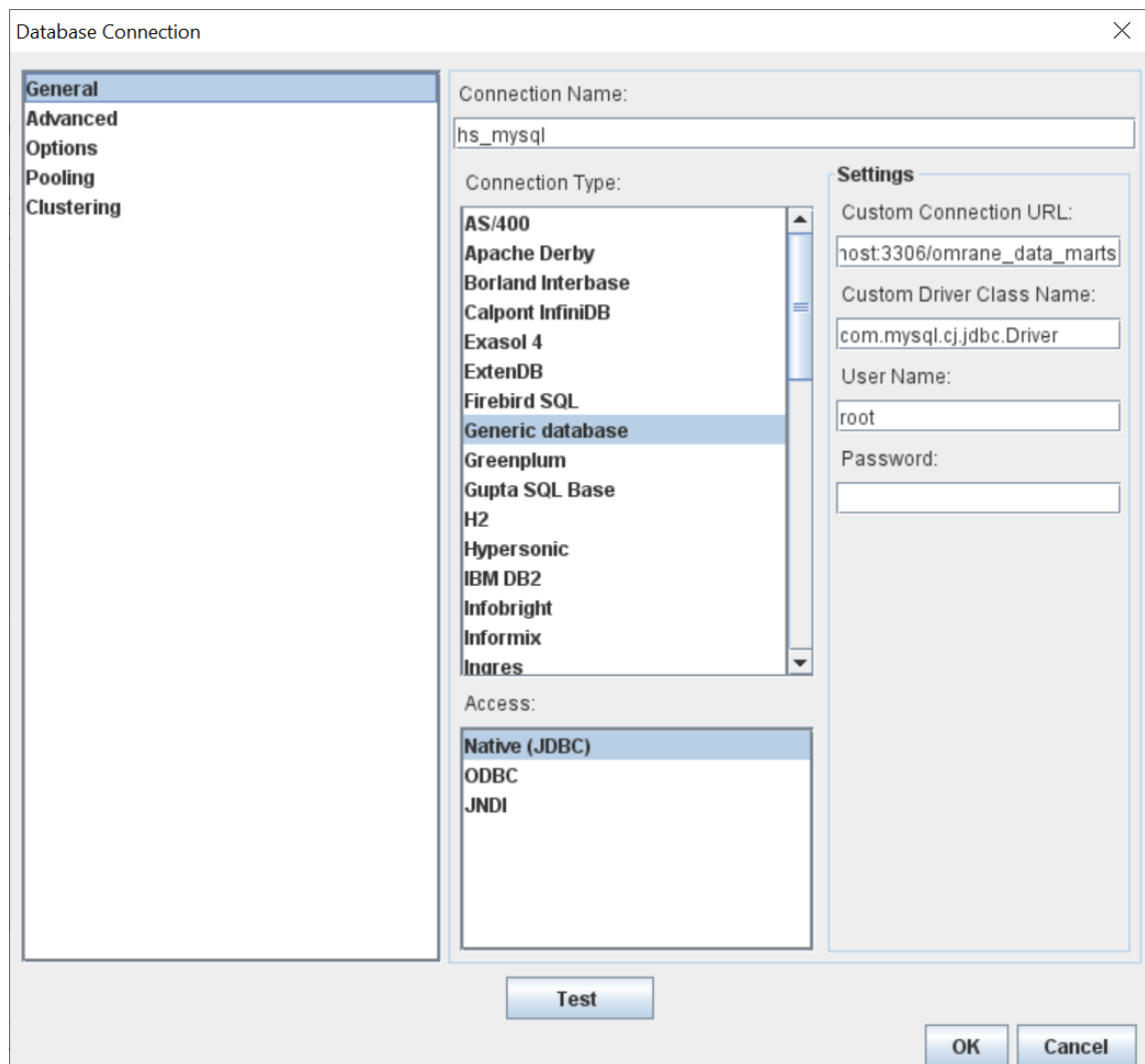


FIGURE 32 – Configuration de la Connexion à la base de données

3. Testez la connexion; Cliquez sur **OK** en cas de succès afin de sauvegrader la connexion.

### 6.2 OLAP CUBE

Un cube OLAP est une base de données multidimensionnelle optimisée pour les Data Warehouses et les applications OLAP. Il s'agit d'une méthode permettant de stocker les données sous forme multidimensionnelle, notamment pour le reporting. En général, ces cubes sont pré-résumés pour accélérer le temps de requête par rapport aux bases de données relationnelles.



FIGURE 33 – OLAP Cube

## 6.2.1 DIMENSIONS

### Code XML : Dimension **DATE**

```

<Dimension type="TimeDimension" visible="true" foreignKey="date_debut" highCardinality="false"
  name="Dim date">
  <Hierarchy name="Date" visible="true" hasAll="true" primaryKey="date_debut">
    <Table name="dim_date">
    </Table>
    <Level name="annee" visible="true" table="dim_date" column="annee" type="String"
      uniqueMembers="false" levelType="TimeYears" hideMemberIf="Never">
    </Level>
    <Level name="Semestre" visible="true" table="dim_date" column="semestre" type="String"
      uniqueMembers="false" levelType="TimeHalfYear" hideMemberIf="Never">
    </Level>
    <Level name="trimestre" visible="true" table="dim_date" column="trimestre" type="String"
      uniqueMembers="false" levelType="TimeQuarters" hideMemberIf="Never">
    </Level>
    <Level name="mois" visible="true" table="dim_date" column="mois" type="String"
      uniqueMembers="false" levelType="TimeMonths" hideMemberIf="Never">
    </Level>
    <Level name="date_debut" visible="true" table="dim_date" column="date_debut" type="Date"
      uniqueMembers="true" levelType="TimeDays" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>

```

### Code XML : Dimension **PROJET**

```

<Dimension type="StandardDimension" visible="true" foreignKey="id_projet" highCardinality="false"
  name="Dim projet">
  <Hierarchy name="Projects" visible="true" hasAll="true" primaryKey="tk_projet">
    <Table name="dim_projet">
    </Table>
    <Level name="all_projects" visible="true" table="dim_projet" column="all_projects" type="
      String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    <Level name="nature_projet" visible="true" table="dim_projet" column="nature_projet" type="
      String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>

```

## Code XML : Dimension **OPERATION**

```
<Dimension type="StandardDimension" visible="true" foreignKey="nature_operation" highCardinality="false" name="Dim operation">
  <Hierarchy name="Operations" visible="true" hasAll="true" primaryKey="nature_operation">
    <Table name="dim_operations">
    </Table>
    <Level name="all_operations" visible="true" table="dim_operations" column="all_operations" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    <Level name="nature_operation" visible="true" table="dim_operations" column="nature_operation" type="String" uniqueMembers="true" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

## Code XML : Dimension **REGION**

```
<Dimension type="StandardDimension" visible="true" foreignKey="id_commune" highCardinality="false" name="Dim region">
  <Hierarchy name="regionH1" visible="true" hasAll="true" primaryKey="id_commune">
    <Table name="dim_reg">
    </Table>
    <Level name="Province" visible="true" table="dim_reg" column="province" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    <Level name="Commune" visible="true" table="dim_reg" column="commune" type="String" uniqueMembers="true" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
  <Hierarchy name="regionH2" visible="true" hasAll="true" primaryKey="id_commune">
    <Table name="dim_reg">
    </Table>
    <Level name="Type de commune" visible="true" table="dim_reg" column="type_commune" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    <Level name="Commune" visible="true" table="dim_reg" column="commune" type="String" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

## 6.2.2 MESURES

### Code XML : **Chiffre d'affaires**

```
<Measure name="chiffre_d_affaire" column="Chiffre_D_affaires" datatype="Numeric" aggregator="sum" visible="true">
</Measure>
```

### Code XML : **Coût d'opération**


```
<Measure name="cout_operation" column="cout_operation" datatype="Numeric" aggregator="sum" visible="true">
</Measure>
```

### Code XML : **Superficie**

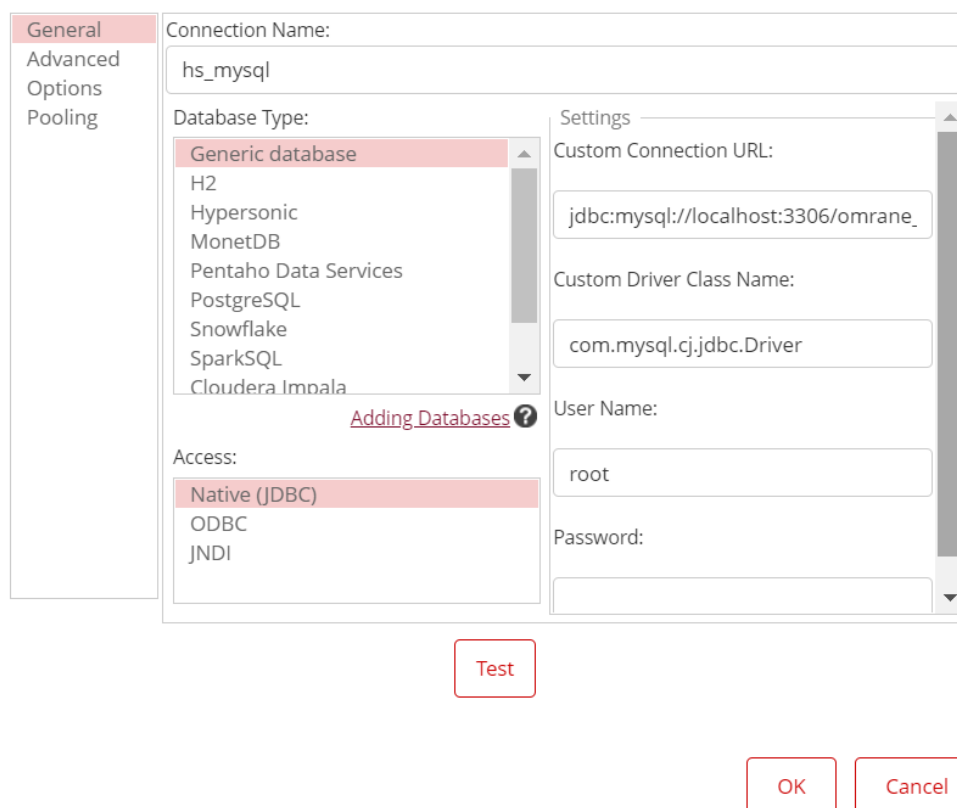
```
<Measure name="superficie" column="superficie" datatype="Numeric" aggregator="sum" visible="true">
</Measure>
```

## 7 DÉPLOIEMENT DE LA SOLUTION SUR PENTAHO SERVER

### 7.1 CRÉATION DE LA CONNEXION SUR PENTAHO SERVER

1. Copiez le fichier .jar du pilote (mysql-connector-java-8.0.20.jar) précédemment téléchargé et collez-le dans le dossier `pentaho-server/tomcat/lib`
2. Lancez le serveur pentaho en cliquant sur `start-pentaho.bat`, puis allez dans le navigateur et ouvrez cette URL `http://localhost:8080/pentaho`
3. Entrez les champs d'identification :
  - **Login** : admin
  - **Password** : password
4. Allez sur `Manage Data Source` →  → `New Connection...`, et remplissez la configuration comme indiqué dans la figure ci-dessous :

#### Database Connection



General  
Advanced  
Options  
Pooling

Connection Name:  
hs\_mysql

Database Type:  
Generic database  
H2  
Hypersonic  
MonetDB  
Pentaho Data Services  
PostgreSQL  
Snowflake  
SparkSQL  
Cloudera Impala


Access:  
Native (JDBC)  
ODBC  
JNDI

Settings  
Custom Connection URL:  
jdbc:mysql://localhost:3306/omrane\_  
Custom Driver Class Name:  
com.mysql.cj.jdbc.Driver  
User Name:  
root  
Password:

Test

OK Cancel

FIGURE 34 – Création d'une nouvelle connexion sur Pentaho Server

5. Testez la connexion; Cliquez sur `OK` en cas de succès afin de sauvegrader la connexion.
6. Cliquez encore une fois sur  → `Import Analysis...` pour importer le schéma du modèle dimensionnel précédemment créé avec shema workbench

7. Choisissez le fichier XML du schéma et sélectionnez le nom de la connexion précédemment créée à l'étape 4.

Import Analysis

Mondrian File:

C:\fakepath\omrane.xml

☒ Select from available data sources.  
☐ Manually enter data source parameter values.

Data Source:

hs\_mysql

Import Close


FIGURE 35 – Importation du schéma `omrane.xml`

## 7.2 VISUALISATION SUR PENTAHO SERVER

### 7.2.1 MÉTHODE TYPIQUE POUR CRÉER UN GRAPHIQUE SUR UN TABLEAU DE BORD CDE :

Cette tâche suppose que vous êtes dans la console utilisateur Pentaho (PUC).

Avant de commencer à créer un tableau de bord, il est important de définir un flux de travail :

1. **Planifiez votre tableau de bord.** Avant de commencer à créer votre tableau de bord, pensez à quoi il ressemblera et comment il se comportera.
2. **Créer un nouveau tableau de bord CDE.** Cliquez sur **Create New** → **CDE Dashboard**
3. **Créez et enregistrez votre tableau de bord.** Saisissez un nom pour votre tableau de bord et enregistrez-le dans le dossier solutions de votre choix.
4. **Créez le layout du tableau de bord.** Dans l'éditeur, sélectionnez le Layout et définissez le tableau de bord layout en fonction de votre conception.
5. **Ajoutez les sources de données.** Sélectionnez les sources de données qui alimenteront votre tableau de bord.
6. **Ajoutez et configurez les composants.** Ajoutez les composants qui composent votre tableau de bord : les éléments visuels, les paramètres et éventuellement certains scripts.
7. **Prévisualisez votre travail.** Cliquez sur l'icône de Preview  régulièrement pour voir comment votre travail progresse.

### 7.2.2 APPLICATION 1 : GRAPHIQUE À SECTEURS (PIE CHART) SUR UN TABLEAU DE BOARD CDE :

1. Dans le panneau de layouts, sélectionnez le le layout (template) du tableau de bord.

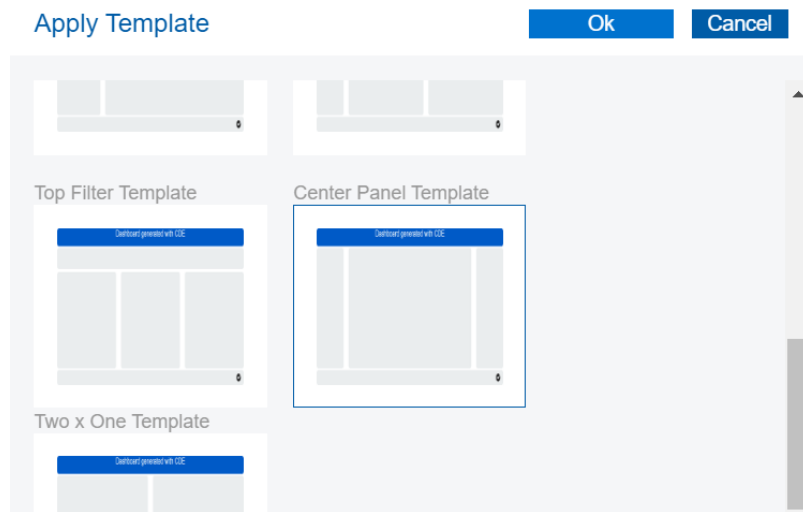


FIGURE 36 – Application d'un modèle dans le tableau de bord

2. Définissez les propriétés du graphique dans un assistant OLAP (OLAP Wizard) :
  - **Name** : pie\_chart
  - **Html Object** : Panel\_1
  - **Catalog** : omrane
  - **Cube** : omrane\_cube
  - **Chart Type** : Pie Chart
3. Glissez les mesures vers les colonnes (Columns) et les attributs de dimension vers les lignes (Rows).



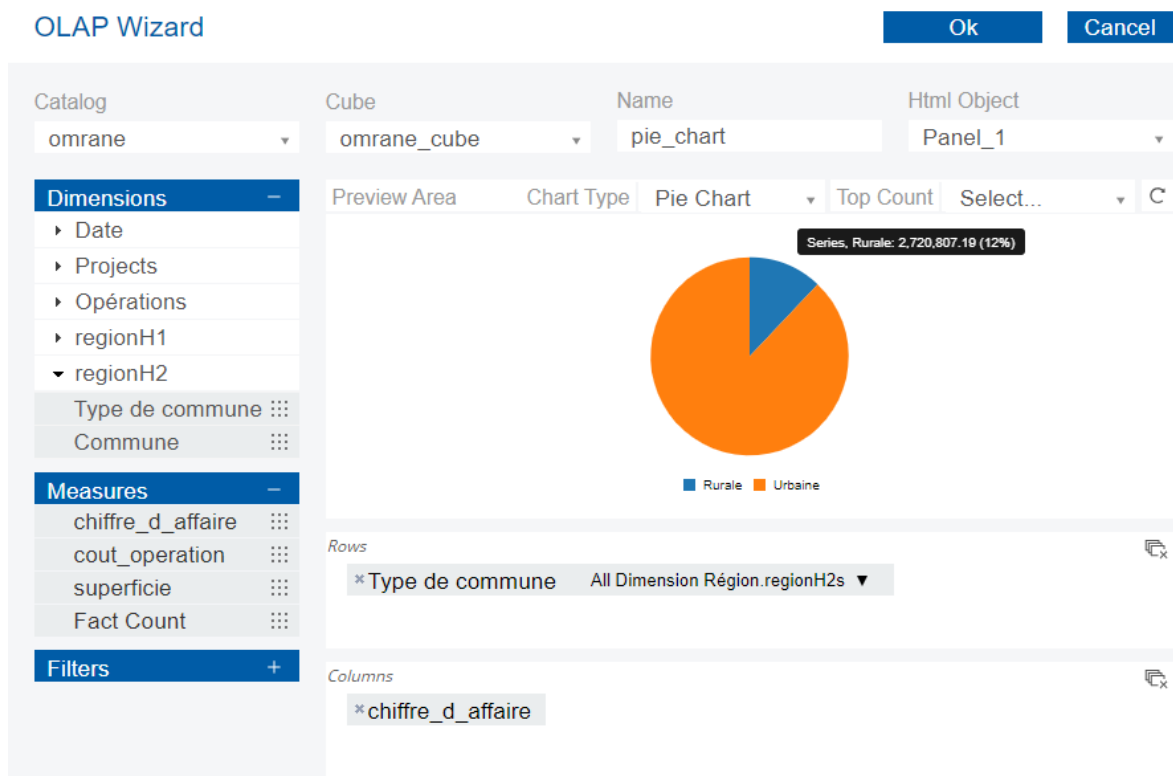


FIGURE 37 – Répartition du chiffre d'affaire par type de commune

4. Cliquez sur **OK** pour insérer le graphique dans le modèle de tableau de bord.

### 7.2.3 APPLICATION 2 : RÉPARTITION GÉOGRAPHIQUE SUR UNE CARTE CDE :

#### Création du modèle de tableau de bord CDE

1. Cliquez sur pour ajouter une division HTML de type DIV et entrez les propriétés suivantes :

Name	container
Css Class	container
Element Tag	div

2. Sélectionnez cette division Container et ajoutez-y une autre à l'intérieur avec les propriétés suivantes :

Name	title
Css Class	title
Element Tag	div

3. Sélectionnez maintenant la division title et cliquez sur pour ajouter du contenu HTML :

```
<h2>Répartition géographique des <span>coûts</span>, des  
<span>CA</span> et des <span>superficies</span></h2>
```

4. Ajoutez les divisions suivantes à l'intérieur de Container :

- Une division Bootstrap de type **Row**

<b>Name</b>	map_container
<b>Css Class</b>	map_container

- **Espace html**


- Une division Bootstrap de type **Row**

<b>Name</b>	popup
-------------	-------

Le balisage HTML devrait ressembler à ceci :

```
<div id="container" class="container">
  <div id="title" class="title">
    <h2>Répartition géographique des <span>coûts</span>, des
    <span>CA</span> et des <span>superficies</span></h2>
  </div>
  <div id="map_container" class="map_container row clearfix"></div>
  <hr class="space" style="height:0px;">
  <div id="popup" class="row clearfix"></div>
</div>
```

## Ajout des Resources css

Cliquez sur  (Add Resource) pour ajouter une ressource CSS :

```
*{
  box-sizing: border-box;
  padding: 0;
  margin: 0;
}
.container{
  height: 400px;
  min-height: 100%;
}
.title{
  text-align: center;
}
.title h2 span{
  color: blue;
}
.map_container{
  height: 100%;
  width: 100%;
  margin: 0;
}
```

```

/* fixed bugs in popup */
#popup.row{
    margin: 0;
}
#popupprotovis svg, #featurePopup394246_contentDi{
    width: 100% !important;
    height: auto !important;
}
#featurePopup394246{
    border: 2px solid #f1828d !important;
}

```

## Ajout de composants et de paramètres de la cartographie

Allez dans le panneau de composants  et ajoutez les composants suivantes :

1. Dans le menu de gauche, cliquez sur **Parameters** → **Simple Parameter** et entrez la configuration suivante :

<b>Name</b>	province
<b>Property value</b>	-

2. Dans le menu de gauche, cliquez sur **Standard** → **Map Component** et entrez la configuration suivante :

<b>Center Latitude</b>	28.75
<b>Center Longitude</b>	-8
<b>Default zoom Level</b>	5
<b>Shape Fill Opacity</b>	-
<b>Location Resolver AddIn</b>	OpenStreetMap
<b>Map Engine</b>	openlayers
<b>Operation Mode</b>	Markers
<b>Marker image</b>	-
<b>Marker Height</b>	25
<b>Marker Width</b>	25
<b>Name</b>	map
<b>Parameters</b>	[["province",""]]
<b>Div for popup window</b>	popup
<b>Popup Height</b>	-
<b>Marker Click Parameters</b>	[["province","province"]]
<b>Popup Width</b>	320
<b>Datasource</b>	geo
<b>Priority</b>	5
<b>HtmlObject</b>	map_container
<b>Execute at start</b>	True

3. Dans le menu de gauche, cliquez sur **Charts** → **CCC Bar Chart** et entrez la configuration suivante :

<b>Title</b>	-
<b>Listeners</b>	["province"]
<b>Parameters</b>	[["province","province"]]
<b>Datasource</b>	data
<b>Height</b>	360
<b>Name</b>	barchart
<b>Width</b>	360
<b>HtmlObject</b>	popup
<b>Compatibility Version</b>	3
<b>Crosstab Mode</b>	True
<b>Series In Rows</b>	False
<b>Time Series</b>	False

Pre Execution :

```
function f(){
    var province = this.dashboard.parameters["province"];
    this.chartDefinition.title = "Province: " + province;
}
```

**Remarque :** Laissez les champs non spécifiés avec les valeurs par défaut.

### Ajout de ressources de données

Il existe plusieurs façons de définir une source de données (NoSQL, requêtes MDX, requêtes SQL, ...). Dans notre cas nous utiliserons des requêtes sql pour définir les sources de données.

Allez dans le panneau de source de données  et ajoutez les sources de données suivantes :

1. Dans le menu de gauche, cliquez sur **SQL Queries** → **sql over sqlJndi** et entrez la configuration suivante :

<b>Name</b>	data
<b>Access Level</b>	Public
<b>Jndi</b>	hs_mysql
<b>Parameters</b>	[["province","province","String","", ""]]

Query :

```
SELECT fact_table.commune ,
sum(fact_table.chiffre_d_affaires) AS "Chiffre d'affaires",
sum(fact_table.cout_operation) AS "Coût d'opération",
sum(fact_table.Superficie) AS "Superficie"
FROM fact_table
INNER JOIN dim_reg ON dim_reg.commune = fact_table.commune
WHERE dim_reg.province=${province}
GROUP BY commune;
```


2. Créez une nouvelle source de données `sql over sqlJndi` et entrez la configuration suivante :

Name	geo
Access Level	Public
Jndi	hs_mysql

Query :

```
SELECT DISTINCT province,latitude,longitude from dim_reg;
```

### Visualisation de la carte

Cliquez sur , ou bien accédez directement au fichier du tableau de bord pour pouvoir visualiser la MAP :

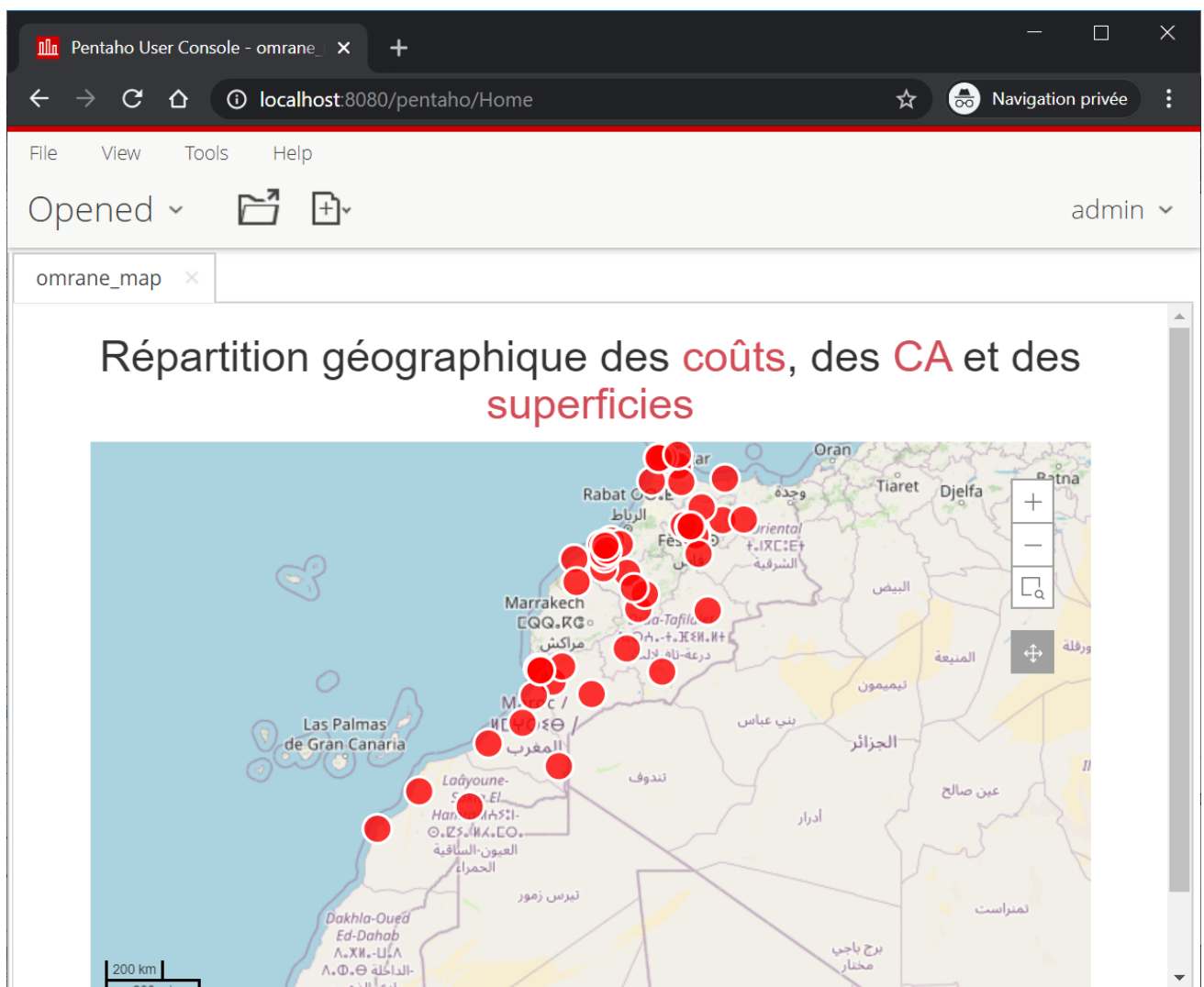


FIGURE 38 – Carte de la répartition géographique des coûts, des CA, et des superficies par province

Maintenant, cliquez sur n'importe quel marqueur pour afficher le graphique à bandes (Bar chart) correspondant.

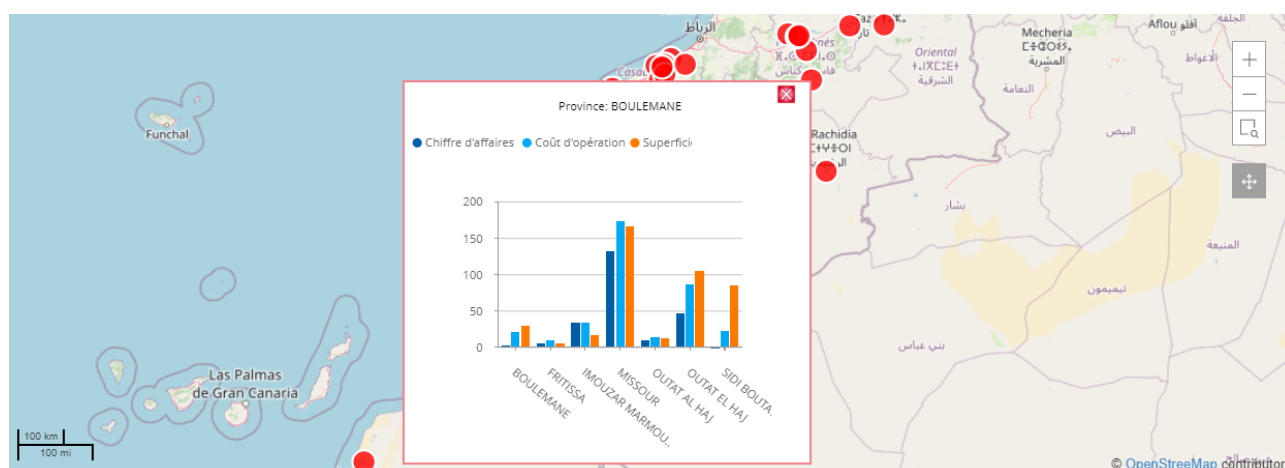


FIGURE 39 – Exemple : Répartition des coûts, des CA, et des superficies dans la province de BOULEMANE

## 7.2.4 APPLICATION 3 : ÉVOLUTION DU CHIFFRE D'AFFAIRES EN FONCTION DES COÛTS

Dans cette application, nous utiliserons le plugin **datafor-visualizer** de son développeur [Shanghai Datafor](#).

Ce plugin a de nombreuses fonctionnalités et caractéristiques :

- **Facile à utiliser** : conception par «glisser-déposer», panneau de données intelligent, WY-SI WYG.
- **Éléments de visualisation** : y compris le graphique à barres, le graphique linéaire, le graphique en aires, le graphique circulaire, etc.
- **Style** : réglage de style riches, y compris la police, la couleur, la mise en page, la taille.
- **Analyse** : exploration, liaison, saut, rotation, découpage, filtrage avancé, etc.

Pour commencer à utiliser ce plugin :

1. Allez dans la page Marketplace.
2. Entrez "Datafor-visualizer" pour rechercher le plugin.
3. Installez le plugin et redémarrez le serveur pentaho.
4. Vous pouvez désormais créer des graphiques, pour ce faire cliquez sur [Create New](#) → [Visualizer](#).

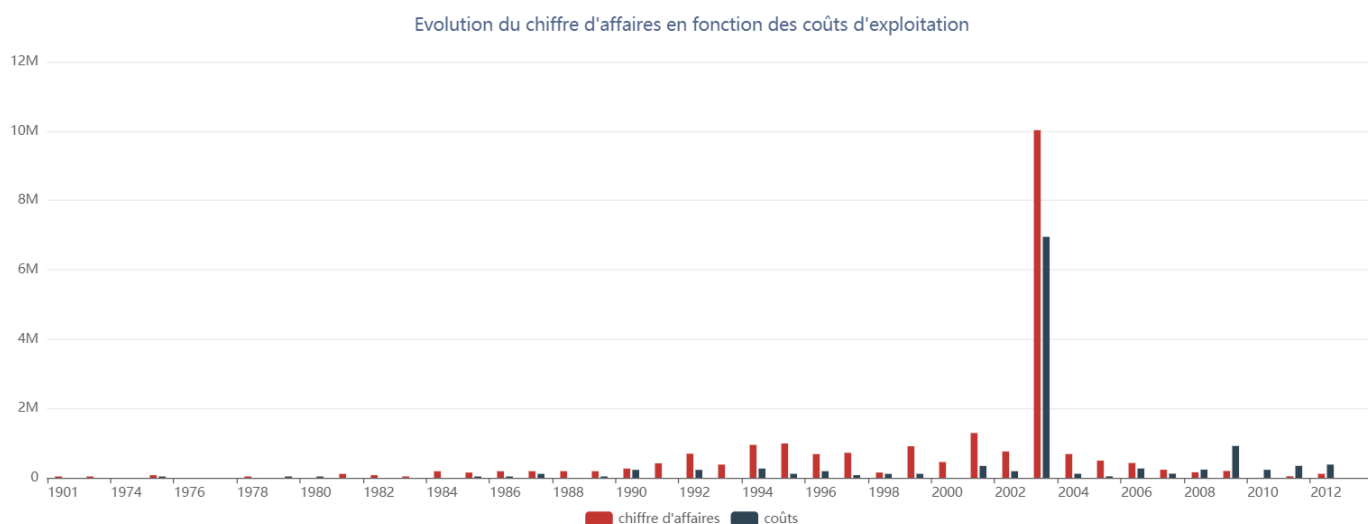


FIGURE 40 – Évolution du chiffre d'affaires en fonction des coûts pour la période de 1901 à 2012

## 7.2.5 APPLICATION 4 : CHIFFRE D'AFFAIRES, COÛTS D'EXPLOITATION, ET SUPERFICIE PAR PROVINCE

Tout comme l'application 3, nous utiliserons le plugin **datafor-visualizer**. Cette fois pour créer un graphique plus complexe.

Il s'agit du graphique de dispersion (Scatter plot) représentant les CA, les coûts et les superficies par province des 8 premières provinces classées par CA.

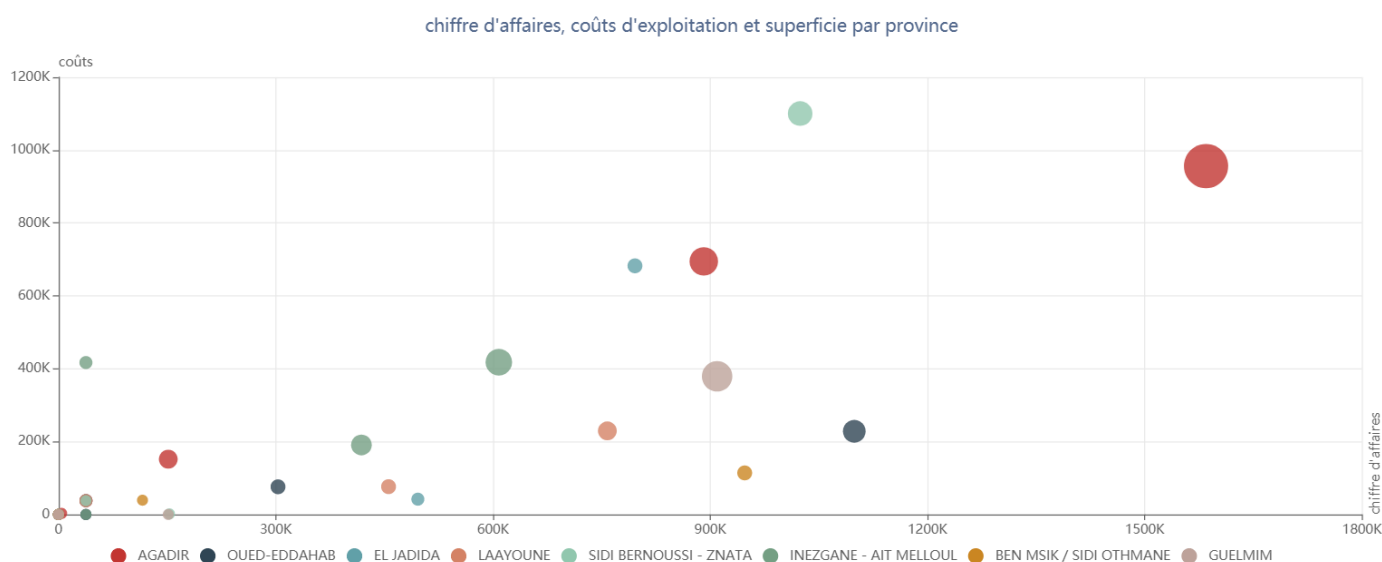


FIGURE 41 – CA, coûts, et superficies par province

## 8 AUTOMATISATION DES TÂCHES AVEC JENKINS

Nous utiliserons le serveur d'automatisation Jenkins pour automatiser l'exécution des transformations KTr de manière à ce que les transformations soient exécutées toutes les 6 heures.

### 8.1 CONFIGURATION JENKINS

Configuration de l'envoi d'un e-mail :

1. Survolez et cliquez sur le lien **Jenkins** → **Manage Jenkins** → **Configure System**.
2. Descendez dans la section **E-mail Notification** :
  - (a) Enter your SMTP server name to "SMTP server"
  - (b) Cliquez sur "Advanced"
  - (c) Cochez la case "Use SMTP Authentication"
  - (d) Entrez les informations requises pour l'authentification

The screenshot shows the Jenkins configuration interface. At the top, there's a breadcrumb: **Jenkins** > **configuration**. Below this is the **E-mail Notification** section. It contains several fields:   
- **SMTP server**: A text box containing 'smtp.gmail.com'.   
- **Default user e-mail suffix**: An empty text box with a help icon.   
- **Use SMTP Authentication**: A checkbox that is checked.   
- **User Name**: A text box containing 'hamza.talaghzi@gmail.com'.   
- **Password**: A field with a lock icon, the text 'Concealed', and a 'Change Password' button.   
- **Use SSL**: A checked checkbox.   
- **Use TLS**: An unchecked checkbox.   
- **SMTP Port**: A text box containing '465'.   
- **Reply-To Address**: An empty text box with a help icon.   
- **Charset**: A text box containing 'UTF-8'.   
At the bottom, there is an unchecked checkbox labeled 'Test configuration by sending test e-mail'. Below the form are two buttons: **Save** (in a dark blue box) and **Apply** (in a light gray box).

FIGURE 42 – CA, coûts, et superficies par province

3. Testez la configuration; Cliquez sur **Save** en cas de succès afin de sauvegrader la configuration.

### 8.2 CRÉATION D'UNE TÂCHE (JOB) RÉGULIÈRE

1. Cliquez sur le lien **New Item** du menu gauche.



2. Renseignez le nom du job.
3. Sélectionnez l'option **Freestyle project**. Puis cliquez sur le bouton **OK**



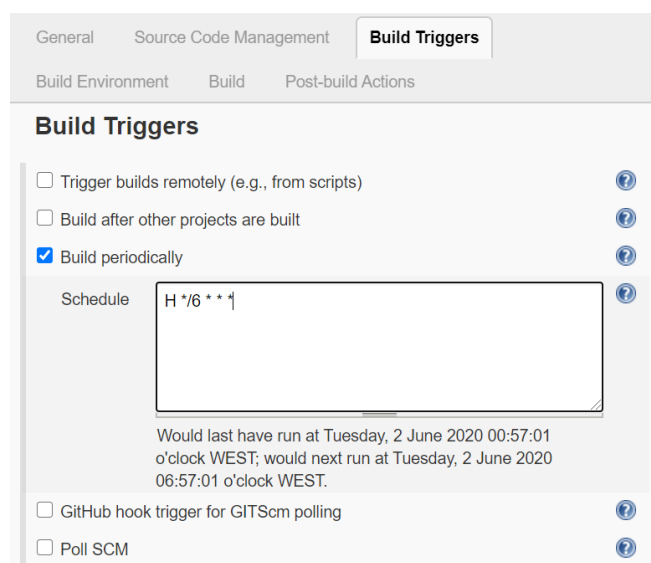
The image shows the Jenkins 'Enter an item name' dialog box. At the top, there's a header with the Jenkins logo and navigation icons. Below the header, the title 'Enter an item name' is followed by a text input field containing 'nom\_du\_job'. A small note below the field says '» Required field'. Below the input field, there are three options, each with an icon and a description:

- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

At the bottom left, there is a blue 'OK' button.

4. Cliquez sur le lien **Build Triggers** et cochez la case **Build periodically**.
5. Modifier la configuration de build pour automatiser le lancement de la build toutes les 6 heures.

H \*/6 \* \* \*



The image shows the Jenkins 'Build Triggers' configuration page. At the top, there are tabs for 'General', 'Source Code Management', and 'Build Triggers' (which is selected). Below the tabs, there are sub-tabs for 'Build Environment', 'Build', and 'Post-build Actions'. The main section is titled 'Build Triggers' and contains the following options:

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☒ Build periodically

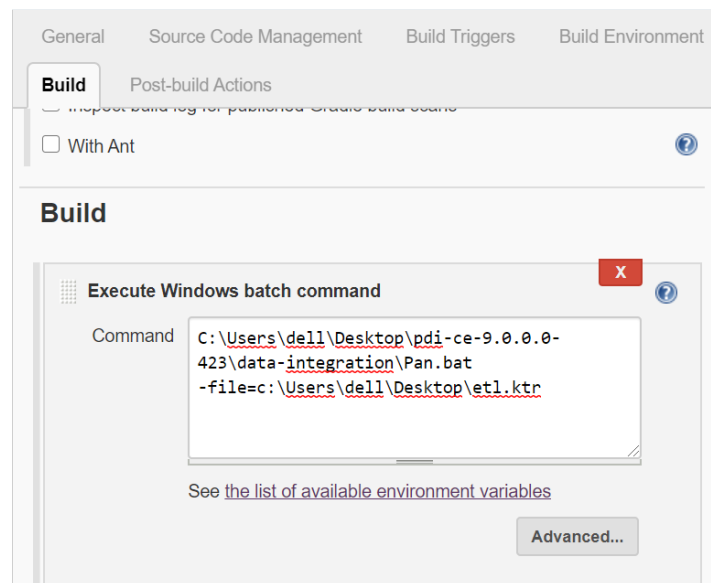
Below the 'Build periodically' option, there is a 'Schedule' field with the text 'H \*/6 \* \* \*'. Below the schedule field, there is a note: 'Would last have run at Tuesday, 2 June 2020 00:57:01 o'clock WEST; would next run at Tuesday, 2 June 2020 06:57:01 o'clock WEST.' At the bottom, there are two more options:

- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

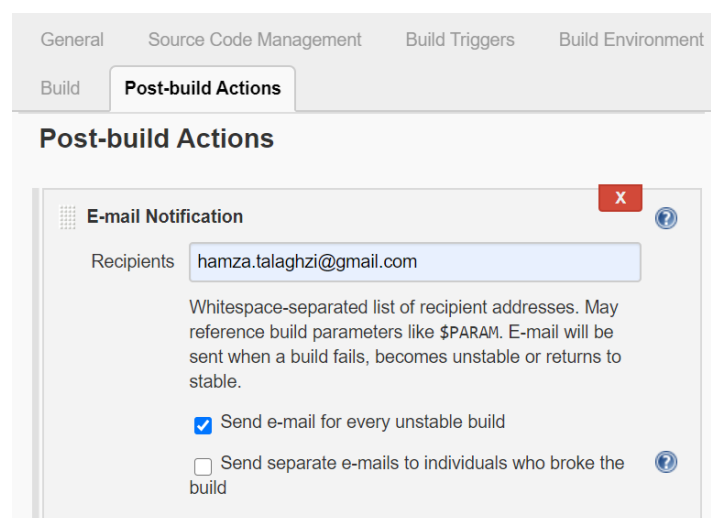
6. Cliquez maintenant sur le lien **Build**, puis sélectionnez l'option **Execute Windows batch command**.
7. Saisissez le chemin absolu du script (**Kitchen** ou **Pan**) suivi de l'argument **-file** puis du chemin du fichier de transformation / job.

Dans notre cas, la commande ressemble à ceci :

```
C:\Users\dell\Desktop\pdi-ce-9.0.0.0-423\data-integration\Pan.bat  
-file=c:\Users\dell\Desktop\etl.ktr
```



8. Cliquez maintenant sur le lien **Post-build Actions**, puis cliquez sur **Add post-build action** et choisissez **E-mail Notifications**.
9. Saisissez les e-mails des destinataires.  
Notez qu'un e-mail sera envoyé lorsqu'une build échoue, devient instable ou revient à stable.

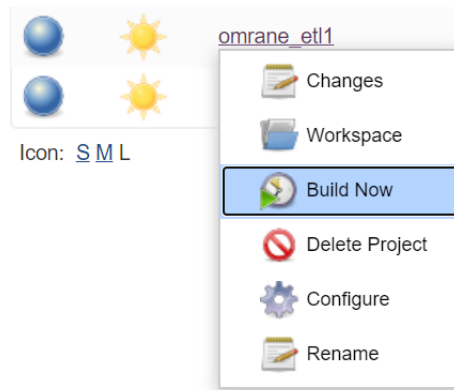


10. Cliquez sur **Save** afin de sauvegrader le Job.

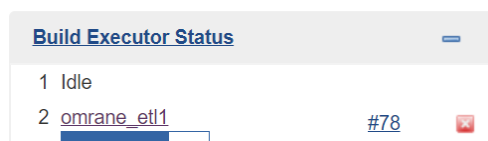
## 8.3 ÉPREUVE DE LA BUILD

Pour tester le fonctionnement du Job, vous pouvez lancer une build instantanément :

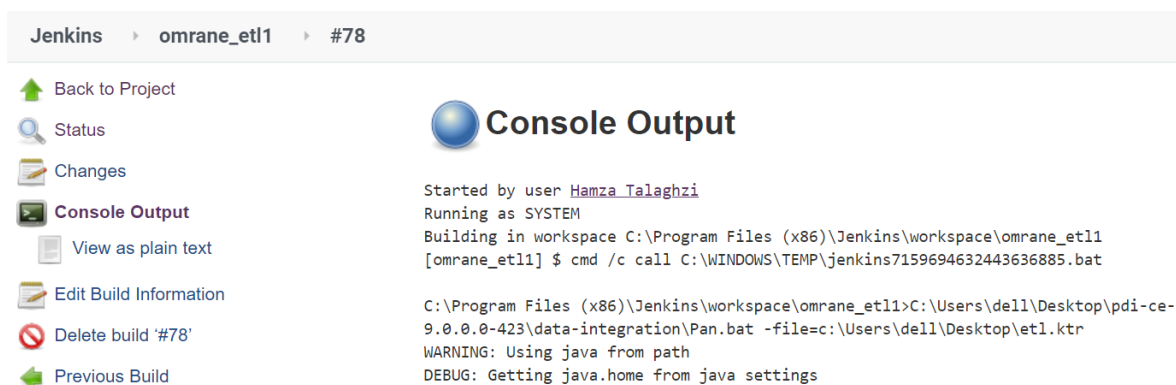
1. Dans le tableau de bord Jenkins, cliquez sur le nom du Job, dans son menu contextuel, cliquez sur [Build Now](#).



2. Vous pouvez voir le statut de la build en cours d'exécution dans le panneau [Build Executor Status](#) situé à gauche.



3. Analysez les logs console en cliquant sur le lien [Console Output](#).



### A.1 EXTRACTION DE LATITUDE ET LONGITUDE À L'AIDE D'API DE GÉOCODAGE EXTERNES, ET DE PDI

Le but de cette implémentation est de pouvoir alimenter la table de dimension REGION avec les valeurs de latitude et de longitude de chaque province afin de les cartographier et de les visualiser correctement sur une carte Map.

Pour atteindre cette solution, nous utiliserons deux API de géocodage (Forward geocoding) pour convertir un texte (Province) en coordonnées géographiques (latitude, longitude) :

1. **Mapbox** est une plate-forme de données de localisation pour les applications mobiles et Web. Ils fournissent des blocs de construction pour ajouter des fonctionnalités basées sur la localisation telles que des cartes, la recherche et la navigation dans toute expérience que vous créez.

Pour **Mapbox**, le format de l'URL peut être écrit sous la forme suivante :

[/geocoding/v5/mapbox.places/{search\\_text}.json?bbox={bounding\\_box}&access\\_token={SECRET\\_KEY}](https://api.mapbox.com/geocoding/v5/mapbox.places/{search_text}.json?bbox={bounding_box}&access_token={SECRET_KEY})

- **search\_text** : La fonctionnalité que vous essayez de rechercher. Il peut s'agir d'une adresse, d'un nom de point d'intérêt, d'un nom de ville, etc.
- **bbox** : Un cadre de délimitation (**bounding box**) est un mécanisme permettant de décrire une zone particulière d'une carte. Il est généralement exprimé sous la forme d'un tableau de paires de coordonnées, la première paire de coordonnées se référant au coin sud-ouest de la boîte (la longitude et la latitude minimales) et la seconde se référant au coin nord-est de la boîte (la longitude et la latitude maximales).
- **access\_token** : Lorsque vous vous inscrivez pour un compte, un jeton privé vous est attribué. Toutes les demandes de géocodage doivent inclure ce jeton d'accès.[5]

2. **LocationIQ** est une plate-forme de données qui offre un bouquet de services de localisations aux développeurs, axés principalement sur la fourniture d'API pour le géocodage, les cartes et le routage.

Pour **LocationIQ** les demandes peuvent être envoyées à l'un des points de terminaison suivants :

Region 1 : US

[https://us1.locationiq.com/v1/search.php?key={YOUR\\_PRIVATE\\_TOKEN}&q={SEARCH\\_STRING}&format=json](https://us1.locationiq.com/v1/search.php?key={YOUR_PRIVATE_TOKEN}&q={SEARCH_STRING}&format=json)

Region 2 : Europe

[https://eu1.locationiq.com/v1/search.php?key={YOUR\\_PRIVATE\\_TOKEN}&q={SEARCH\\_STRING}&format=json](https://eu1.locationiq.com/v1/search.php?key={YOUR_PRIVATE_TOKEN}&q={SEARCH_STRING}&format=json)

- **SEARCH\_STRING** : Chaîne de requête à rechercher. Les virgules sont facultatives, mais améliorent les performances en réduisant la complexité de la recherche. Ne pas combiner avec des paramètres structurés / code postal. [6]

## A.1.1 CONCEPTION DE L'ALGORITHME

Nous utiliserons cet algorithme pour mettre à jour la table de la dimension Région avec les valeurs de latitude et de longitude de chaque province :

```
for each province in provinces do
  if province.latitude == NULL then
    res = API1.getJSON();
    province.latitude = parseJSON(res, API1_PATH_lat)
    province.longitude = parseJSON(res, API1_PATH_lon)
    if province.latitude == NULL then
      res = API2.getJSON();
      province.latitude = parseJSON(res, API2_PATH_lat)
      province.longitude = parseJSON(res, API2_PATH_lon)
    endif
  endif
  updateTable(dim_reg, province.latitude, province.longitude)
endfor
```

## A.1.2 RÉALISATION DE LA SOLUTION

1. Extraire les provinces dont les coordonnées sont nulles
  - (a) Pour commencer, allez d'abord dans le logiciel **Pentaho Data Integration**, et créer une nouvelle transformation.
  - (b) Glissez l'étape **Table Input** dans l'espace de travail.
  - (c) Sélectionnez le champ (province) de la table de dimension Region dont les coordonnées sont nulles.

Table input

Step name: select province from dim\_reg

Connection: hs\_mysql [Edit... New... Wizard...]

SQL: [Get SQL select statement...]

SELECT distinct province FROM dim\_reg WHERE latitude is NULL;

Line 1 Column 61

Store column info in step meta data ☐

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step [v]

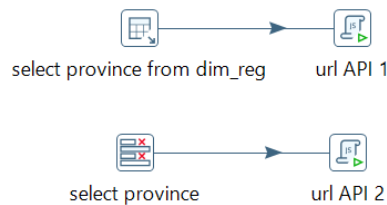
Execute for each row? ☐

Limit size: 0

[Help] [OK] [Preview] [Cancel]

2. Sélectionnez province et générer les URL API pour chacune des deux fournisseurs de géocodage **Mapbox** et **LocationIQ**.

- (a) Ajoutez les étapes **Select values** et **Modified JavaScript value** pour chacune des deux URL API dans l'espace de travail.



Nous utiliserons l'étape **Modified JavaScript value** pour générer dynamiquement l'URL de chaque API.

- (b) Utilisez les scripts ci-dessous pour générer les URL des deux API :

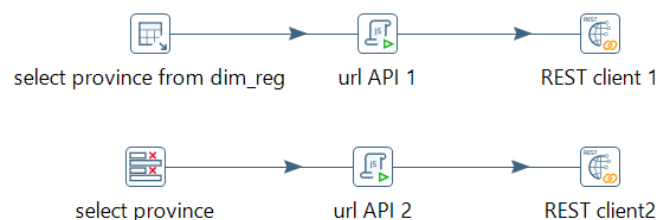
#### Mapbox API

```
var url = "https://api.mapbox.com/geocoding/v5/mapbox.places/" + encodeURIComponent(province) + ".json?bbox=-17.1217979999044,21.4124730239044,-0.996975110129415,35.976465997498&access_token=pk.eyJ1IjoilWgiLCJhIjoiiY2thZXJwZXdiMjJ5NTJ6bXRhbTQxMGYyaSJ9.9ZxoFODGZZ2a66dTtm1ZoQ"
```

#### LocationIQ API

```
var url = "https://eu1.locationiq.com/v1/search.php?key=ddb388db41fd72&q="+encodeURIComponent(province)+"&format=json";
```

3. Utilisez la transformation **REST client** pour envoyer la requête d'API et stocker le résultat dans un champ.



REST client

Step name: REST client 1

General | Authentication | SSL | Headers | Parameters | Matrix Parameters

Settings

URL:

Accept URL from field? ☒

URL field name: url

HTTP method: GET

Get Method from field: ☐

Method field name:

Body field:

Application type: TEXT PLAIN

Output fields

Result field name: result

HTTP status code field:

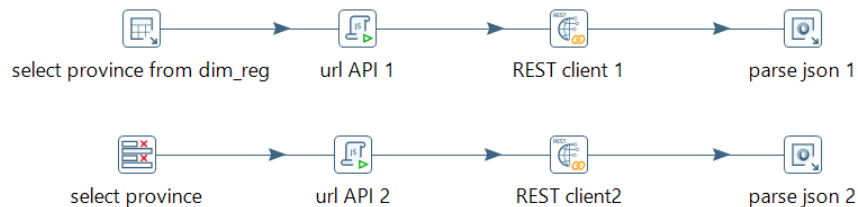
Response time:

Response header field:

Help OK Cancel

FIGURE 43 – Étape **REST client**

4. Analyser le résultat JSON et extraire les valeurs de latitude et de longitude.



Exemple d'extraction des valeurs de latitude et de longitude renvoyées par l'objet JSON **res**.

JSON input

Step name: parse json 1

File | Content | Fields | Additional output fields

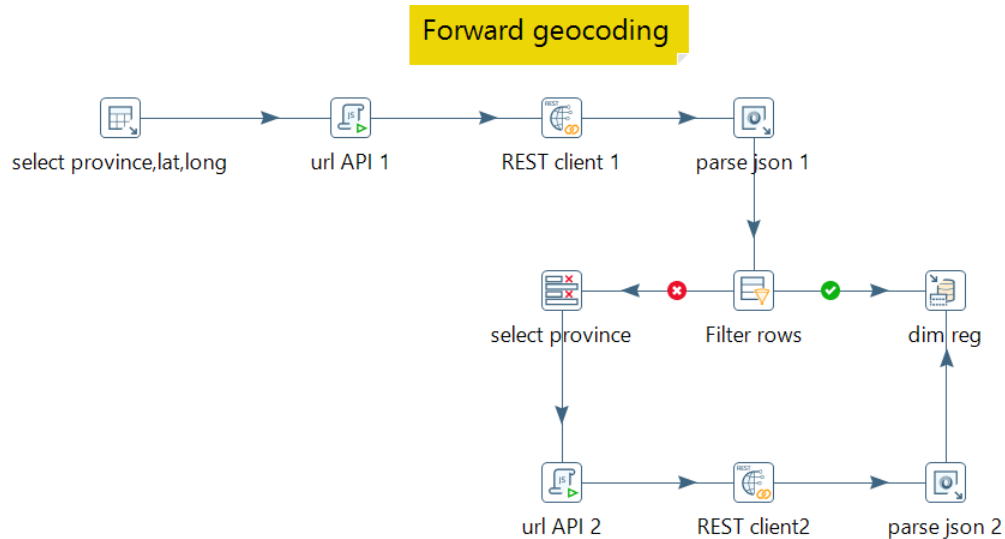
#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	latitude	\$.features[0].geometry.coordinates[1]	Number							none	N
2	longitude	\$.features[0].geometry.coordinates[0]	Number							none	N

Select fields

Help OK Preview rows Cancel

FIGURE 44 – Étape **JSON input**

5. Filtrer les résultats obtenus par la 1ère requête API :
  - Si les résultats des champs de latitude et de longitude sont nuls, alors :  
Appelez la 2ème API pour refaire la recherche
  - Sinon :  
Mettez à jour la table de la dimension Région.



### A.1.3 AUTOMATISATION DE LA TÂCHE

Nous suivons la même procédure décrite dans la section 8.2, à l'exception d'un petit changement dans la façon dont la transformation doit être déclenchée :

1. Cochez l'option **Build after other projects are built** pour pouvoir déclencher le "job" automatiquement après l'exécution d'autres projets.
2. Renseignez le nom du job à surveiller.
3. Choisissez quand il doit être exécuté; parmi les 3 trois options proposées. (Voir figure ci-dessous)

General Source Code Management **Build Triggers**

Build Environment Build Post-build Actions

**Build Triggers**

☐ Trigger builds remotely (e.g., from scripts) ?

☒ Build after other projects are built ?

Projects to watch

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?



## ACRONYMES

---

API	Application Programming Interface
BA	Business Analytics
BI	Business Intelligence
CA	Chiffre d’Affaires
CDA	Community Data Access
CDE	Community Dashboard Editor
CE	Community Edition
DWH	Data Warehouse
EE	Enterprise Edition
ETL	Extract Transform Load
HTTP	HyperText Transfer Protocol
JDBC	Java Database Connectivity
JDK	Java Development Kit
JNDI	Java Naming and Directory Interface
JSON	JavaScript Object Notation
KJB	Kettle Job
KTR	Kettle Transformation
MDX	MultiDimensional eXpressions
OLAP	Online Analytical Processing
OS	Operating System
PDI	Pentaho Data Integration
PSW	Pentaho Schema Workbench
PUC	Pentaho User Console
RAM	Random Access Memory
REST	Representational State Transfer
SCD	Slowly Changing Dimension
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator
WAR	Web Archive
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language

## Références

---

- [1] Fact tables and entities → [https://www.ibm.com/support/knowledgecenter/fr/SS9UM9\\_9.1.1/com.ibm.datatools dimensional.ui.doc/topics/c\\_dm\\_fact\\_tables.html](https://www.ibm.com/support/knowledgecenter/fr/SS9UM9_9.1.1/com.ibm.datatools dimensional.ui.doc/topics/c_dm_fact_tables.html)
- [2] Installing Jenkins → <https://www.jenkins.io/doc/book/installing/>
- [3] Jenkins : Java requirements → <https://www.jenkins.io/doc/administration/requirements/java/>
- [4] Pentaho Data Integration SQL connection → <https://stackoverflow.com/a/61620704/7134209>
- [5] Mapbox search service → <https://docs.mapbox.com/api/search/#geocoding>
- [6] LocationIQ : Search / Forward Geocoding → <https://locationiq.com/docs#forward-geocoding>