

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Ce workshop vise à prendre en main le concept de l'ETL, qui consiste à extraire des données hétérogènes depuis n'importe quelle source, à les manipuler, puis à les stocker pour une éventuelle analyse.

Le WS est divisé en trois parties. Les deux premières parties vous permettront de prendre en main l'outil Talend TOSBD, tandis que la troisième partie consistera à construire le modèle multidimensionnel et à créer un Data Lake dans HDFS

Quelques figures sont présentes pour aider à la réalisation des différentes manipulations.

Partie 1 :

Cette partie consiste à prendre en main Talend en manipulant quelques composants. Commencez par créer votre Job, puis déposez le composant "tRowGenerator" qui nous permettra de générer de nouvelles données. Nous souhaitons obtenir en sortie :

- Id : de type entier
- Nom
- Prénom
- Date de naissance : comprise entre 2000 et 2010
- Ville de naissance
- Une adresse mail sous forme prenom.nom@cesi.fr

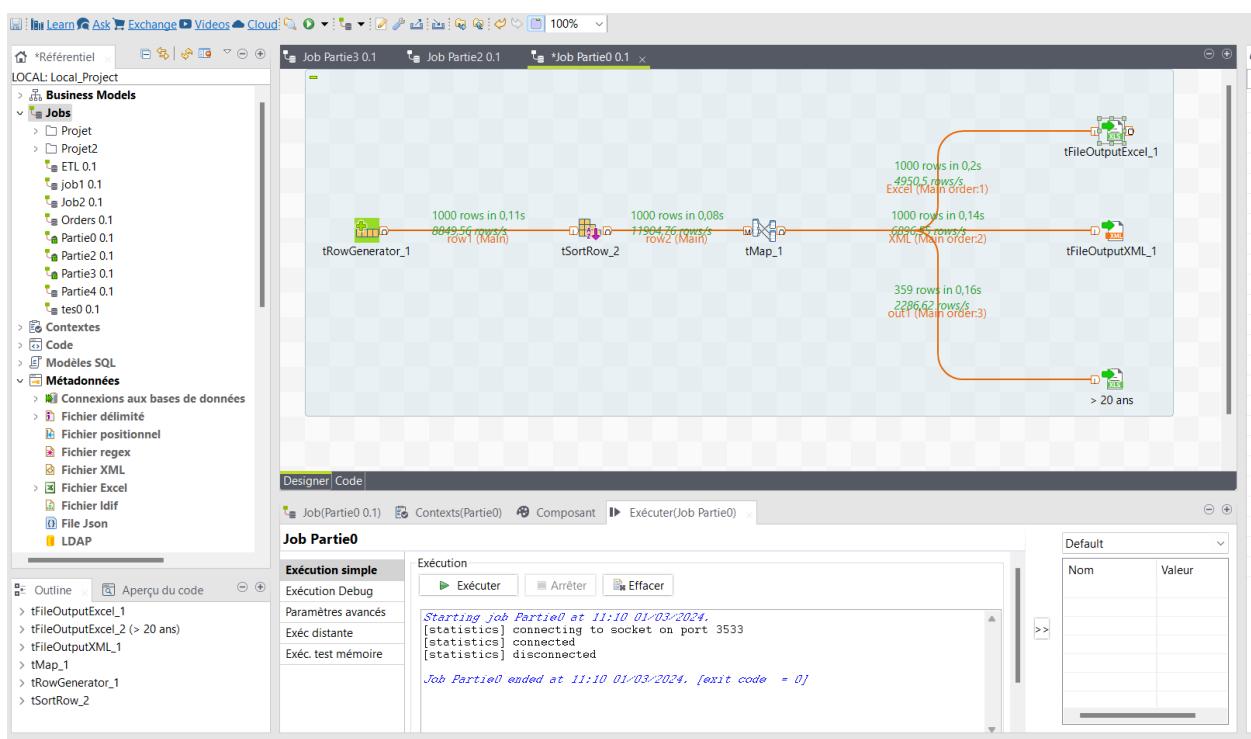
Nous souhaitons également les trier par date de naissance et générer trois fichiers en sortie : un fichier XML et deux fichiers Excel. L'un des deux fichiers contiendra uniquement les personnes âgées de plus de 20 ans.

Voici un aperçu de la configuration :

- Le composant "tRowGenerator" est utilisé pour générer des données.
- Le composant "tSortRow" est utilisé pour trier les données.
- Le composant "tMap" est utilisé pour le traitement et le filtrage des données.
- Les composants "tFileOutputExcel" et "tFileOutputXML" sont utilisés pour exporter les données vers des fichiers Excel et XML respectivement.

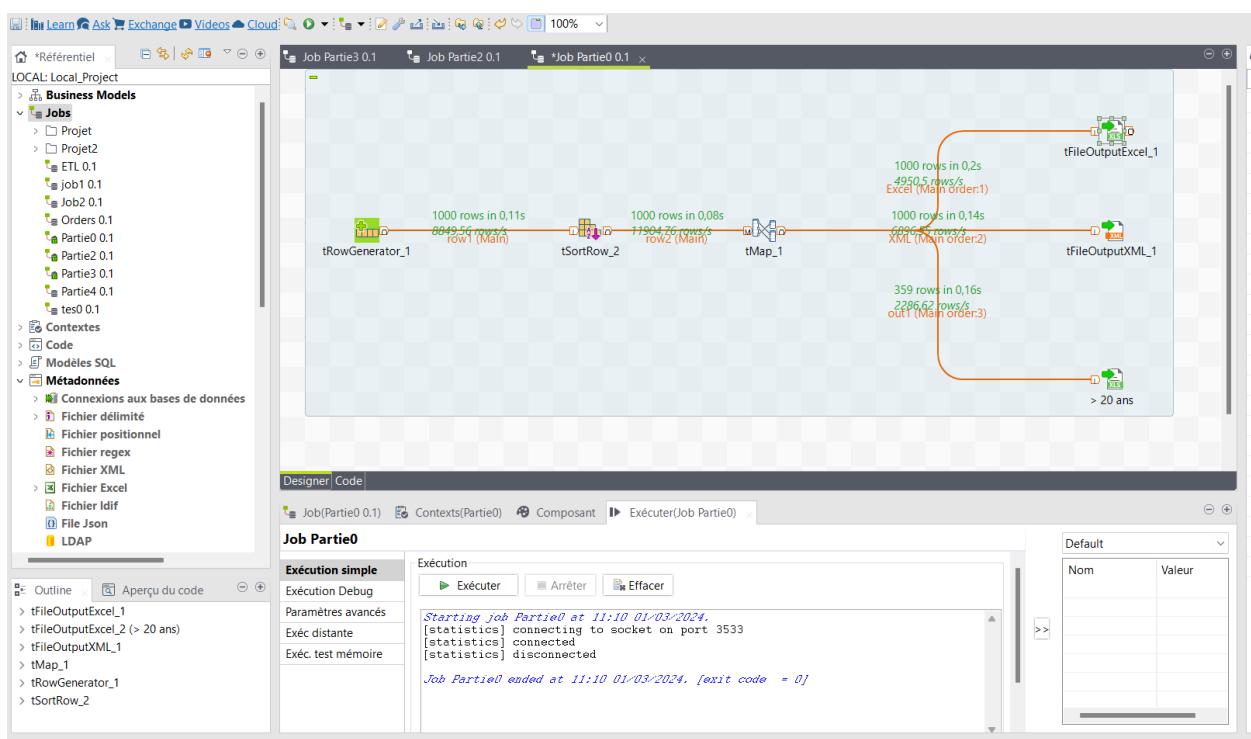
BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



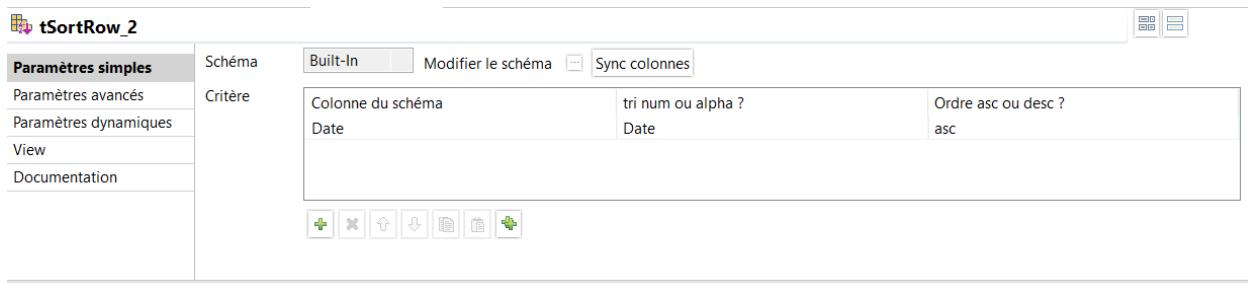
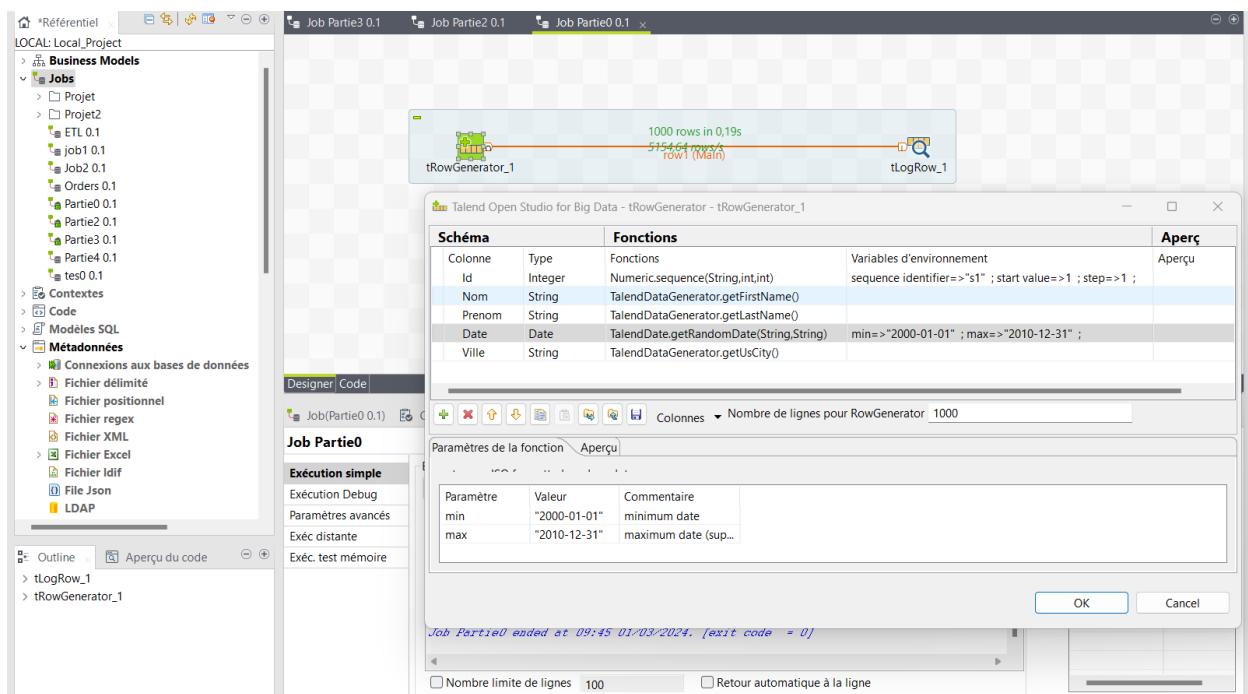
BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



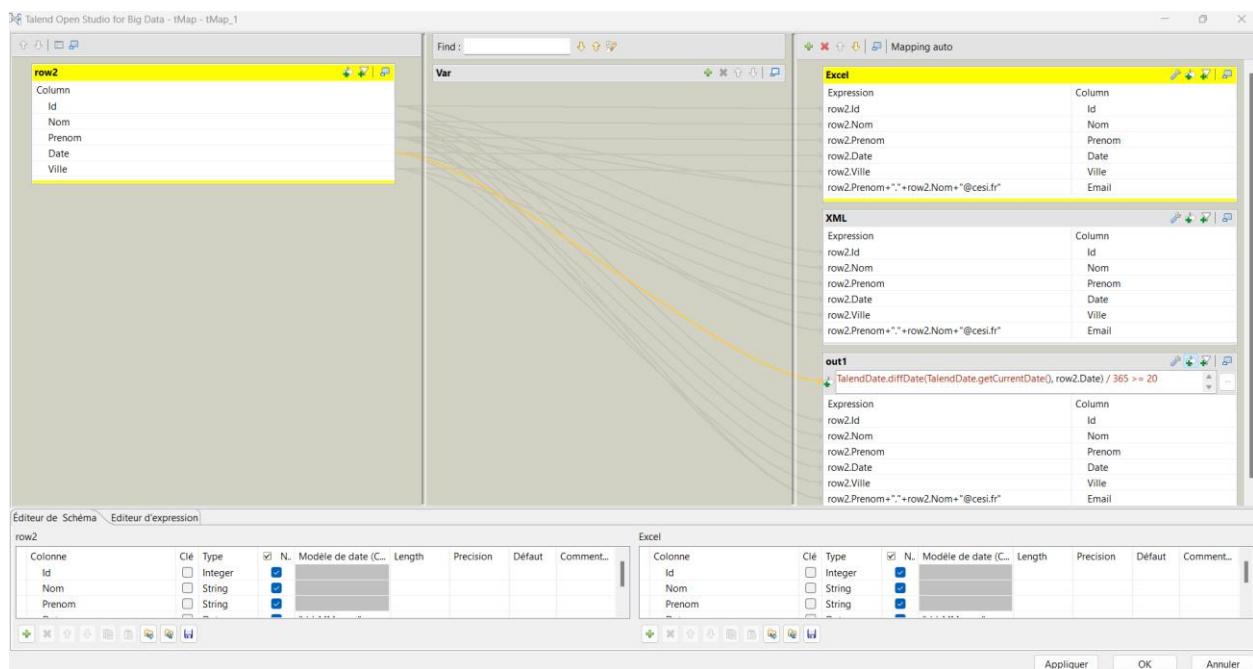
BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES





BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

| A | B | C | D | E | F |
|----|-----|----------|------------|------------|---------------------------|
| Id | Nom | Prenom | Date | Ville | Email |
| 2 | 487 | William | Kennedy | 06-01-2000 | Columbus |
| 3 | 70 | Dwight | Van Buren | 10-01-2000 | Montpelier |
| 4 | 431 | Bill | Hoover | 11-01-2000 | Santa Fe |
| 5 | 78 | Bill | Nixon | 11-01-2000 | Santa Fe |
| 6 | 453 | Harry | Roosevelt | 13-01-2000 | Concord |
| 7 | 497 | Franklin | Grant | 21-01-2000 | Hartford |
| 8 | 111 | Zachary | Harding | 26-01-2000 | Dover |
| 9 | 930 | Abraham | Fillmore | 31-01-2000 | Lansing |
| 10 | 649 | Bill | Roosevelt | 08-02-2000 | Tallahassee |
| 11 | 41 | Herbert | Fillmore | 10-02-2000 | Little Rock |
| 12 | 802 | Calvin | Van Buren | 13-02-2000 | Albany |
| 13 | 241 | Warren | Grant | 22-02-2000 | Baton Rouge |
| 14 | 306 | Ulysses | Jefferson | 23-02-2000 | Salem |
| 15 | 113 | Abraham | Pierce | 23-02-2000 | Helena |
| 16 | 667 | Ulysses | McKinley | 28-02-2000 | Lansing |
| 17 | 859 | Richard | Johnson | 05-03-2000 | Carson City |
| 18 | 565 | Millard | Roosevelt | 06-03-2000 | Nashville |
| 19 | 543 | Millard | Roosevelt | 15-03-2000 | Columbia |
| 20 | 375 | Jimmy | Hayes | 15-03-2000 | Baton Rouge |
| 21 | 293 | William | Cleveland | 29-03-2000 | Salt Lake City |
| 22 | 938 | Millard | Harding | 30-03-2000 | Atlanta |
| 23 | 117 | Zachary | Ford | 31-03-2000 | Albany |
| 24 | 335 | Andrew | Quincy | 03-04-2000 | Cheyenne |
| 25 | 577 | Ronald | Wilson | 09-04-2000 | Harrisburg |
| 26 | 258 | Theodore | Wilson | 10-04-2000 | Columbia |
| 27 | 669 | Harry | Grant | 15-04-2000 | Charleston |
| 28 | 624 | Martin | Reagan | 18-04-2000 | Trenton |
| 29 | 210 | Calvin | McKinley | 20-04-2000 | Raleigh |
| 30 | 956 | Thomas | Johnson | 24-04-2000 | Atlanta |
| 31 | 446 | Dwight | Hayes | 25-04-2000 | Austin |
| 32 | 708 | Bill | Roosevelt | 03-05-2000 | Little Rock |
| 33 | 757 | Calvin | McKinley | 15-05-2000 | Columbus |
| 34 | 945 | Warren | Washington | 19-05-2000 | Austin |
| | | | | | Washington.Warren@cesi.fr |

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<root>
<row>
<Id>487</Id>
<Nom>William</Nom>
<Preénom>Kennedy</Preénom>
<Date>06-01-2000</Date>
<Ville>Columbus</Ville>
<Email>Kennedy.William@cesi.fr</Email>
</row>
<row>
<Id>70</Id>
<Nom>Dwight</Nom>
<Preénom>Van Buren</Preénom>
<Date>10-01-2000</Date>
<Ville>Montpelier</Ville>
<Email>Van.Buren.Dwight@cesi.fr</Email>
</row>
<row>
<Id>431</Id>
<Nom>Bill</Nom>
<Preénom>Hoover</Preénom>
<Date>11-01-2000</Date>
<Ville>Santa Fe</Ville>
<Email>Hoover.Bill@cesi.fr</Email>
</row>
<row>
<Id>78</Id>
<Nom>Bill</Nom>
<Preénom>Nixon</Preénom>
<Date>11-01-2000</Date>
<Ville>Santa Fe</Ville>
<Email>Nixon.Bill@cesi.fr</Email>
</row>
<row>
```

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Partie 2 :

Commencez par télécharger la base de données des ventes que notre client nous a envoyé à travers ce lien : https://drive.google.com/drive/folders/1Vw2BgpVtvXZw85y1nEx1jbdI5gJ_Yul6?usp=sharing

Assurez-vous d'avoir dans le dossier trois fichiers au format csv (*Features data set.csv*, *sales data-set.csv*, *stores data-set.csv*).

La base de données concerne les données de vente, qui peuvent fournir des informations précieuses sur les performances passées, les tendances du marché, les préférences des clients et d'autres aspects commerciaux. Le but de cette partie est d'analyser ces données afin que les entreprises puissent prendre des décisions éclairées sur leur stratégie de vente, leur tarification, leur marketing, etc.

Commençons d'abord par l'extraction des données sources (**ETL**)

- 1- Lancez Talend et créez votre Projet/Workspace.
- 2- Récupérez et chargez les données sources des 3 fichiers CSV. Utilisez l'option "Fichier délimité" pour extraire les données sources. Allez dans le Référentiel, puis accédez à "Métadonnées", et cliquez avec le bouton droit sur "Fichier délimité" afin de charger les 3 fichiers CSV.

Assurez-vous d'avoir récupéré les trois fichiers de données.

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

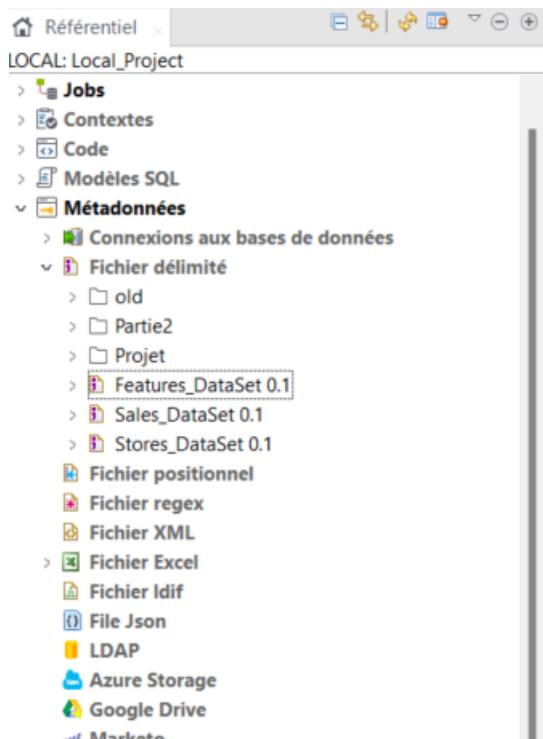
The screenshot shows the Talend Open Studio interface for creating a new job. The main window displays the 'Job Workshop 1.0' configuration.

- Step 1:** The left sidebar shows the project structure under 'Projet_Big_Data'. A red arrow points to the 'Fichiers délimités' section in the 'Métadonnées' category.
- Step 2:** A red box highlights the 'Créer un fichier délimité' option in the 'Fichiers délimités' list.
- Step 3:** A red box highlights the 'Nom' field in the 'Nouveau fichier délimité' dialog, which contains the value 'Stores_DataSet'.
- Step 4:** A red box highlights the 'Objectif' field, containing the text 'Le fichier CSV contenant les données en stock'.
- Step 5:** A red box highlights the 'Next >' button at the bottom of the dialog.
- Step 6:** The right sidebar shows the palette with various components, and a red box highlights the 'Fichier' component.
- Step 7:** A red box highlights the 'Next >' button in the 'Fichier - Etape 2 de 4' dialog.
- Step 8:** The 'Fichier - Etape 3 de 4' dialog is shown. A red box highlights the 'Encodage' dropdown set to 'US-ASCII'. Another red box highlights the 'Séparateur de champs' dropdown set to 'Comma'. A green arrow points from the 'Caractère correspondant' dropdown to the 'En-tête' dropdown, which is set to '1'. A red box highlights the 'Pied de page' checkbox.
- Step 9:** A red box highlights the 'Définir les lignes d'en-tête comme nom de colonne' checkbox.
- Step 10:** A red box highlights the 'Rafraîchir l'aperçu' button.
- Step 11:** A green arrow points up to the 'Aperçu' tab, where a preview of the data is shown. A red box highlights the 'Export en tant que contexte' button.
- Step 12:** A red box highlights the 'Next >' button at the bottom of the dialog.
- Step 13:** The 'Créer/réutiliser un groupe de contextes' dialog is shown. A red box highlights the 'Créer un nouveau contexte dans le référentiel' radio button.
- Step 14:** A red box highlights the 'Next >' button at the bottom of the dialog.
- Step 15:** The 'Fichier - Etape 4 de 4' dialog is shown. A red box highlights the 'Finish' button.

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Refaites la même procédure pour importer les deux autres fichiers CSV. Assurez-vous d'avoir récupéré les trois fichiers de données.



Nous allons maintenant passer à l'étape de traitement (**ETL**)

Pour rappel, le client nous a informé de l'existence de plusieurs factures erronées, des clients en double et plusieurs enregistrements vides !

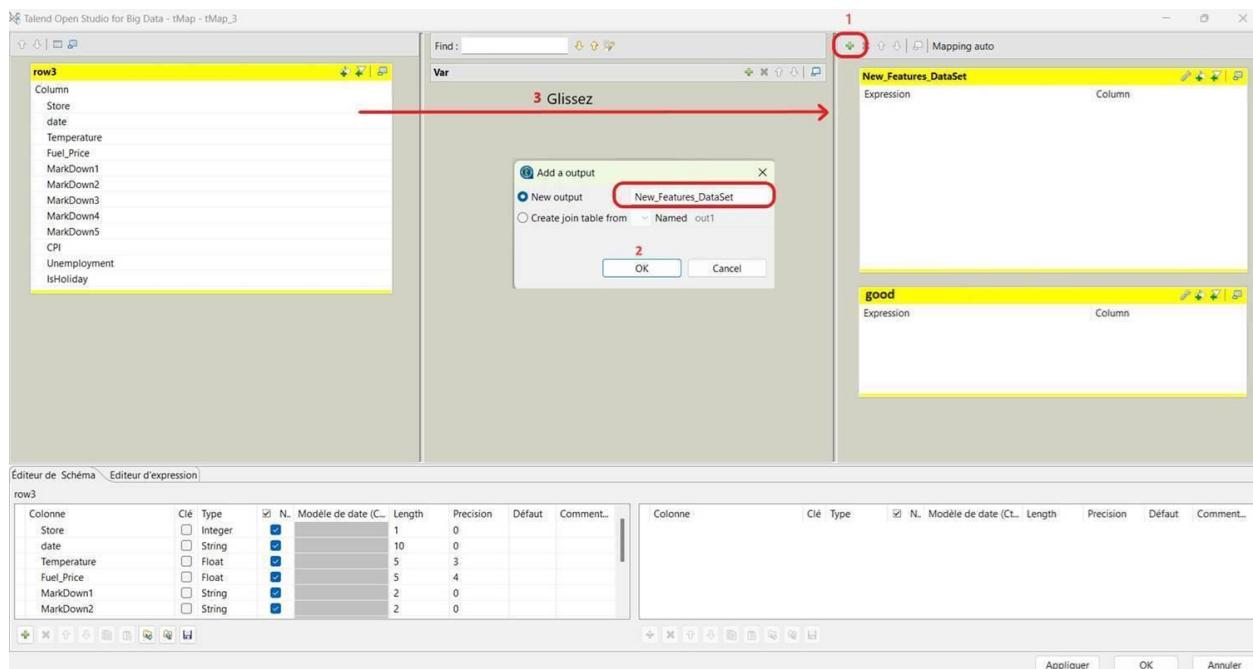
Remarque : Pour ajouter des composants Talend sur le Designer, par exemple "tMap", vous pouvez soit passer par la "Palette", puis dérouler "Transformation" et glisser le composant "tMap" sur le Designer. Vous pouvez également rechercher n'importe quel composant dans la barre de recherche. Une autre

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

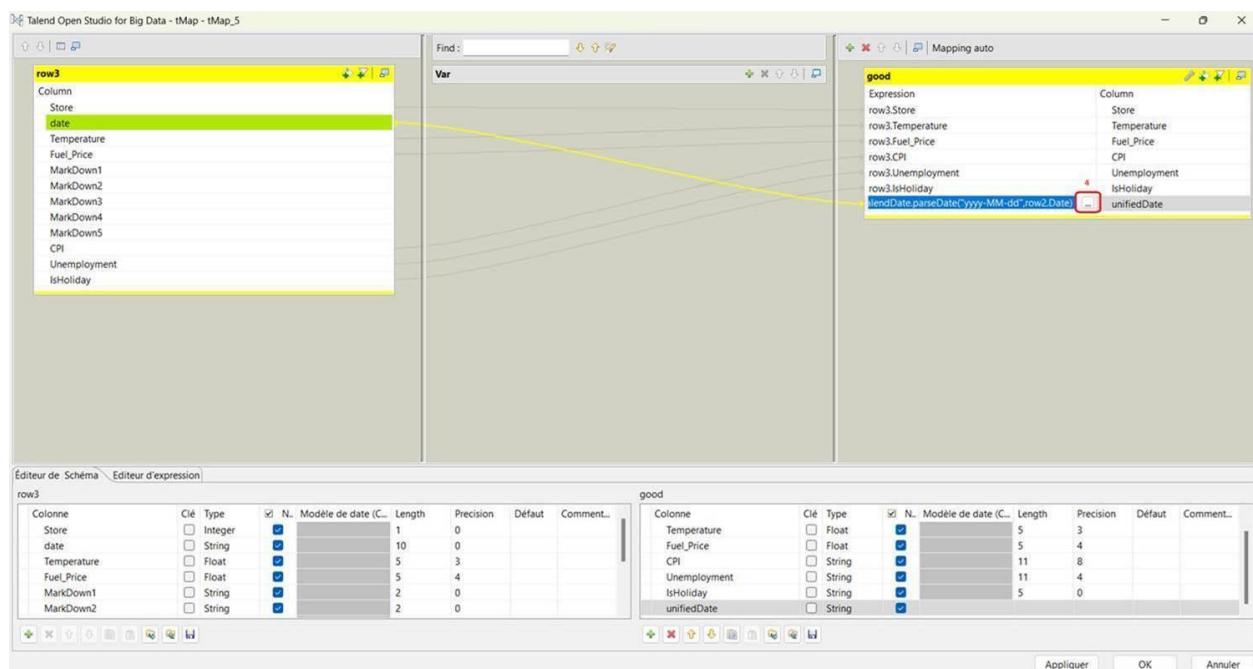
manière de sélectionner un composant est de se mettre dans le Designer, de cliquer sur l'interface et d'écrire la lettre "t" suivie du nom du composant.

- 1- Créez votre premier Job,
- 2- Déposez maintenant le premier fichier « Features_DataSet » sur le Designer (choisissez tFileInputDelimited)
- 3- Commencez par analysez les dates !! que remarquez-vous ?
Format des dates non unifié, elles ont deux formats différents *01-01-2000* et *01/01/2000*
- 4- Rajoutez le composant « tMap » qui nous permettra d'effectuer la majorité des traitements,
- 5- Reliez les deux composants : cliquez avec le bouton droit sur le composant «Features_DataSet» puis cliquez sur « Row » puis sur « Main », et reliez-le au « tMap »
- 6- Double-cliquez sur tMap. La partie de gauche représente les entrées et celle de droite les sorties. Sur la partie droite, cliquez sur « + » et choisissez un nom. Faites glisser les colonnes d'entrée vers la droite (les sorties) (Store, Temperature, Fuel_Price, CPI, Unemployment, IsHoliday, date)
- 7- Trouvez la formule qui nous permettrait d'unifier les dates



BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



- Pour unifier les dates, choisissez une des trois solutions suivantes :

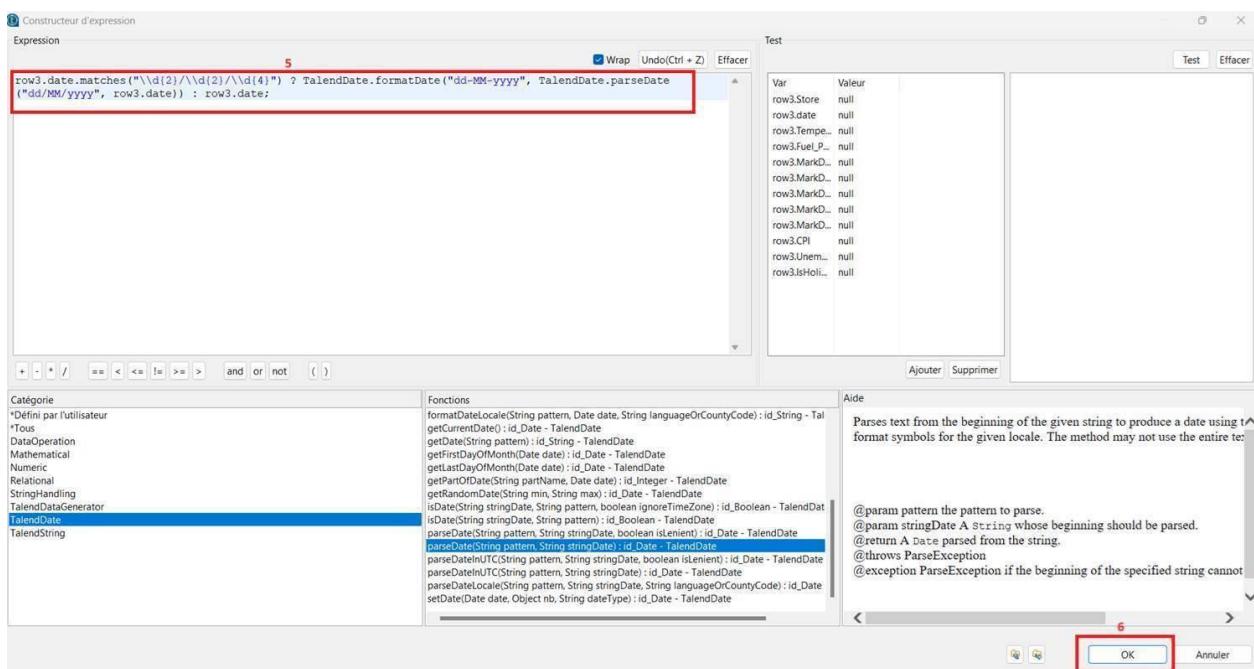
```
row2.Date.matches("\d{2}/\d{2}/\d{4}") ? TalendDate.formatDate("dd-MM-yyyy", TalendDate.parseDate("dd/MM/yyyy", row2.Date)) : row2.Date;
```

Ou

```
row2.Date.matches("\d{2}/\d{2}/\d{4}") ? TalendDate.formatDate("MM-dd-yyyy", TalendDate.parseDate("MM/dd/yyyy", row2.Date)) :
TalendDate.formatDate("dd-MM-yyyy", TalendDate.parseDate("dd-MM-yyyy", row2.Date));
```

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



The screenshot shows the Talend Open Studio interface for a tMap component. The left panel displays the schema for 'row3' with columns: Column, Store, date, Temperature, Fuel_Price, MarkDown1, MarkDown2, MarkDown3, MarkDown4, MarkDown5, CPI, Unemployment, IsHoliday. The right panel shows the 'Mapping auto' dialog with a single mapping entry:

| Colonne | Expression | Column |
|---------|--|-------------|
| date | 'row3.date.matches("\\d{2}/\\d{2}/\\d{4}") ? TalendDate.formatDate("dd-MM-yyyy", TalendDate.parseDate("dd/MM/yyyy", row3.date)) : row3.date' | unifiedDate |

Below the mapping dialog are two tables:

| Colonne | Clé | Type | N° | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|-------------|--------------------------|---------|-------------------------------------|-----------------------|--------|-----------|--------|------------|
| Store | <input type="checkbox"/> | Integer | <input checked="" type="checkbox"/> | | 1 | 0 | | |
| date | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 10 | 0 | | |
| Temperature | <input type="checkbox"/> | Float | <input checked="" type="checkbox"/> | | 5 | 3 | | |
| Fuel_Price | <input type="checkbox"/> | Float | <input checked="" type="checkbox"/> | | 5 | 4 | | |
| MarkDown1 | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 2 | 0 | | |
| MarkDown2 | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 2 | 0 | | |

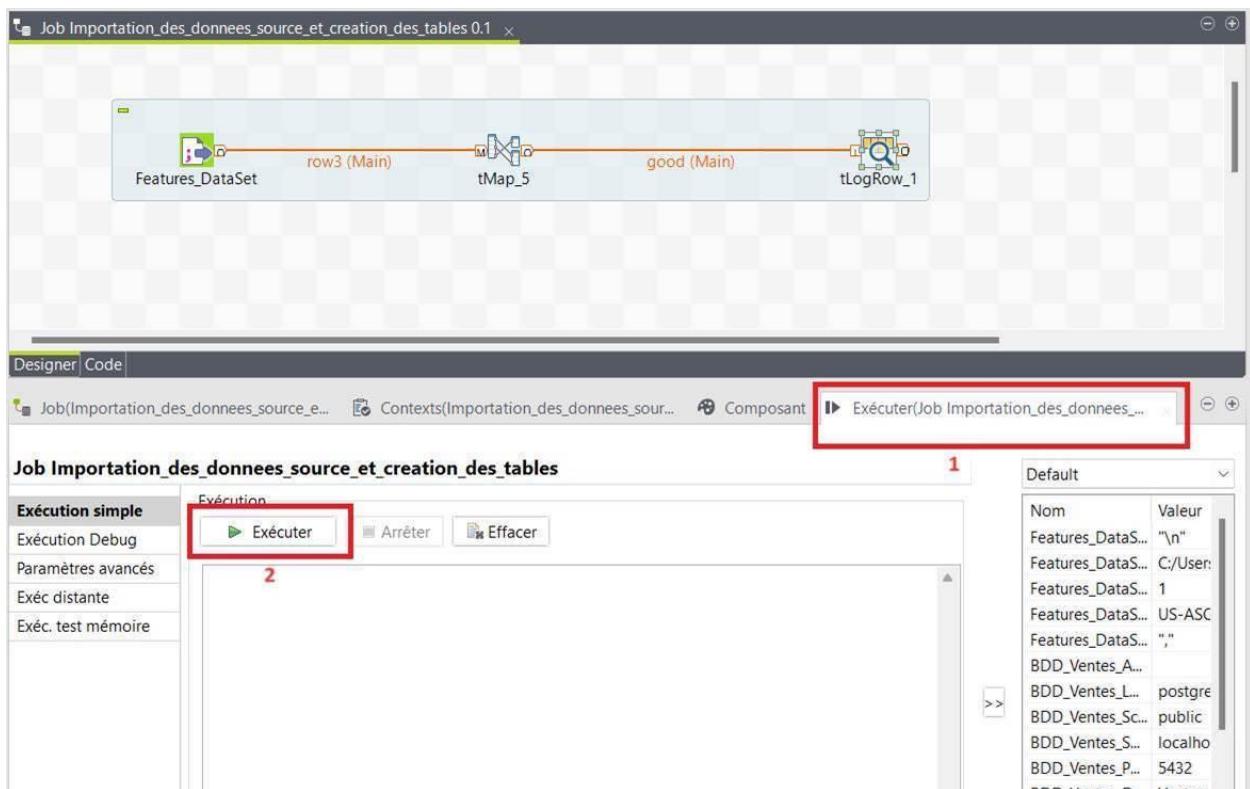
| Colonne | Clé | Type | N° | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|--------------|--------------------------|--------|-------------------------------------|-----------------------|--------|-----------|--------|------------|
| Temperature | <input type="checkbox"/> | Float | <input checked="" type="checkbox"/> | | 5 | 3 | | |
| Fuel_Price | <input type="checkbox"/> | Float | <input checked="" type="checkbox"/> | | 5 | 4 | | |
| CPI | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 11 | 8 | | |
| Unemployment | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 11 | 4 | | |
| IsHoliday | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 5 | 0 | | |
| unifiedDate | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 5 | 0 | | |

The 'OK' button at the bottom right of the mapping dialog is highlighted with a red box.

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

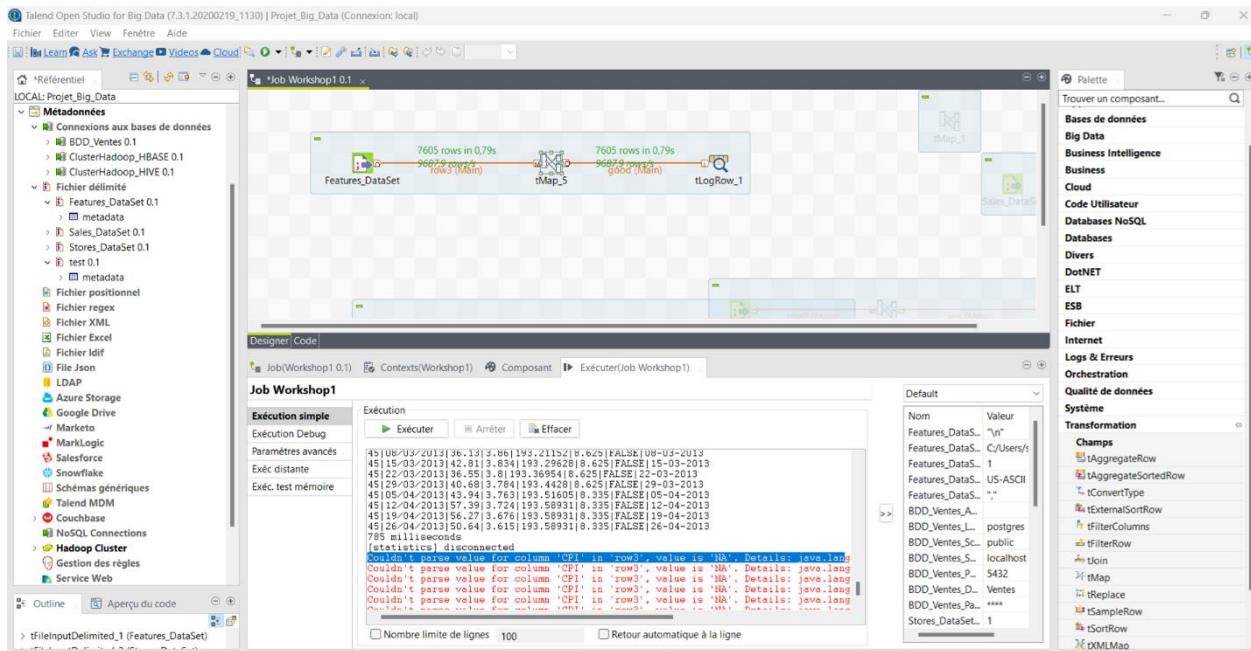
- 8- Ajoutez le composant « tLogRow » qui nous permettra d'afficher le résultat en sortie. Cliquez avec le bouton droit sur le « tMap » puis sélectionnez « Row » et choisissez le nom de sortie que vous avez mentionné dans le tMap puis le reliez au « tLogRow »
- 9- Exécutez le Job pour voir le résultat



- 10- Que remarquez-vous ?

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

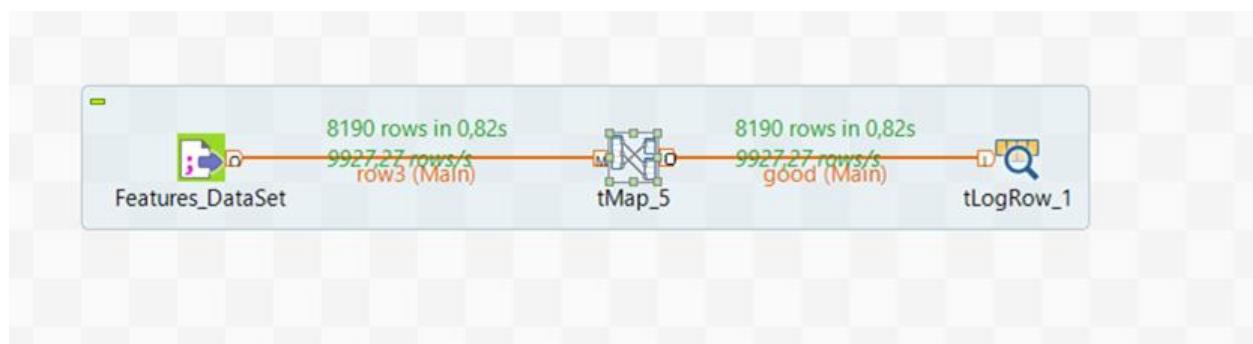
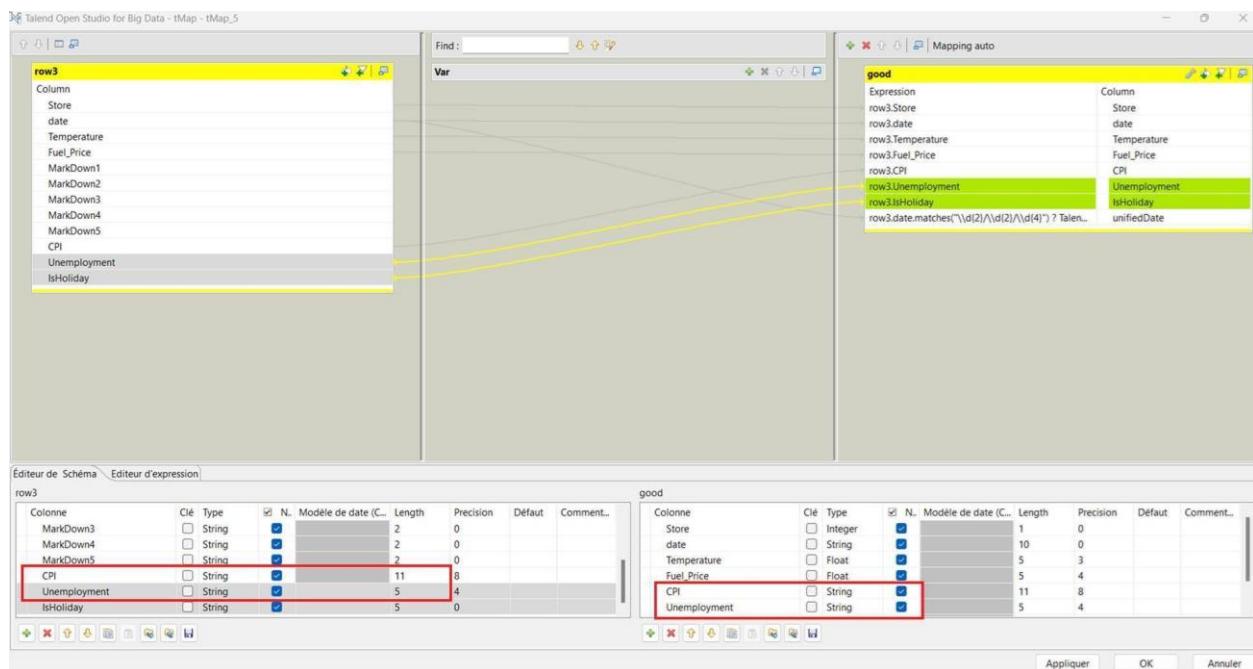


Remarquez bien que le fichier contient 8191 éléments, mais pour l'instant, nous n'en avons récupéré que 7605 ! Voyons ce qui a causé ce problème. Regardons bien l'erreur en rouge !

Remarquez bien que les colonnes "CPI" et "Unemployment" sont au format Float ! Il faut les modifier en "String".

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

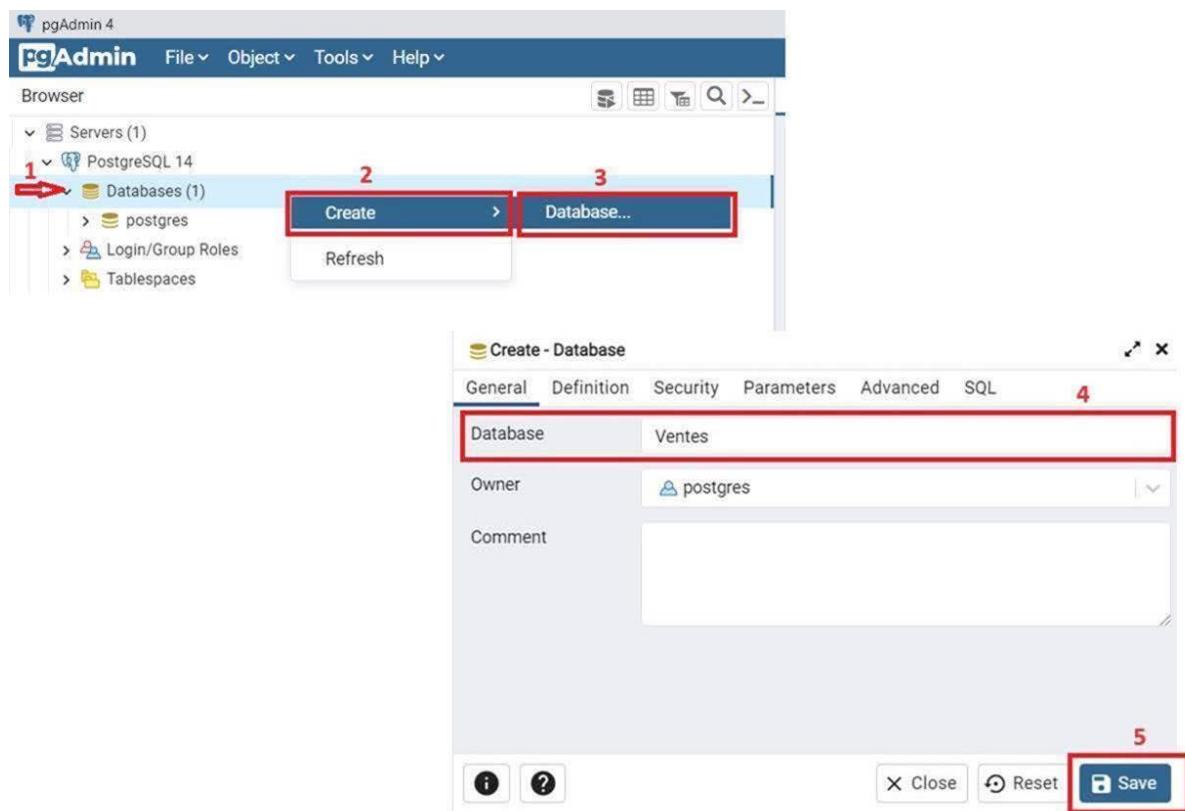


11- Une fois que vous avez récupéré les 8190 éléments, procéderons maintenant à l'alimentation des tables :

- a- Préparons la nouvelle base de données qui accueillera les données du client
- b- Sous le SGBD PostgreSQL, Lancez « PgAdmin 4 » puis cliquez avec le bouton droit sur « Databases », puis sur « Create », puis sur « Database » et donnez-lui un nom par exemple « Ventes ». Enfin validez en cliquant sur « Save »

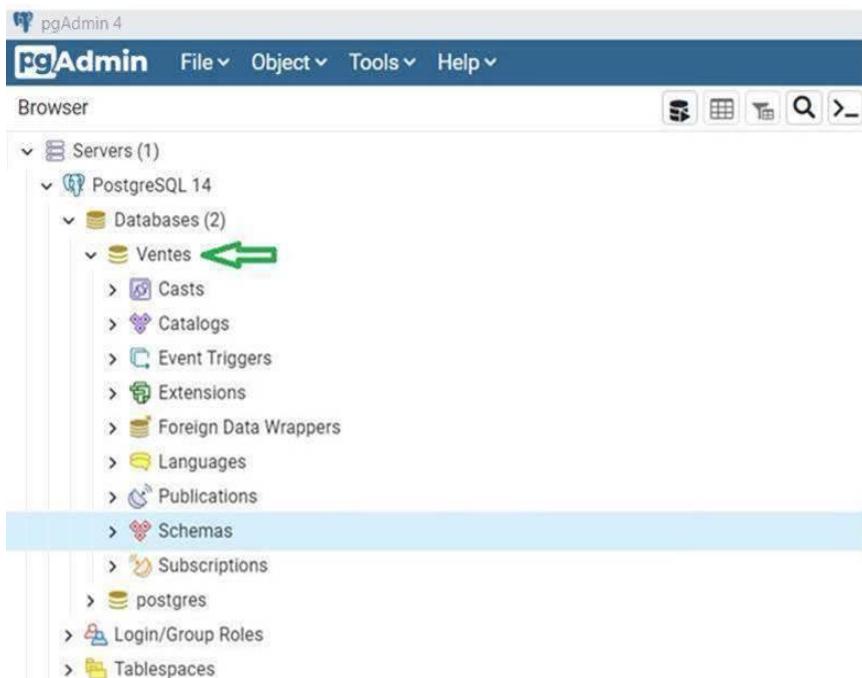
BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

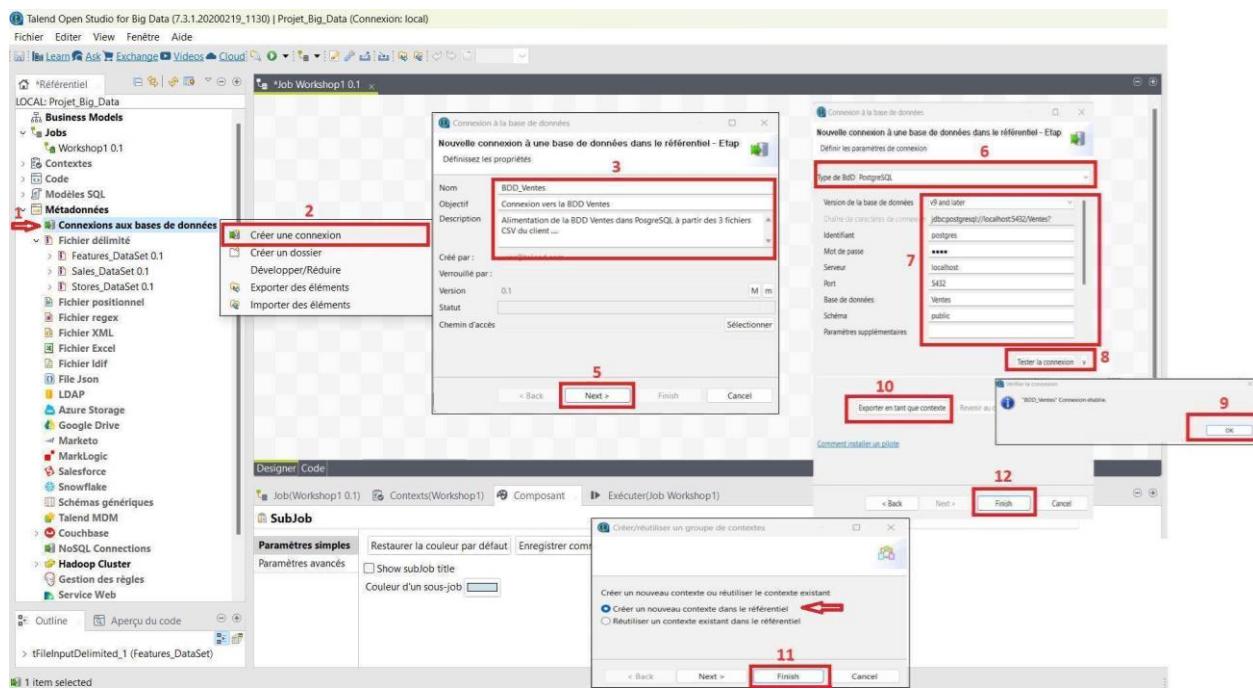


12- Passons maintenant à connecter Talend vers la BDD Ventes.

Toujours dans le Référentiel, cliquez avec le bouton droit sur "Connexions aux bases de données", puis sur "Créer une connexion". Remplissez les informations concernant la base de données et cliquez sur "Next". Choisissez "PostgreSQL" comme SGBD, puis complétez les différents champs relatifs à la base de données "Ventes" créée au préalable. Cliquez ensuite sur "Tester la connexion" pour vous assurer que la connexion est établie. Si ce n'est pas le cas, vérifiez les informations que vous avez saisis. Ensuite, cliquez sur "Exporter en tant que contexte", cochez "Créer un nouveau contexte dans le référentiel" et cliquez sur "Finish".

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



- 13- Vous allez maintenant ajouter un nouveau composant qui vous permettra de créer et d'alimenter les nouvelles tables : tDBOutput PostgreSQL. Vous avez deux options :
- Soit vous supprimez le composant « tLogRow » et depuis le « tMap », vous reliez la sortie vers tDBOutput,
 - Soit vous relier directement le composant « tLogRow » au « tDBOutput » .

Remarque : Afin de créer la table Feature, il va falloir modifier la taille de du champs « unemployment » de 5 à 11

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Schéma de Features_DataSet

Features_DataSet

| Colonne | Clé | Type | N. | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|--------------|--------------------------|--------|-------------------------------------|-----------------------|--------|-----------|--------|------------|
| MarkDown2 | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 2 | 0 | | |
| MarkDown3 | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 2 | 0 | | |
| MarkDown4 | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 2 | 0 | | |
| MarkDown5 | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 2 | 0 | | |
| CPI | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 11 | 8 | | |
| Unemployment | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 11 | 4 | | |
| IsHoliday | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 5 | 0 | | |

Actions :         

OK Cancel

14- Refaites la manip, afin de récupérer les trois fichiers et d'alimenter la BDD
Attention aux enregistrements vides et aux doublons

Talend Open Studio for Big Data (7.3.1.20200219_1130) | Projet_Big_Data (Connexion: local)

Fichier Editeur View Fenêtre Aide

Job Importation_des_donnees_source_et_creation_des_tables 0.1

LOCAL: Projet_Big_Data

- Jobs
 - ETL 0.1
 - Importation_des_donnees_so
- Contextes
- Code
- Modèles SQL
- Métadonnées
 - Connexions aux bases de données
 - BDD_Ventes 0.1
 - ClusterHadoop_HBASE 0.1
 - ClusterHadoop_HIVE 0.1
 - Ventes 0.1
 - Fichier délimité
 - Fichier positionnel
 - Fichier regex
 - Fichier XML
 - Fichier Excel
 - Fichier Idif
 - File Json
 - LDAP
 - Azure Storage
 - Google Drive
 - Marketo
 - MarkLogic
 - Salesforce
 - Snowflake
 - Schémas génériques
 - Talend MDM
 - Couchbase

Palettes

- Trouver un composant...
- Favoris
- Récemment utilisé
- Applications Métier
- Bases de données
 - DB Common
 - DB Specifics
 - tCreateTable
 - tParseRecordSet

Big Data

- Business Intelligence
- Business
- Cloud
- Code Utilisateur
- Databases NoSQL
- Databases
- Divers
- DotNET
- ELT
- ESB
- Fichier
- Internet
- Logs & Erreurs
- Orchestration
- Qualité de données
- Système
- Transformation
- Unstructured
- XML
- Misc

Designer

Job ETL 0.1

Exécution simple

Exécution Exécuter Arrêter Effacer

Starting job ETL at 14:53 24/11/2023.

Execution Contexts(ETL)

Composant

Exécuter(Job ETL)

Job ETL

Execution simple

Execution Debug

Paramètres avancés

Exéc. distante

Exéc. test mémoire

Default

| Nom | Valeur |
|-----------------|-----------|
| Ventes_Port | 5432 |
| Ventes_Schema | public |
| Ventes_Password | *** |
| Ventes_Server | localhost |
| Ventes_Login | postore |

Outline Aperçu

Diagramme Talend ETL (Job ETL 0.1) :

```

graph TD
    BD[Ventes] --> OnSubjobOk1[OnSubjobOk]
    OnSubjobOk1 --> FeaturesDataSet1[Features DataSet]
    FeaturesDataSet1 --> tMap1[tMap_1]
    tMap1 --> Caracteristiques[Caractéristiques]
    Caracteristiques --> OnSubjobOk2[OnSubjobOk]
    OnSubjobOk2 --> SalesDataSet1[Sales DataSet]
    SalesDataSet1 --> tMap3[tMap_3]
    tMap3 --> SalesDataSet2[Sales DataSet]
    SalesDataSet2 --> OnSubjobOk3[OnSubjobOk]
    OnSubjobOk3 --> StoresDataSet1[Stores DataSet]
    StoresDataSet1 --> tMap2[tMap_2]
    tMap2 --> tUniqRow1[tUniqRow_1]
    tUniqRow1 --> StoresDataSet2[Stores DataSet]
  
```

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Talend Open Studio for Big Data - tMap - tMap_1

row3

| Colonne | Clé | Type | N. | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|--------------|-----|---------|----|-----------------------|--------|-----------|--------|------------|
| Store | | Integer | | | 1 | 0 | | |
| date | | String | | | 10 | 0 | | |
| Temperature | | Float | | | 5 | 3 | | |
| Fuel_Price | | Float | | | 5 | 4 | | |
| Markdown1 | | String | | | 2 | 0 | | |
| Markdown2 | | String | | | 2 | 0 | | |
| Markdown3 | | | | | | | | |
| Markdown4 | | | | | | | | |
| Markdown5 | | | | | | | | |
| CPI | | | | | | | | |
| Unemployment | | | | | | | | |
| IsHoliday | | | | | | | | |

good

| Expression | Column |
|---|----------------|
| row3.Store | Magasin |
| row3.Temperature | Temperature |
| row3.Fuel_Price | Prix_carburant |
| row3.CPI | CPI |
| row3.Unemployment | Taux_chomage |
| row3.IsHoliday | Est_conge |
| row3.date.matches("\d{2}/\d{2}/\d{4}") ? Talen... | Date |

Editeur de Schéma \ Editeur d'expression

row3

| Colonne | Clé | Type | N. | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|-------------|-----|---------|----|-----------------------|--------|-----------|--------|------------|
| Store | | Integer | | | 1 | 0 | | |
| date | | String | | | 10 | 0 | | |
| Temperature | | Float | | | 5 | 3 | | |
| Fuel_Price | | Float | | | 5 | 4 | | |
| Markdown1 | | String | | | 2 | 0 | | |
| Markdown2 | | String | | | 2 | 0 | | |

good

| Colonne | Clé | Type | N. | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|----------------|-----|---------|----|-----------------------|--------|-----------|--------|------------|
| Magasin | | Integer | | | 1 | 0 | | |
| Temperature | | Float | | | 5 | 3 | | |
| Prix_carburant | | Float | | | 5 | 4 | | |
| CPI | | String | | | 11 | 8 | | |
| Taux_chomage | | String | | | 11 | 4 | | |
| Est_conge | | String | | | 5 | 0 | | |

Appliquer OK Annuler

Talend Open Studio for Big Data - tMap - tMap_3

row6

| Colonne | Clé | Type | N. | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|--------------|-----|---------|----|-----------------------|--------|-----------|--------|------------|
| Column | | Integer | | | 1 | 0 | | |
| Store | | Integer | | | 1 | 0 | | |
| Dept | | String | | | 10 | 0 | | |
| Date | | Float | | | 8 | 3 | | |
| Weekly_Sales | | String | | | 5 | 0 | | |
| IsHoliday | | | | | | | | |

OutSales

| Expression | Column |
|---|----------------------|
| row6.Store | Magasin |
| row6.Dept | Rayon |
| rows.Date.matches("\d{2}/\d{2}/\d{4}") ? Talen... | Date |
| rows.Weekly_Sales | Ventes_hebdomadaires |
| row6.IsHoliday | Vacances |

Editeur de Schéma \ Editeur d'expression

row6

| Colonne | Clé | Type | N. | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|--------------|-----|---------|----|-----------------------|--------|-----------|--------|------------|
| Store | | Integer | | | 1 | 0 | | |
| Dept | | Integer | | | 1 | 0 | | |
| Date | | String | | | 10 | 0 | | |
| Weekly_Sales | | Float | | | 8 | 3 | | |
| IsHoliday | | String | | | 5 | 0 | | |

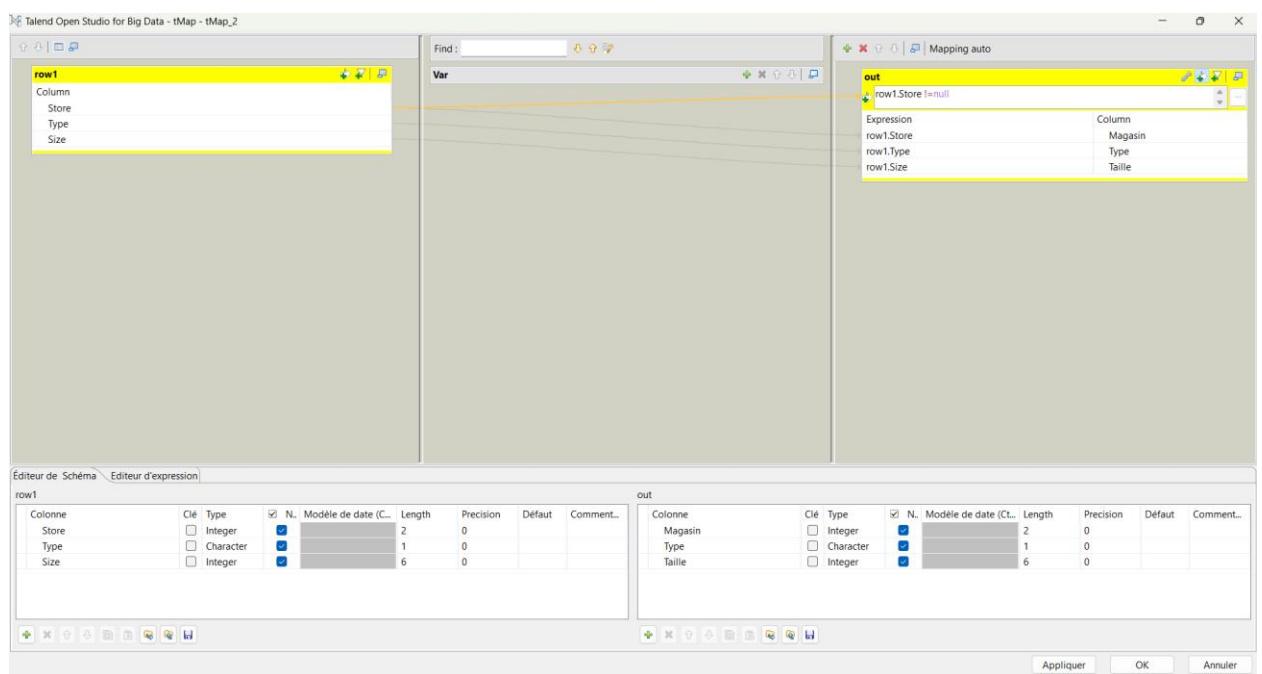
OutSales

| Colonne | Clé | Type | N. | Modèle de date (C...) | Length | Precision | Défaut | Comment... |
|----------------------|-----|---------|----|-----------------------|--------|-----------|--------|------------|
| Magasin | | Integer | | | 1 | 0 | | |
| Rayon | | Integer | | | 1 | 0 | | |
| Date | | String | | | 10 | 0 | | |
| Ventes_hebdomadaires | | Float | | | 8 | 3 | | |
| Vacances | | String | | | 5 | 0 | | |

Appliquer OK Annuler

BLOC : BIG DATA

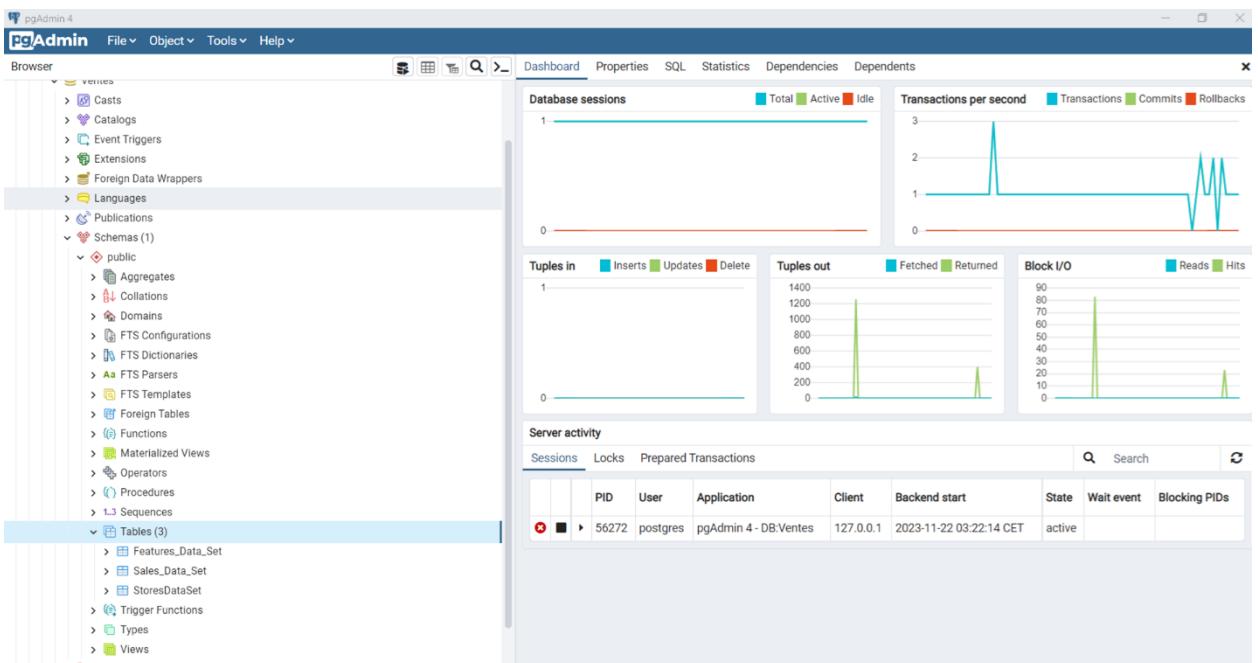
WORKSHOP : INTEGRATION DE DONNEES



15- Vérifiez la création des trois tables dans PostgreSQL

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



Partie 3 :

Téléchargez la 2^{ème} base de données à travers ce lien :

https://drive.google.com/drive/folders/1KyQBu4N8tbNslnamrch9mpnDaPoAc6Xe?usp=drive_link

Assurez-vous d'avoir dans le dossier les deux fichiers au format csv (*Customer.csv & Product.csv*)

Notre second client souhaite récupérer le chiffre d'affaire par produit et par client afin d'anticiper l'agencement des produits par région et de minimiser les pertes tout en récompensant leurs meilleurs clients

Nous allons d'abord analyser sa requête puis construire le schéma décisionnel, et enfin passer par la suite à l'analyse des données

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Commençons par les besoins du client. Pour construire le schéma décisionnel, le choix du modèle en étoile sera privilégié, étant donné que nous manipulons des données provenant d'un seul domaine métier, en l'occurrence les ventes.

- 1- Elaborez le schéma décisionnel ? (Les dimensions, la table de faits et les mesures.

Remarque : la dimension temps est indispensable dans la majorité des cas.

Le besoin : Le besoin principal du client est de comprendre et d'analyser le chiffre d'affaires (CA) généré par les ventes, ce qui constitue une métrique cruciale pour évaluer la performance commerciale de l'entreprise. Pour approfondir cette analyse, il serait également pertinent d'inclure les 10 meilleurs clients en termes de chiffre d'affaires, ainsi que les quantités vendues par produit.

Pour répondre à ces besoins, nous devons structurer notre modèle de données de manière à pouvoir explorer et analyser efficacement les ventes. Nous identifierons donc plusieurs dimensions clés qui seront utilisées pour analyser les ventes :

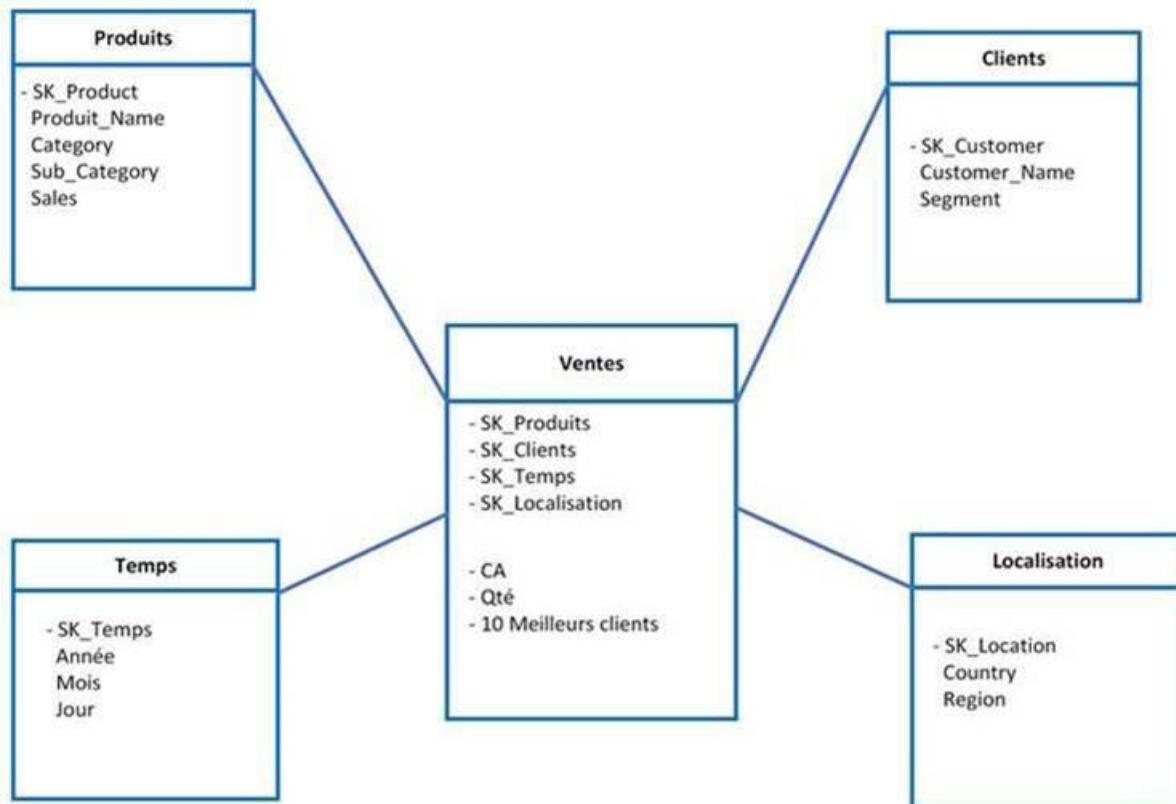
1. Produits : Cette dimension nous permettra d'analyser les performances de vente de chaque produit individuel. Elle inclura des informations telles que le nom du produit, la catégorie, le prix de vente, etc.
2. Clients : Cette dimension nous permettra d'analyser les ventes en fonction des clients. Nous pourrons ainsi identifier les clients les plus rentables, leur comportement d'achat, etc.
3. Régions : Cette dimension nous permettra d'analyser les ventes par région géographique. Cela pourrait inclure des informations sur les ventes par pays, par ville, par région, etc.
4. Temps : Cette dimension nous permettra d'analyser les tendances de vente au fil du temps. Nous pourrons ainsi observer les variations saisonnières, les cycles de vente, etc.

Pour stocker les mesures ou les faits de notre modèle, nous utiliserons une table de faits nommée "Ventes". Cette table contiendra des informations telles que le CA, les quantités vendues, etc. Chaque enregistrement dans cette table représentera une analyse de vente individuelle.

Ci-dessous, le modèle dimensionnel du l'activité « Vente » :

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



Vous allez maintenant analyser les données sources, puis commencez par alimenter d'abord les différentes dimensions. Ce n'est qu'une fois que toutes les dimensions auront été créées que vous passerez à l'étape de la création de la table de faits, qui accueillera l'ensemble des clés de substitution des dimensions ainsi que les mesures.

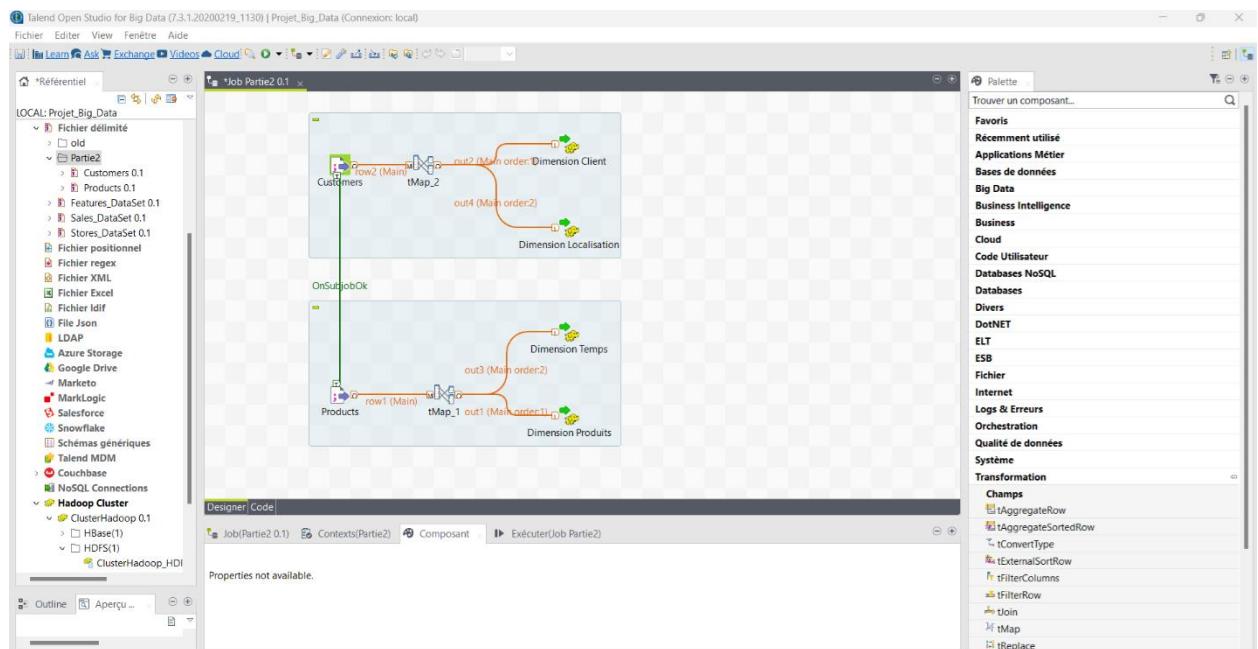
Voici quelques indications à suivre :

1. Sur Talend, commencez par récupérer les données sources sous forme de fichier délimité.
2. Ensuite, créez un Job et placez sur le Designer les deux fichiers Customers et Products. Vous pouvez utiliser les composants tMap, tHDFS(Output/Input), et tAggregateRow.
3. Alimentez d'abord les différentes dimensions, puis la table de faits. Pour la dimension temps, agrégez la date en jour, mois et année pour une analyse plus significative. Vous avez à votre disposition une centaine de fonctions dans le composant tMap pour réaliser cette tâche.

BLOC : BIG DATA

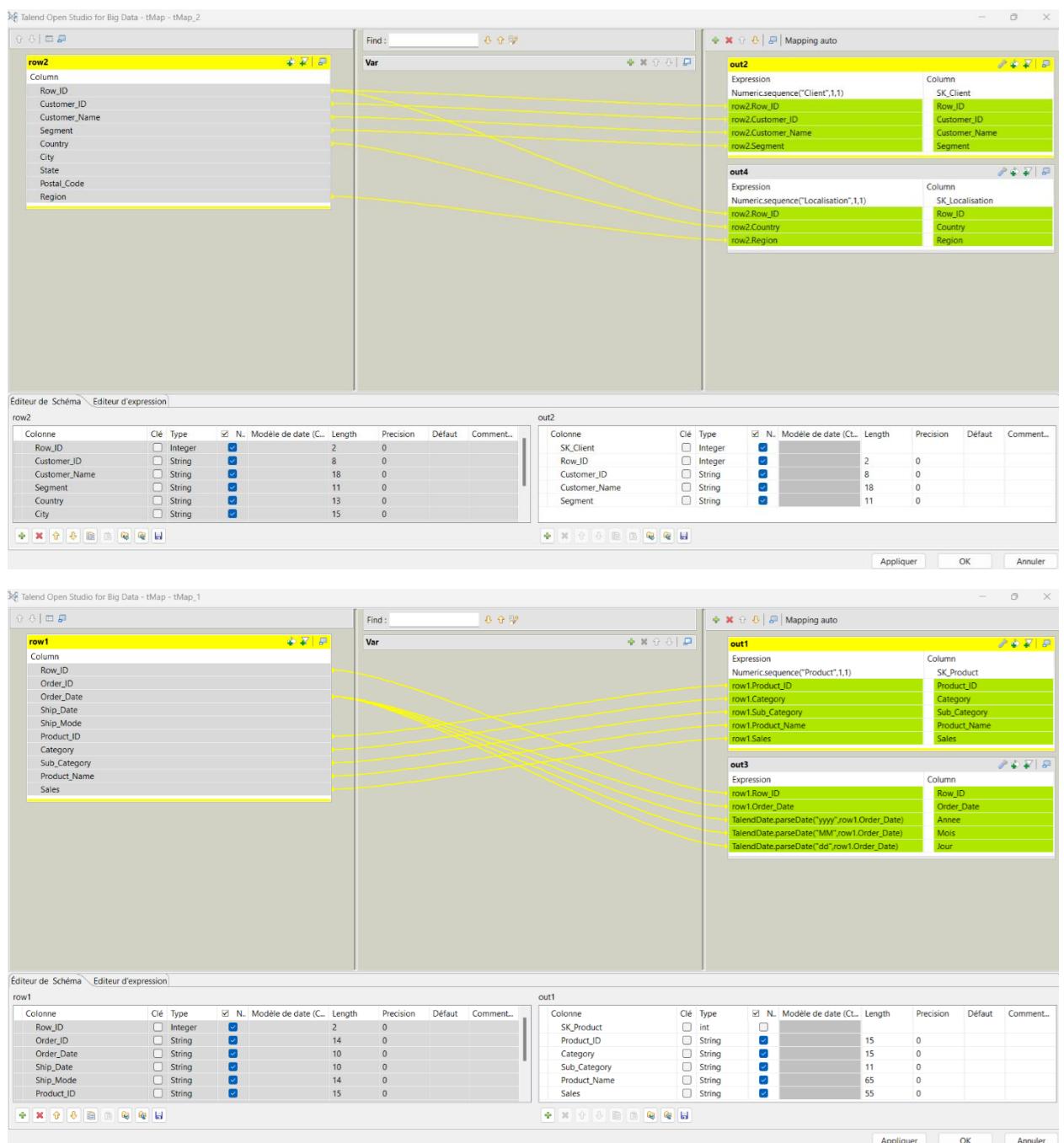
WORKSHOP : INTEGRATION DE DONNEES

4. Assurez-vous de vérifier l'existence des données dans le Datalake dans HDFS après l'exécution du Job.



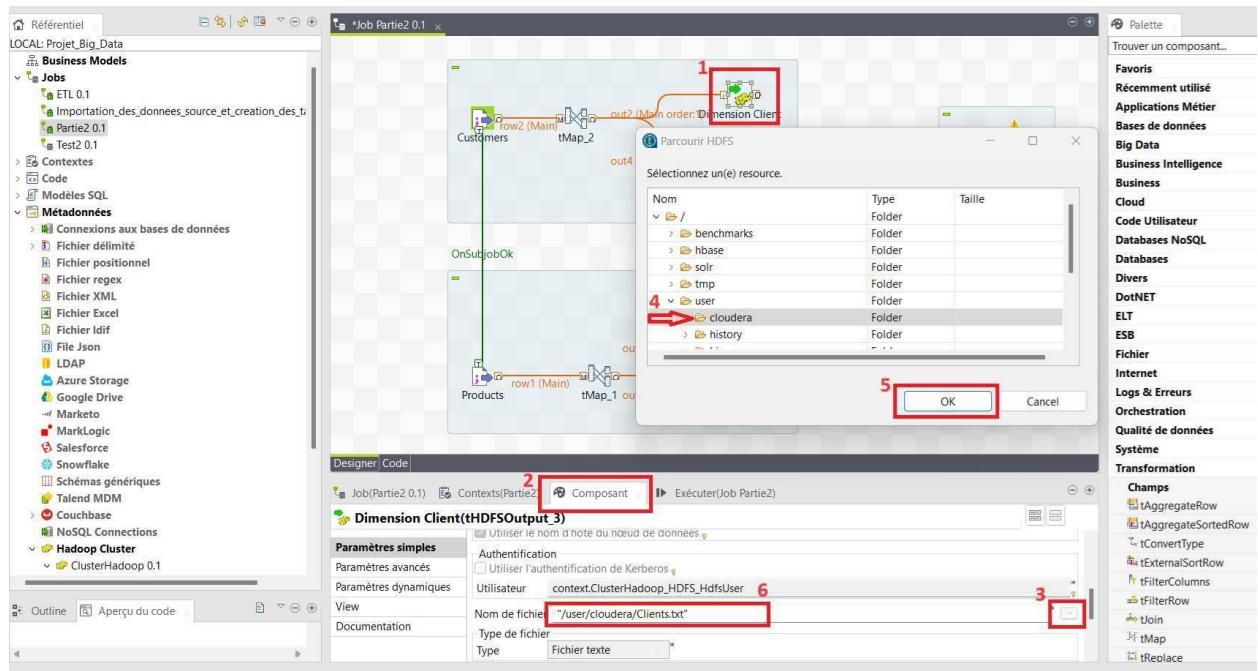
BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



Refaites la même manip pour les 3 autres composants THDFSOutPut à savoir Localisation, Temps et Produits

| | |
|----------------|---------------------------------|
| Nom de fichier | /user/cloudera/Localisation.txt |
| Nom de fichier | /user/cloudera/Temps.txt |
| Nom de fichier | /user/cloudera/Produits.txt |

Remarque & recommandation : Si vous exécutez plusieurs fois le job, vous risquez probablement de rencontrer une erreur ! Pour résoudre ce problème, vous avez deux options :

- Soit vous renommez à chaque fois le nom du fichier de destination.
- Soit, dans la partie "Action", vous modifiez le paramètre par défaut qui est "Create". Vous pouvez le changer soit en "Ecraser" pour remplacer l'ancien fichier, soit en "Ecrire après" si vous souhaitez compléter l'ancien fichier avec de nouvelles données.

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Nom de fichier : "/user/cloudera/Clients.txt"

Type de fichier : Fichier texte *

Action : Ecraser

Séparateur de champ : ,

Encodage : UTF-8

Compression : Compresser les données

Inclure l'en-tête

Find : Var

Mapping auto

| row2 | out2 |
|---------------|--------------------------------|
| Column | Expression |
| Row_ID | Numeric.sequence("Client",1,1) |
| Customer_ID | row2.Row_ID |
| Customer_Name | row2.Customer_ID |
| Segment | row2.Customer_Name |
| Country | row2.Segment |
| City | |
| State | |
| Postal_Code | |
| Region | |

| row2 | out2 | | | |
|---------------|--|---|--|--|
| Colonne | Colonne | | | |
| Row_ID | Clé | | | |
| Customer_ID | Type | | | |
| Customer_Name | <input type="checkbox"/> Integer | | | |
| Segment | <input type="checkbox"/> String | | | |
| Country | <input type="checkbox"/> String | | | |
| City | <input type="checkbox"/> String | | | |
| | <input checked="" type="checkbox"/> N. | | | |
| | Modèle de date (C..) | | | |
| | Length | | | |
| | Precision | | | |
| | Défaut | | | |
| | Comment.. | | | |
| Row_ID | 2 | 0 | | |
| Customer_ID | 8 | 0 | | |
| Customer_Name | 18 | 0 | | |
| Segment | 11 | 0 | | |
| Country | 13 | 0 | | |
| City | 15 | 0 | | |

| out2 | | | | |
|---------------|--|---|--|--|
| Colonne | | | | |
| SK_Client | Clé | | | |
| Row_ID | Type | | | |
| Customer_ID | <input type="checkbox"/> Integer | | | |
| Customer_Name | <input type="checkbox"/> String | | | |
| Segment | <input type="checkbox"/> String | | | |
| | <input checked="" type="checkbox"/> N. | | | |
| | Modèle de date (C..) | | | |
| | Length | | | |
| | Precision | | | |
| | Défaut | | | |
| | Comment.. | | | |
| SK_Client | 2 | 0 | | |
| Row_ID | 8 | 0 | | |
| Customer_ID | 18 | 0 | | |
| Customer_Name | 11 | 0 | | |

16- Que remarquerez-vous en analysant le fichier « Sales » ? Y a-t-il des incohérences ?

Si nous analysons le fichier « Sales », nous constatons que les données ne sont pas toutes du même type, alors que cela devraient être de type « double ». Des données de type « texte » sont également présentes.

Nous appliquons un filtre sur la colonne « Sales » afin de garantir que seules les valeurs de type « double » sont conservées, pour ce faire, nous utilisons cette expression :

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

```
row3.Sales.matches("-?\\d+(\\.\\d+)?") ? row3.Sales : "0"
```

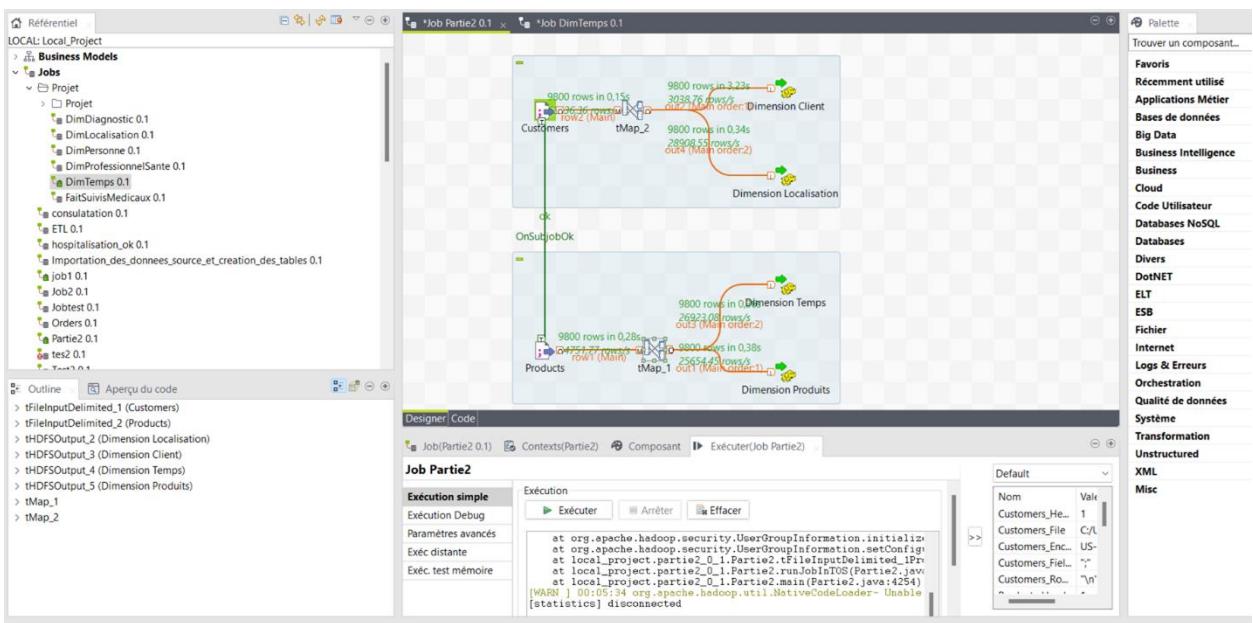
Explication de l'expression régulière utilisée :

- ^ : Début de la chaîne.
- -? : Correspond à un signe négatif éventuel (-).
- \d+ : Correspond à un ou plusieurs chiffres avant la virgule.
- (\.\d+)? : Correspond à une virgule suivie d'un ou plusieurs chiffres, mais cette partie est facultative (?), donc elle peut ou non être présente.
- \$: Fin de la chaîne.

The screenshot shows the Talend Open Studio interface for a tMap component named 'tMap_1'. The left panel displays the input schema 'row1' with columns: Row_ID, Order_ID, Order_Date, Ship_Date, Ship_Mode, Product_ID, Category, Sub_Category, Product_Name, and Sales. The middle panel shows the mapping configuration. The 'Var' section contains an expression: `TalendDate.formatDate("dd-MM-yyyy", TalendDate.parseDate("dd/MM/yyyy", row1.Order_Date))`. The 'out1' section maps this to columns: SK_Product, Product_ID, Category, Sub_Category, Product_Name, and Sales. The 'out3' section maps it to columns: Sk_Temps, Row_ID, Jour, Mois, Annee, and Date. The bottom panels show the detailed schema editor for 'row1' and 'out3', where specific column properties like length and precision are defined. Yellow arrows highlight the connection from the input 'Row_ID' to the output 'Row_ID' and from the input 'Order_Date' to the output 'Date'.

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



17- Dans la VM Cloudera, vérifiez les 4 fichiers dans HDFS

Nous pouvons consulter maintenant l'existence des 4 fichiers dans la VM Cloudera dans HDFS :

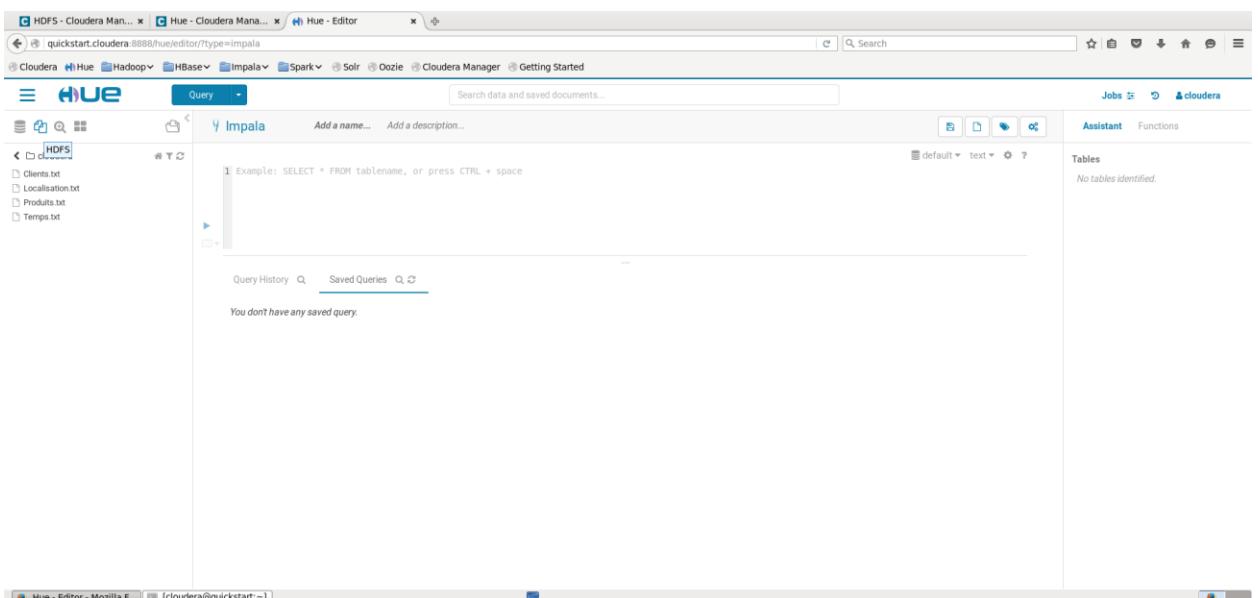
Soit via la commande : « dhfs dfs -ls /user/cloudera/ »

```
cloudera@quickstart:~$ dhfs dfs -ls /user/cloudera/
[cloudera@quickstart ~]$ Found 4 items
-rw-r--r-- 1 cloudera cloudera 417390 2024-02-06 15:05 /user/cloudera/Clients.txt
-rw-r--r-- 1 cloudera cloudera 290415 2024-02-06 15:05 /user/cloudera/Localisation.txt
-rw-r--r-- 1 cloudera cloudera 837040 2024-02-06 15:05 /user/cloudera/Produits.txt
-rw-r--r-- 1 cloudera cloudera 263493 2024-02-06 15:05 /user/cloudera/Temps.txt
[cloudera@quickstart ~]$
```

Sinon via l'interface « HUE »

BLOC : BIG DATA

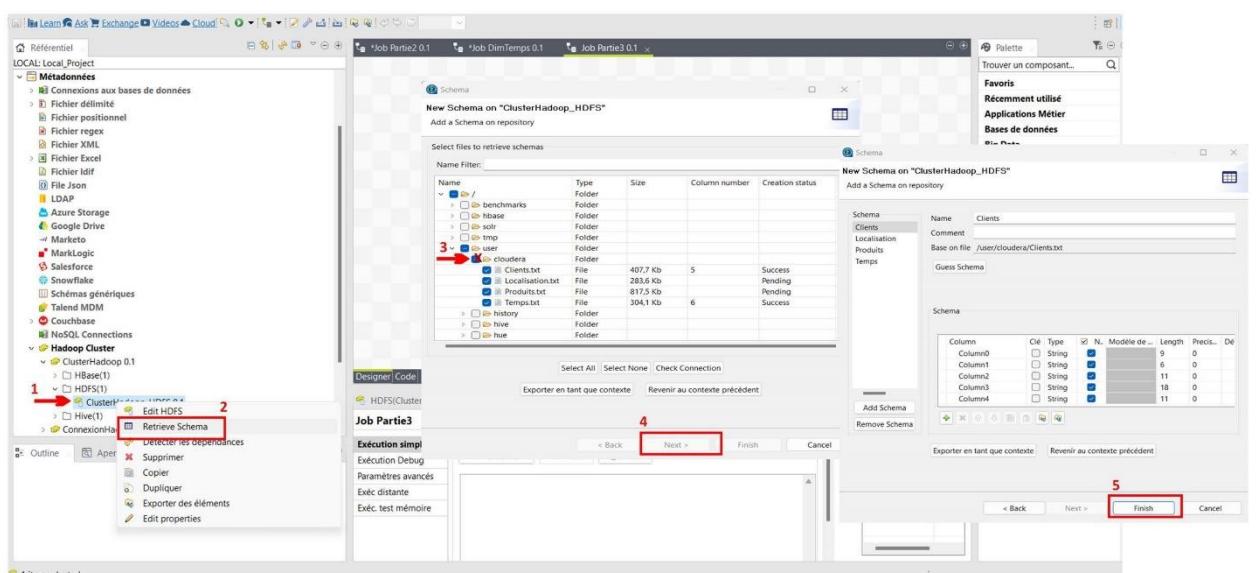
WORKSHOP : INTEGRATION DE DONNEES



Passons maintenant à la Création de la table des faits :

1ère étape consiste à récupérer les schémas des tables que nous venons de charger dans HDFS;

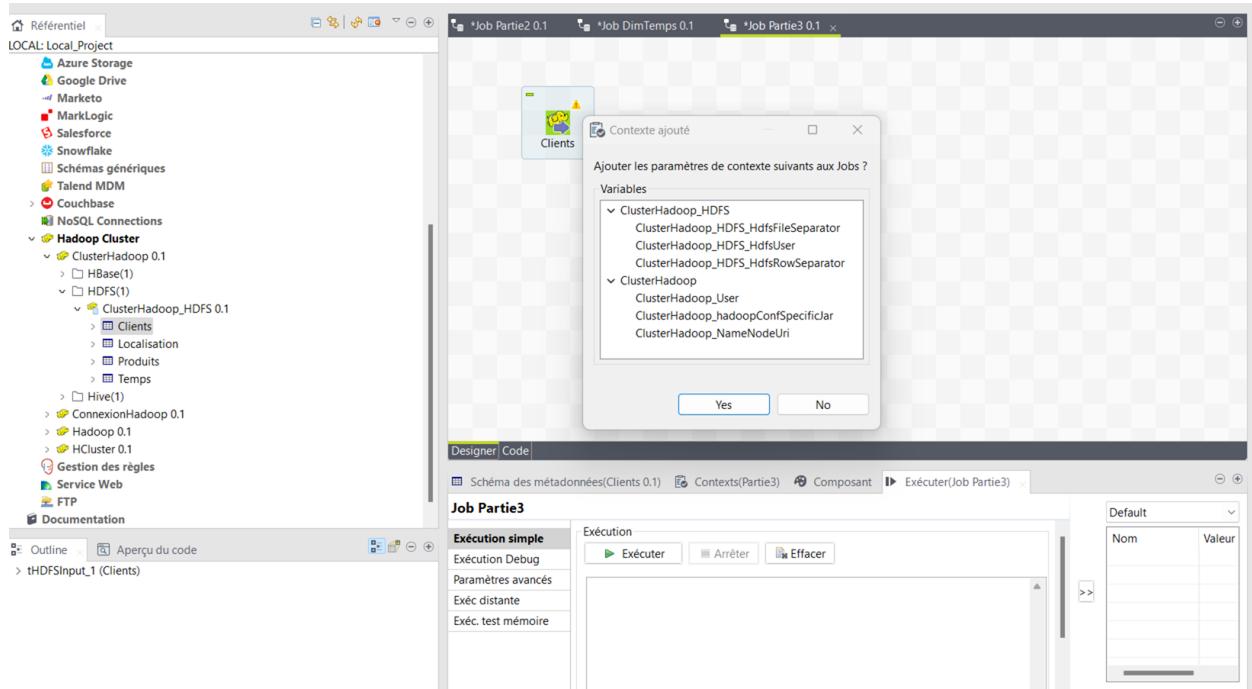
Pour illustrer cela, la figure suivante est présentée :



BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

Nous passerons maintenant à la génération de la table de fait à travers des différentes dimensions :

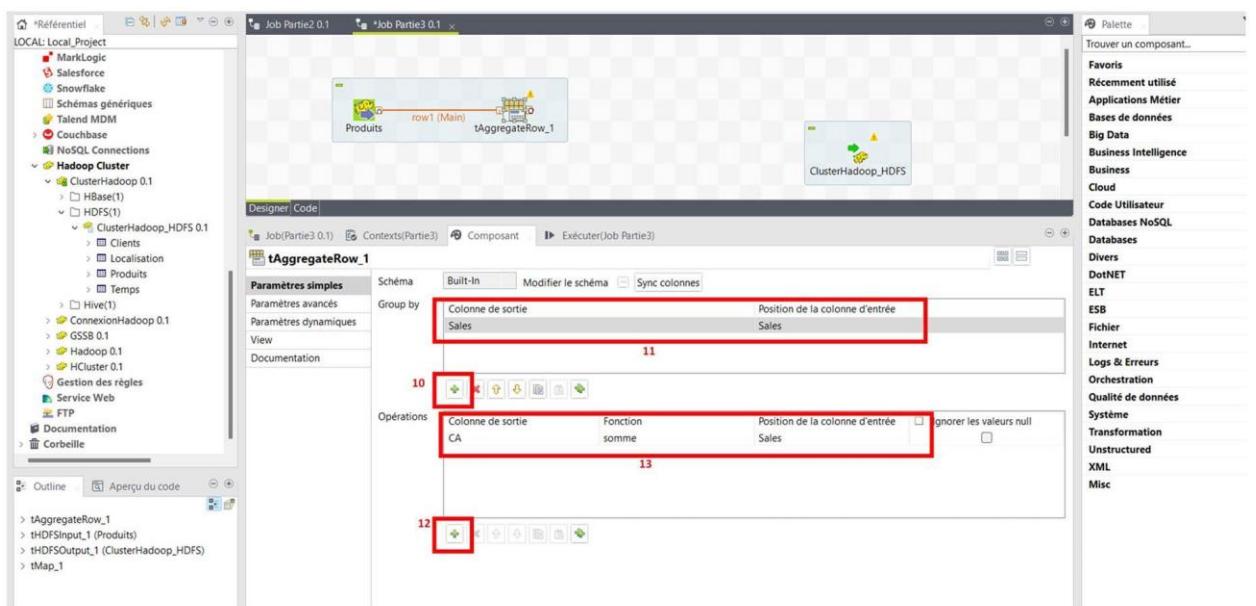
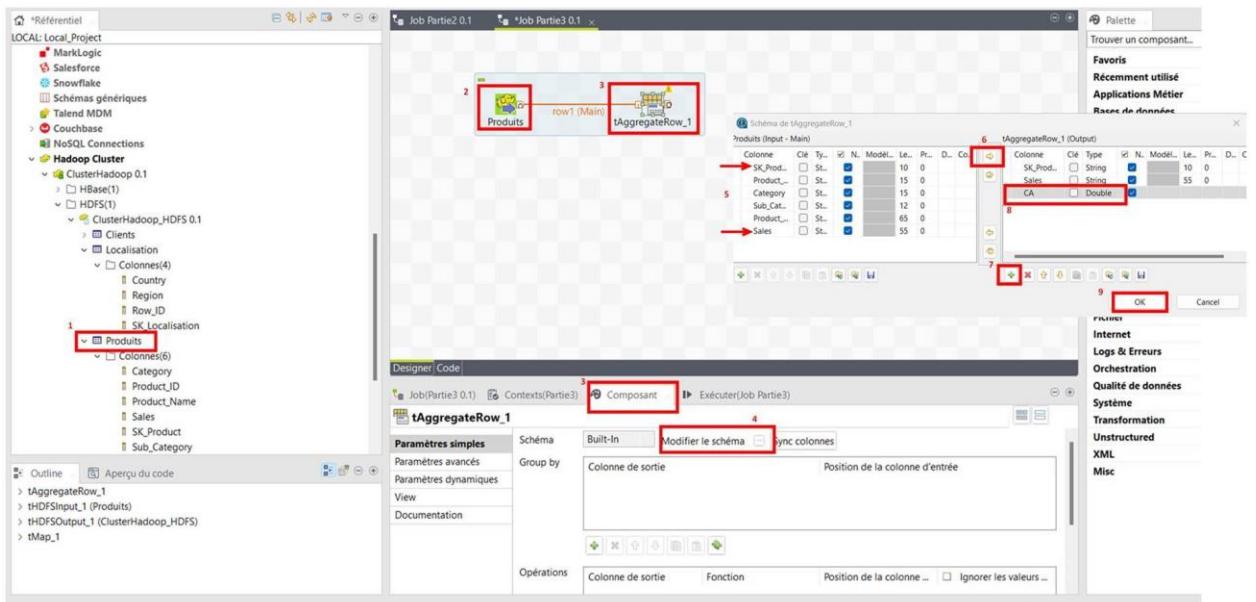


Déposez également un composant tHDFSOutput qui servira à stocker la table de faits.

Rappelez-vous que cette table contient les clés de substitution des dimensions avec la mesure du chiffre d'affaires (CA), ce qui nous permettra par la suite de répondre aux besoins initiaux.

BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES



BLOC : BIG DATA

WORKSHOP : INTEGRATION DE DONNEES

