
Atelier 1 : ETL Talend et MySQL

Les objectifs de cet atelier sont les suivants :

1. **Compréhension de l'ETL (Extract, Transform, Load)** : Les participants devraient comprendre le concept général de l'ETL et son importance dans le processus d'intégration de données. Cela inclut la compréhension des différentes étapes du processus ETL, à savoir l'extraction des données à partir de sources variées, leur transformation selon les besoins métier spécifiques, et enfin leur chargement dans une base de données cible.
2. **Maîtrise de Talend** : Les participants devraient acquérir une connaissance pratique de Talend en tant qu'outil d'intégration de données. Cela implique de comprendre l'interface de Talend Studio, la manière de créer et de gérer des projets, ainsi que la manipulation des composants et des connexions pour concevoir des flux ETL efficaces.
3. **Manipulation des composants** : Les participants devraient apprendre à utiliser des composants spécifiques tels que TMap, LogRow et MySQLInput dans le cadre de la création de flux ETL. Cela comprend la compréhension des fonctionnalités de chaque composant, leur configuration appropriée et leur intégration dans des flux de données complexes.
4. **Gestion des erreurs et débogage** : Les participants devraient être capables de gérer les erreurs et les exceptions qui peuvent survenir lors de l'exécution des flux ETL. Cela implique l'utilisation de techniques de débogage telles que l'enregistrement des données à différentes étapes du processus, la gestion des exceptions et la mise en place de stratégies de récupération en cas d'échec.

1. Télécharger Talend

- https://perso.isima.fr/~jofontan/Install/TOS_DI-Win32-20200219_1130-V7.3.1.exe

2. Préparer une instance MySQL avec les données sources

Assurez-vous d'avoir le fichier SQL ([gestion_livres_init.sql](#)) dans le même répertoire que votre fichier `docker-compose.yml`.

Pour exécuter ce fichier `docker-compose.yml`, suivez ces étapes :

- a) Placez votre `docker-compose.yml` et vos fichiers SQL [gestion_livres_init.sql](#) dans un même répertoire.

```
CREATE DATABASE IF NOT EXISTS `gestion_livres` DEFAULT CHARACTER SET utf8;
USE `gestion_livres`;

-- Table `auteur`
DROP TABLE IF EXISTS `auteur`;
CREATE TABLE `auteur` (
  `NUMERO_A` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NOM` varchar(450) DEFAULT NULL,
  `PRENOM` varchar(450) DEFAULT NULL,
  `DOMICILE` varchar(450) DEFAULT NULL,
  PRIMARY KEY (`NUMERO_A`)
) ENGINE=InnoDB AUTO_INCREMENT=8547586 DEFAULT CHARSET=latin1;

-- Données pour table `auteur`
INSERT INTO `auteur` VALUES
(8547, 'Chambord', 'Emilie', 'Nice'),
(52136, 'Fabiere', 'Sylvie', 'Bordeaux'),
(78545, 'TINTIN', 'Thierry', 'Clermont'),
(85478, 'Castafiore', 'Emilie', 'Paris'),
(542536, 'Dupont', 'Pierre', 'Avignon'),
(8547585, 'Momo', 'Roland', 'Toulouse');

-- Table `livre`
DROP TABLE IF EXISTS `livre`;
CREATE TABLE `livre` (
  `NUMERO_L` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NUMERO_A` int(10) unsigned NOT NULL,
  `TITRE` varchar(128) DEFAULT NULL,
  `PRIX` float DEFAULT NULL,
  PRIMARY KEY (`NUMERO_L`)
) ENGINE=InnoDB AUTO_INCREMENT=11324 DEFAULT CHARSET=latin1;

-- Données pour table `livre`
INSERT INTO `livre` VALUES
(10101, 85478, 'aaaaaa', 10),
(11111, 8547, 'eeeeee', 54),
```

```
(11122,52136,'fffff',12),  
(11231,542536,'bbbbbb',34),  
(11321,8547585,'eeeeeee',29),  
(11323,8547585,'ccccccccc',45);
```

- b) Ouvrez un terminal dans ce répertoire.
- c) Exécutez la commande suivante pour démarrer les services définis dans le fichier `docker-compose.yml`:

```
version: '3.8'

services:
  mysql_db:
    image: mysql:latest
    container_name: mysql_db
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: mysecret
      MYSQL_DATABASE: gestion_livres
    volumes:
      - ./gestion_livres_init.sql:/docker-entrypoint-
initdb.d/gestion_livres_init.sql
    restart: always # Ajout de cette ligne pour redémarrer le conteneur en
cas d'échec

networks:
  - mysql_network

networks:
  mysql_network:
    driver: bridge
```



Talend for Data Integration



Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

```
mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_gestion_livres |
+-----+
| auteur
| livre
+-----+
2 rows in set (0.00 sec)

mysql> select * from livre
->
+-----+
| NUMERO_A | NUMERO_B | TITRE | PRIX |
+-----+
| 10101 | 85478 | aaaaaa | 16 |
| 11111 | 8547 | eeeee | 54 |
| 11122 | 52136 | ffffff | 12 |
| 11231 | 542536 | bbbbb | 34 |
| 11321 | 8547585 | eeeee | 29 |
| 11323 | 8547585 | cccddd | 45 |
+-----+
6 rows in set (0.00 sec)

mysql> select * from auteur
->
+-----+
| NUMERO_A | NOM | PRENOM | DOMICILE |
+-----+
| 8547 | Chambord | Emilie | Nice |
| 52136 | Fabriere | Sylvie | Bordeaux |
| 78545 | TINTIN | Thierry | Clermont |
| 85478 | Castaflore | Emilie | Paris |
| 542536 | Dupont | Pierre | Avignon |
| 8547585 | Momo | Roland | Toulouse |
+-----+
6 rows in set (0.00 sec)

mysql>
```

3. Créer un projet

Select what you want to do next:

Create a new project: Local_Project

Import a demo project

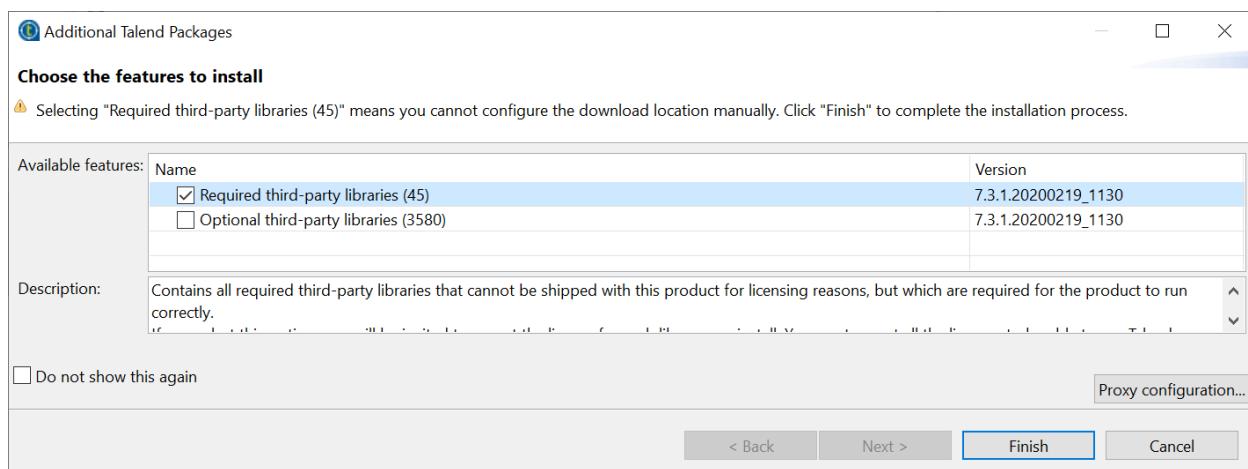
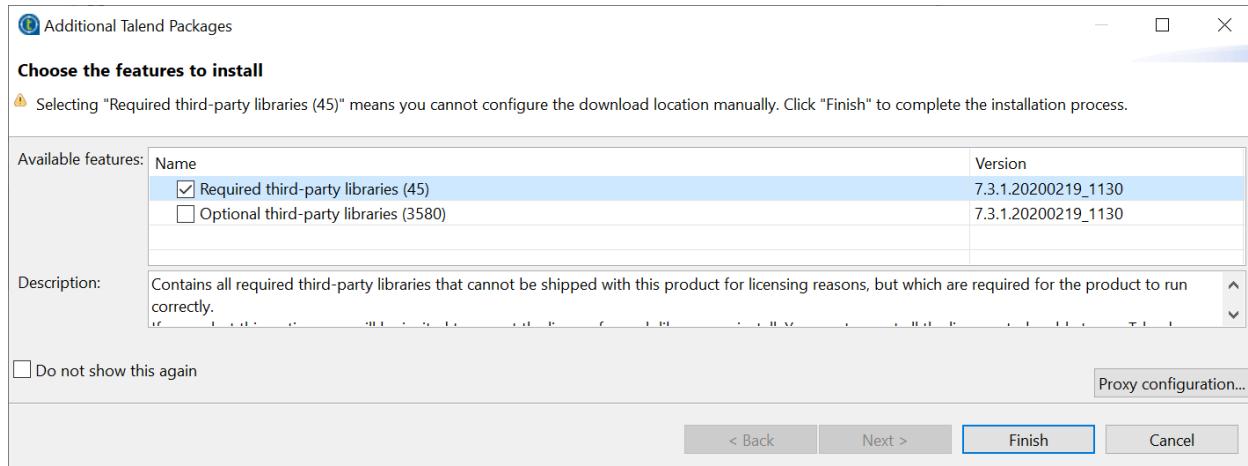
Import an existing project

Manage Connections

Always ask me at startup

Finish

4. Installer les fonctionnalités avancées



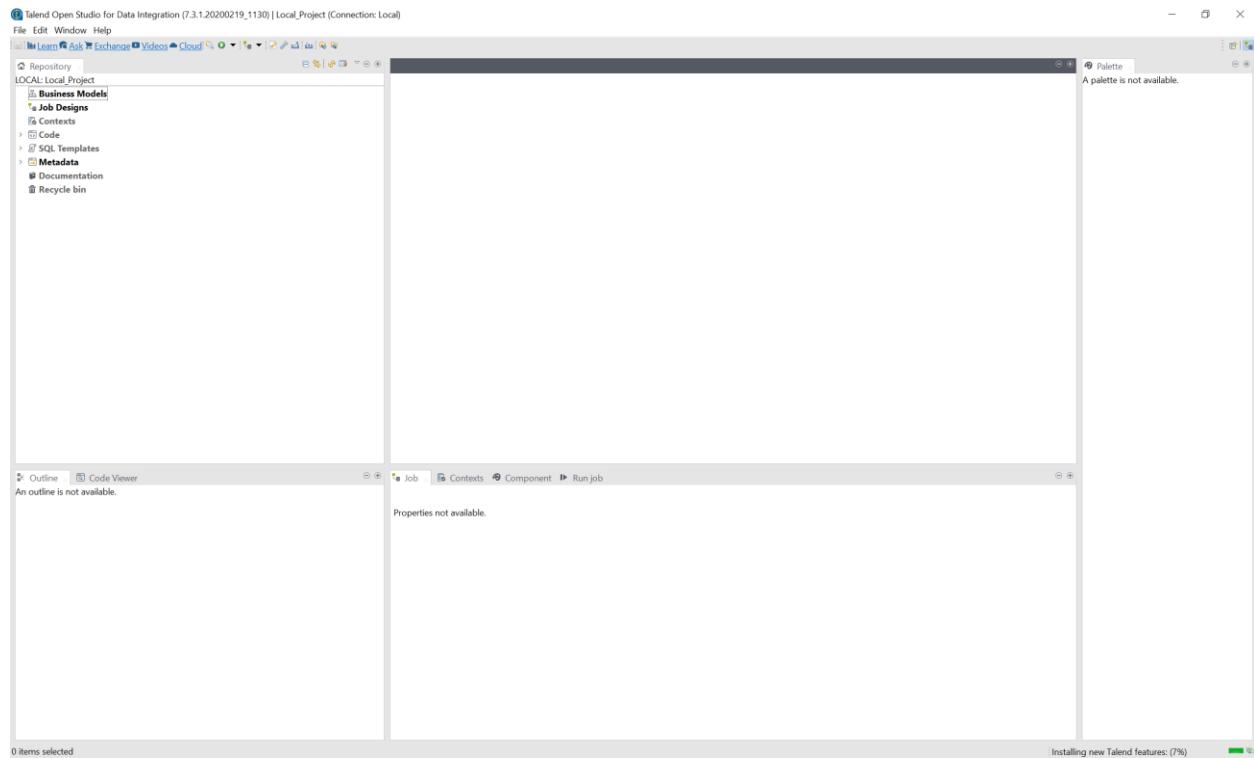
5. Interface d'accueil de Talend



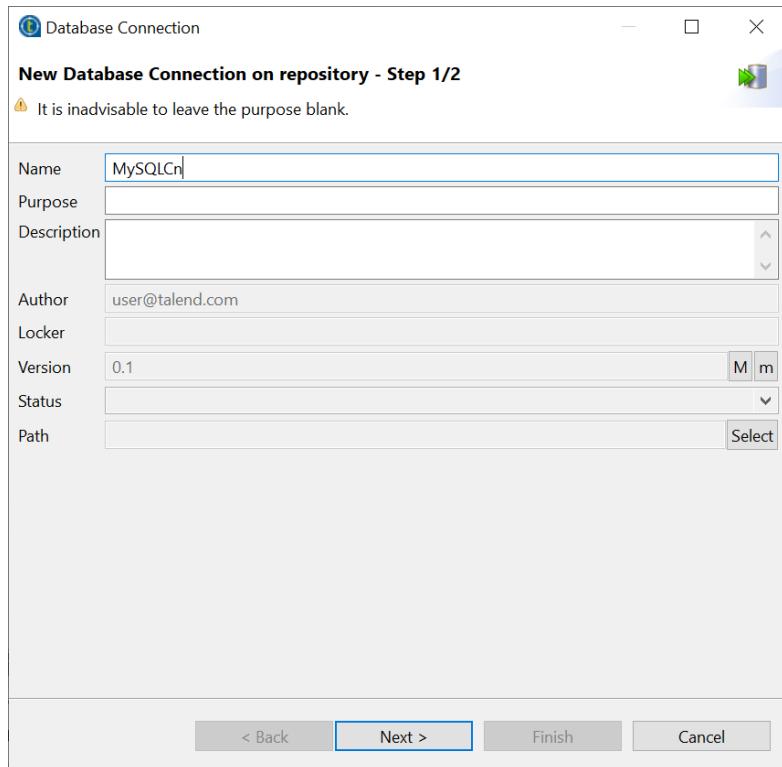
Talend for Data Integration



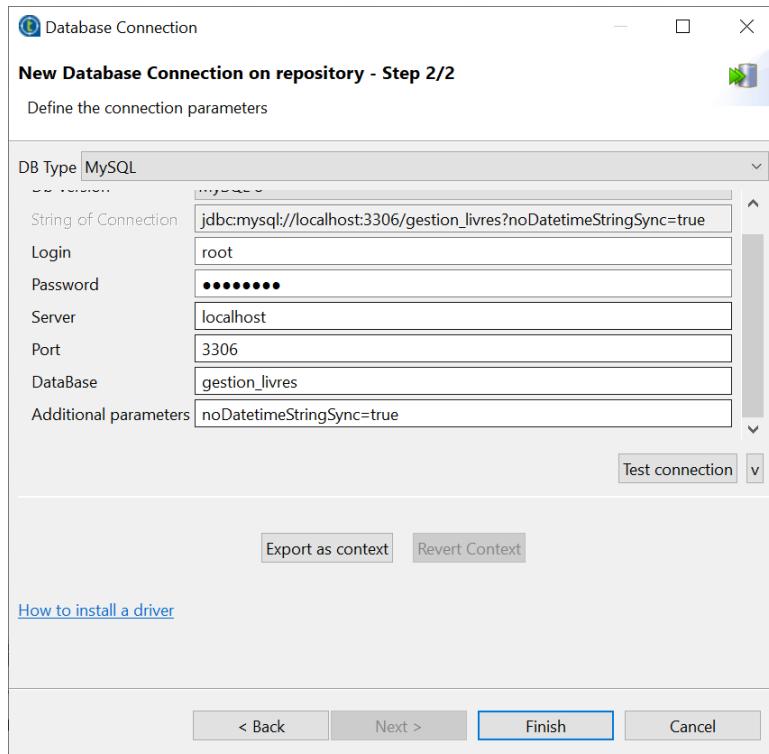
Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



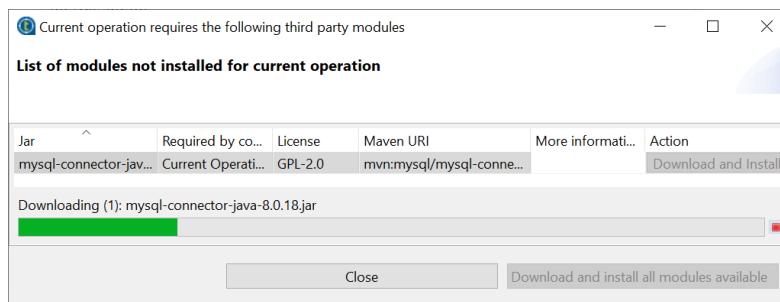
6. Crédit d'une connexion à une base MySQL :



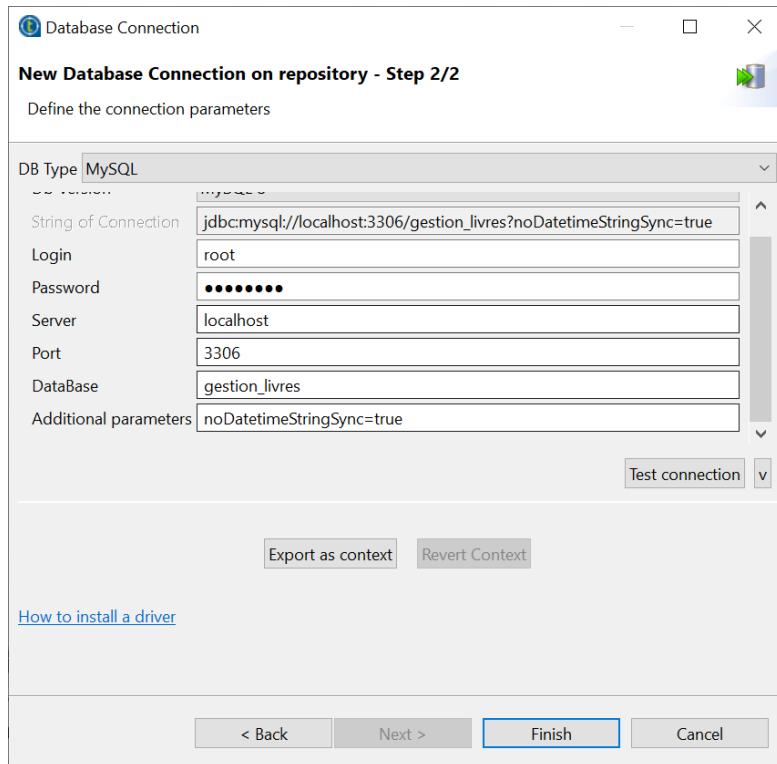
7. Configurer les paramètres de connexion à la base de données



8. Installer les jars nécessaires

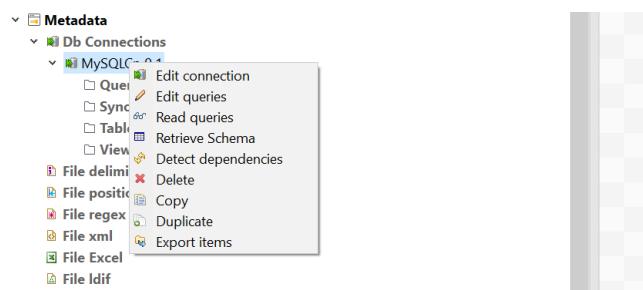


9. Tester la connexion

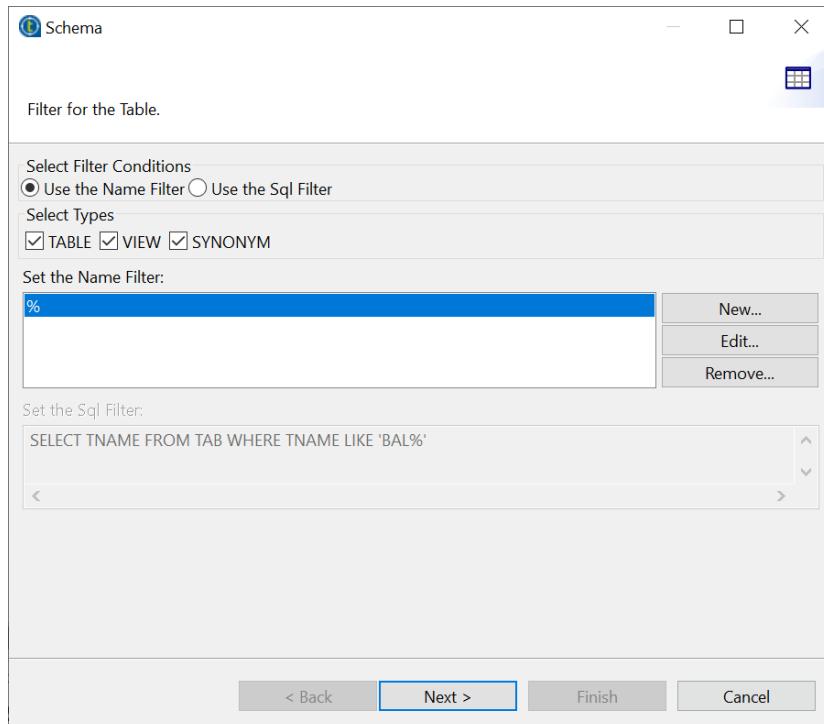


10. Création d'une connexion à une base MySQL :

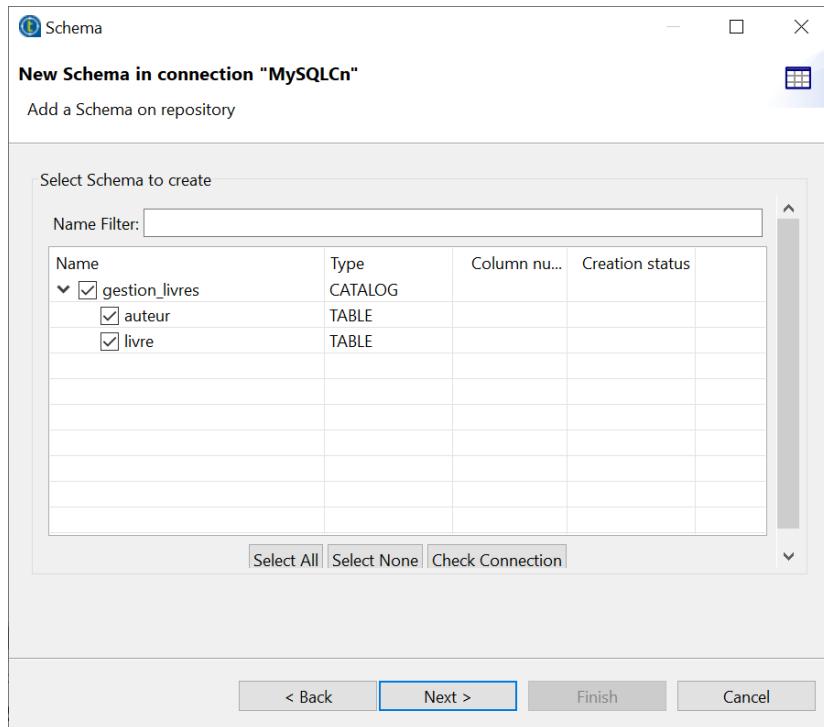
– Récupération du schéma des tables



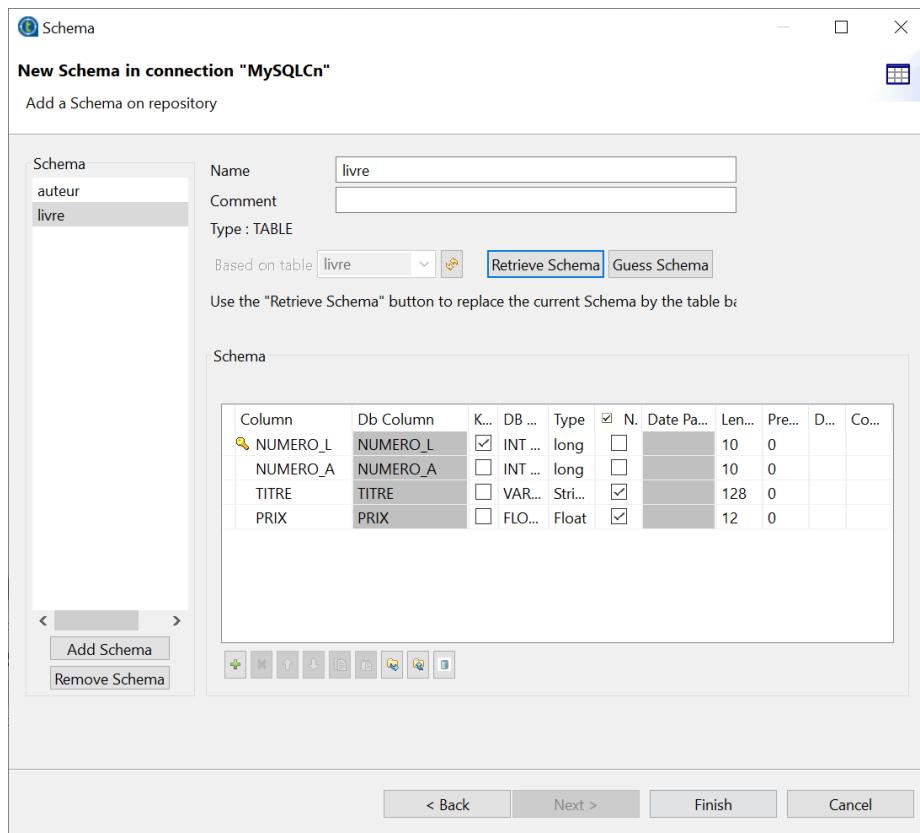
11. Cocher Table et cliquer « Next »



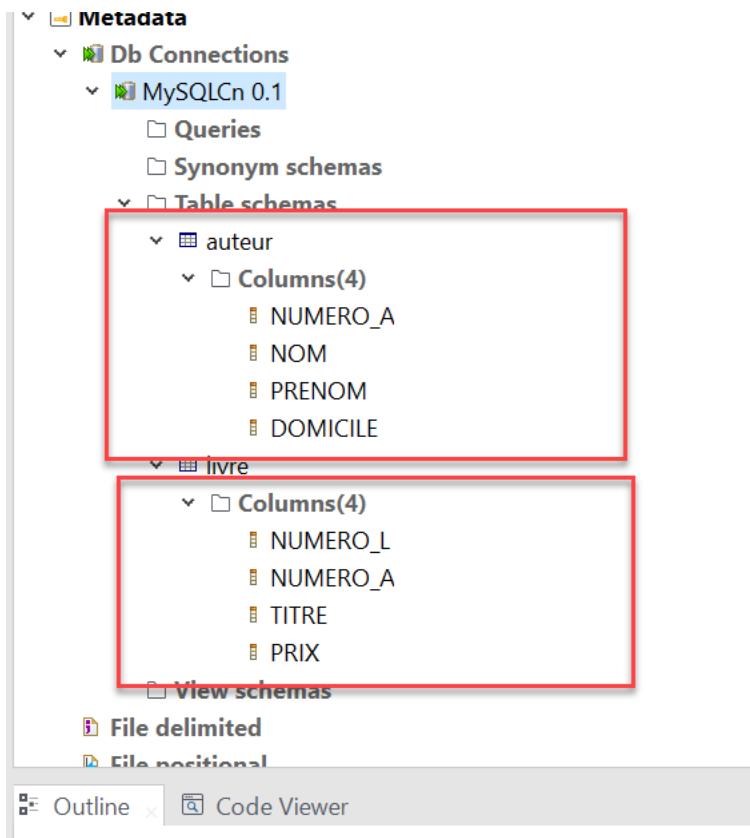
12. Sélectionner la base de données « gestion livres »



13. Choisir les colonnes désirées en cas de besoins et cliquer « Finish »

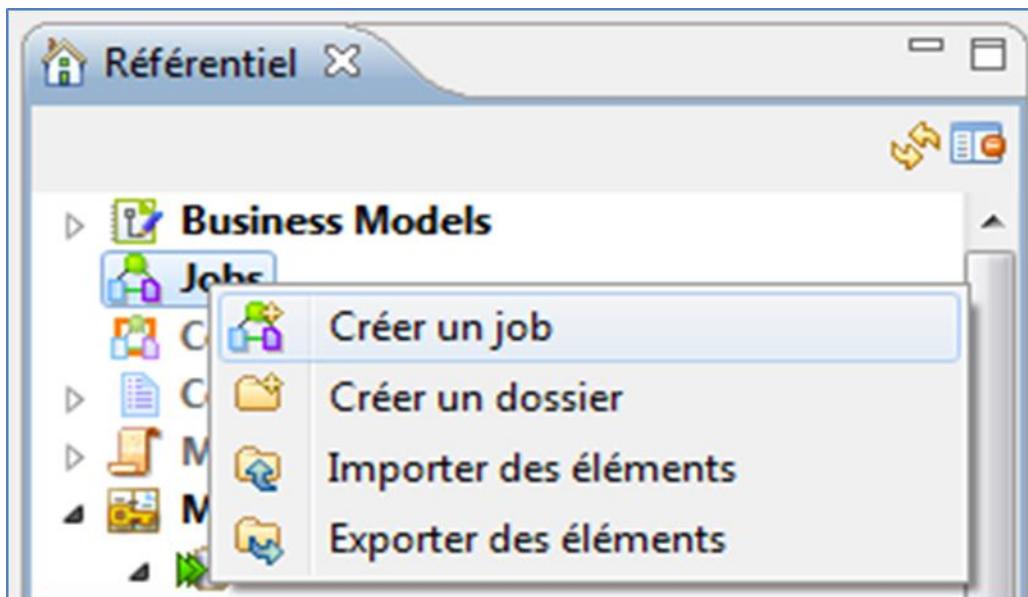


14. Vérifier les schémas de chaque table



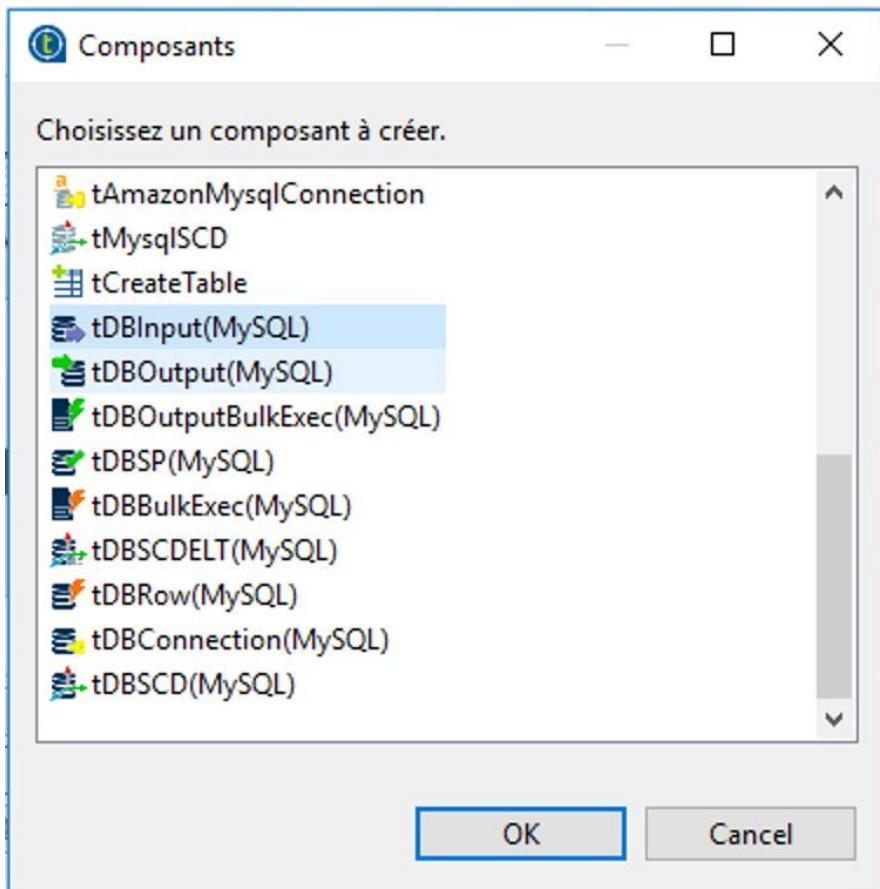
The screenshot shows the Talend Studio interface with the 'Metadata' view open. Under 'Db Connections', the 'MySQLCn 0.1' connection is selected. The 'Table schemas' section contains two entries: 'auteur' and 'livre'. Both entries have a red box drawn around them. The 'auteur' entry shows four columns: NUMERO_A, NOM, PRENOM, and DOMICILE. The 'livre' entry shows four columns: NUMERO_L, NUMERO_A, TITRE, and PRIX. At the bottom of the interface, there are tabs for 'Outline' and 'Code Viewer', with 'Outline' being the active tab.

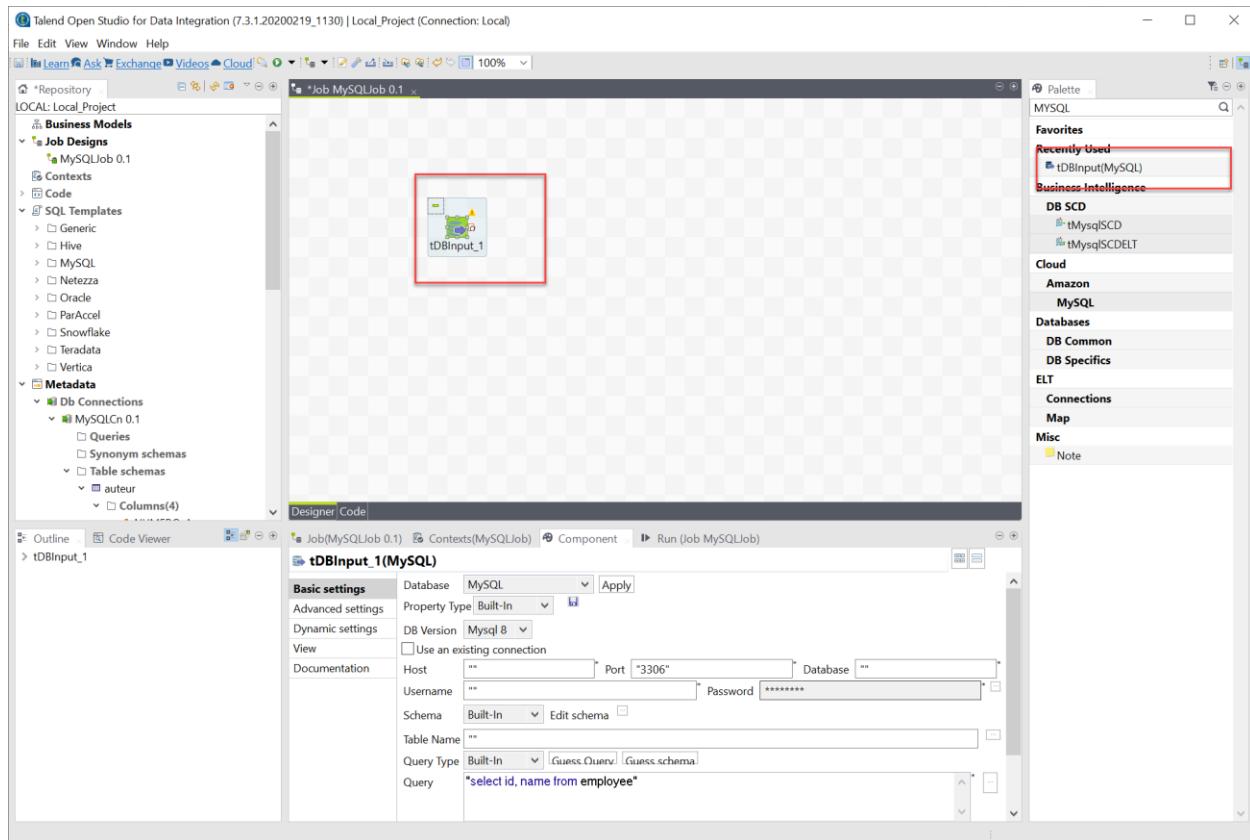
15. Création d'un premier job :



16. Création d'un premier job : ajout de la connexion

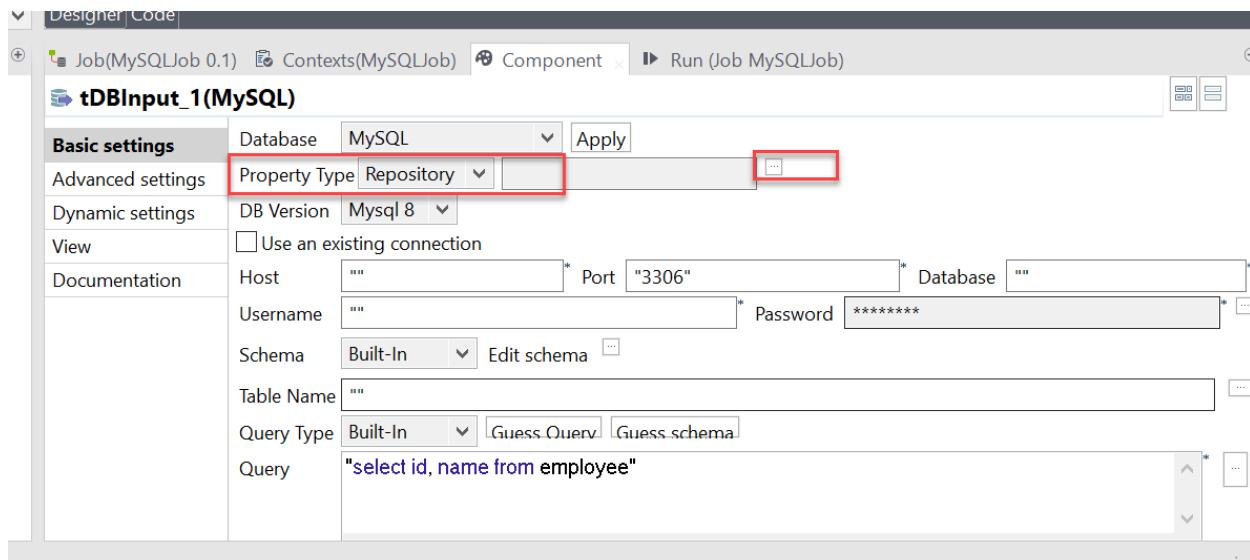
– Faire glisser le tDBInput (MySQL) dans la partie Design avec la table Auteur



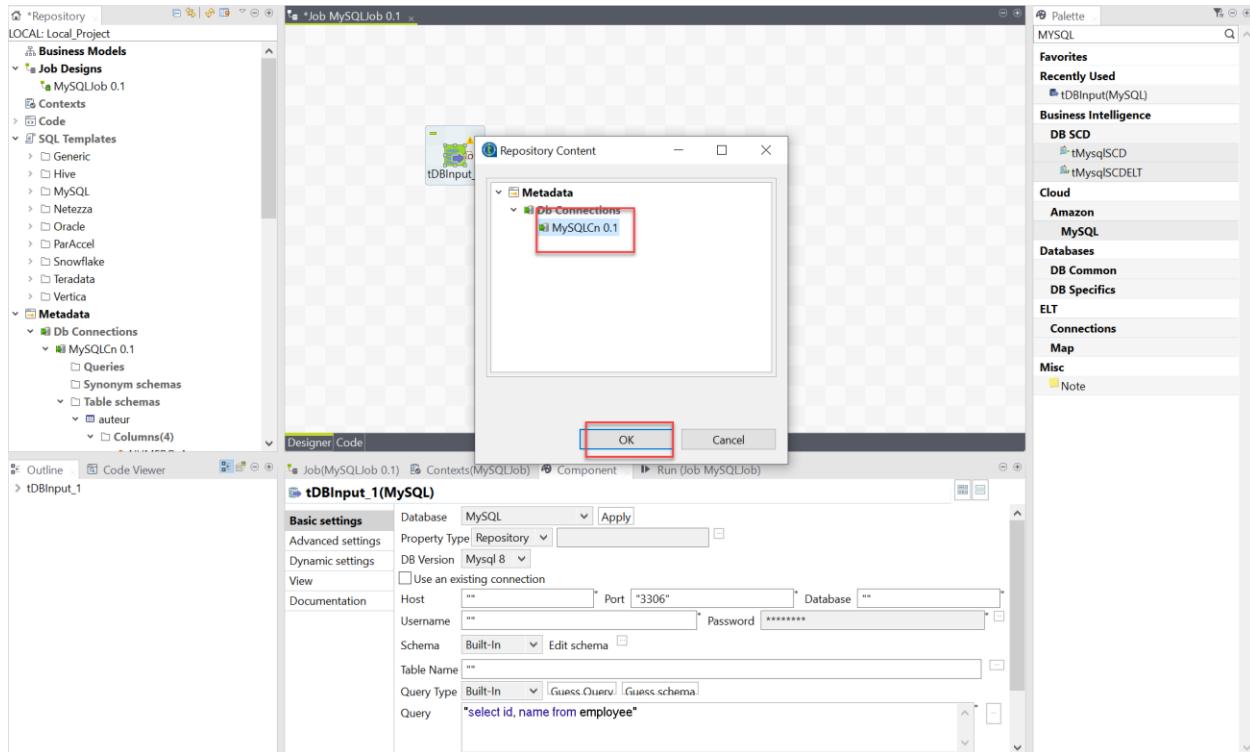


17. Configurer le composant tDBInput(MYSQL)

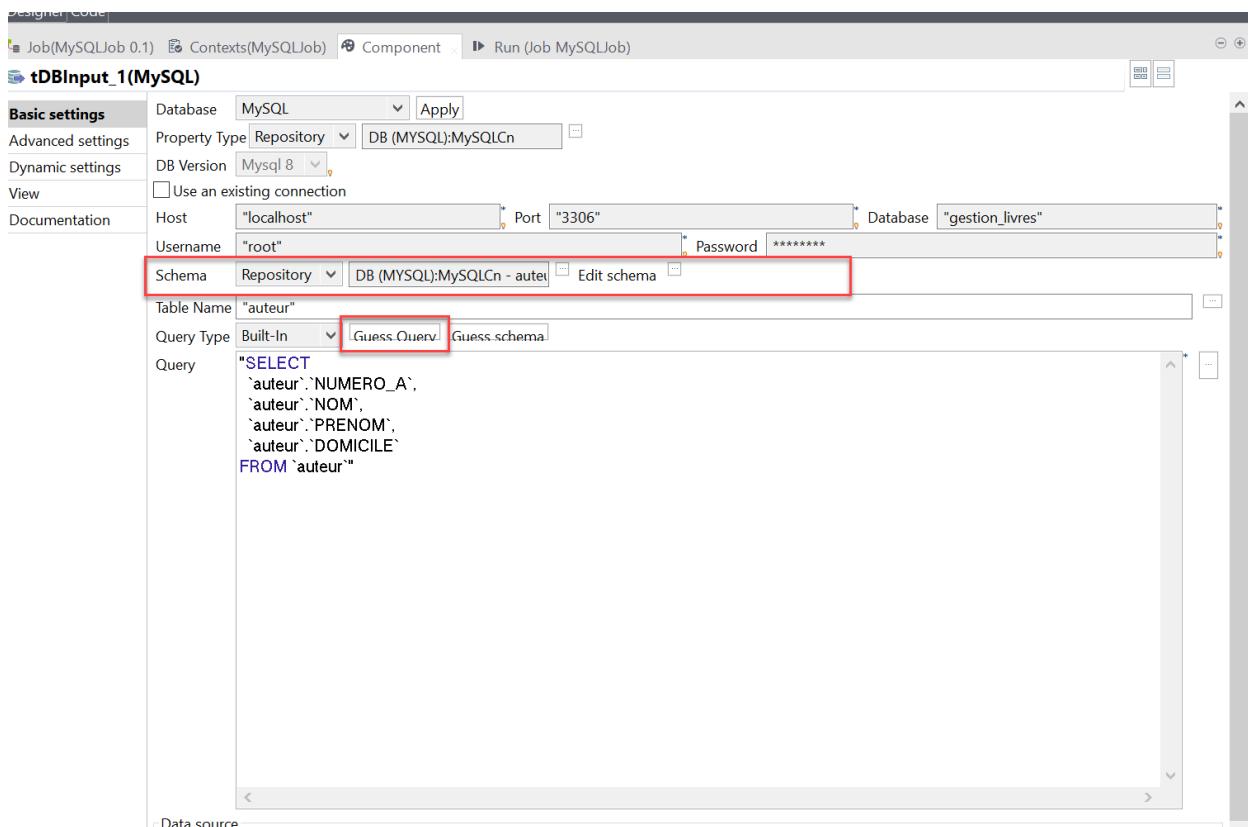
Selectionner le type « Property Type » la valeur « Repository » et cliquer sur



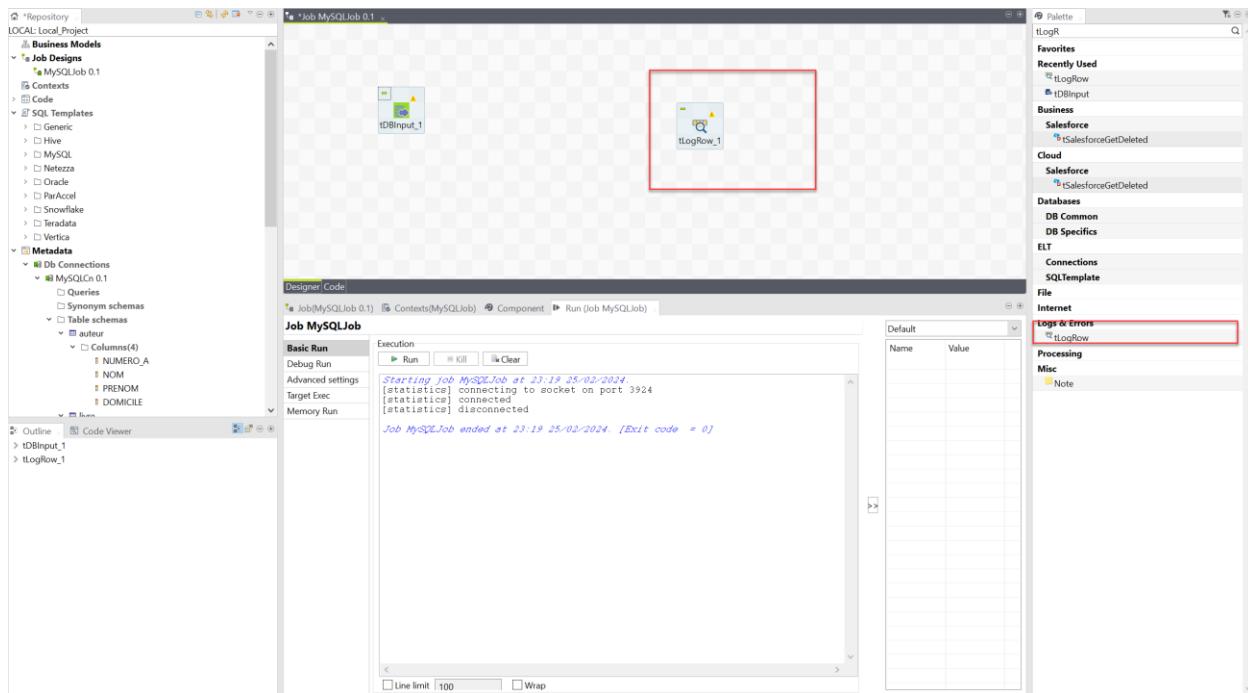
Sélectionner le type « MYSQLCN 0.1et cliquer sur **OK**



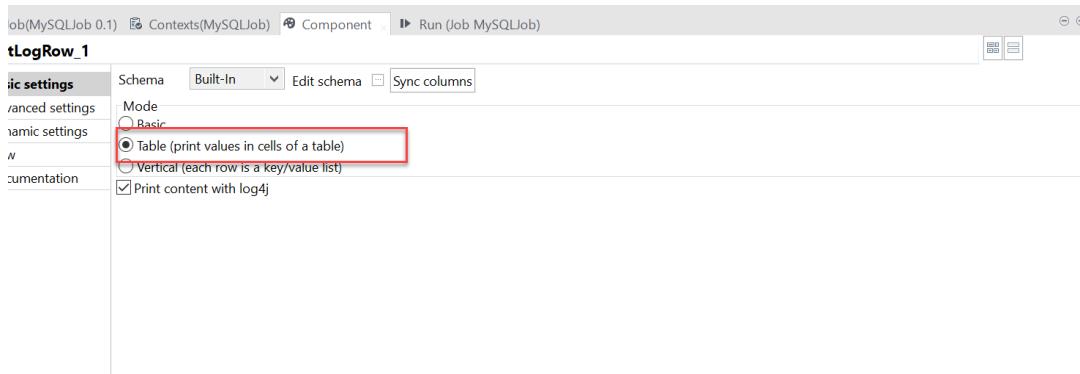
configurer le schéma et la requête à exécuter en cliquant sur Gues Query pour générer une requête de sélection avec les champs nécessaires



18. Création d'un premier job : ajout d'un composant de log

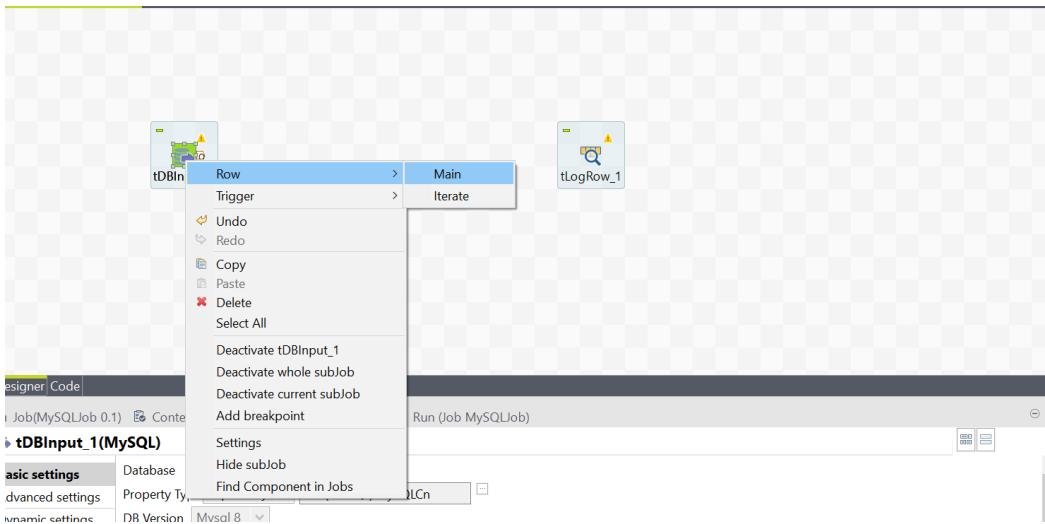


19. Configurer le composant de log



20. Liaison des composants

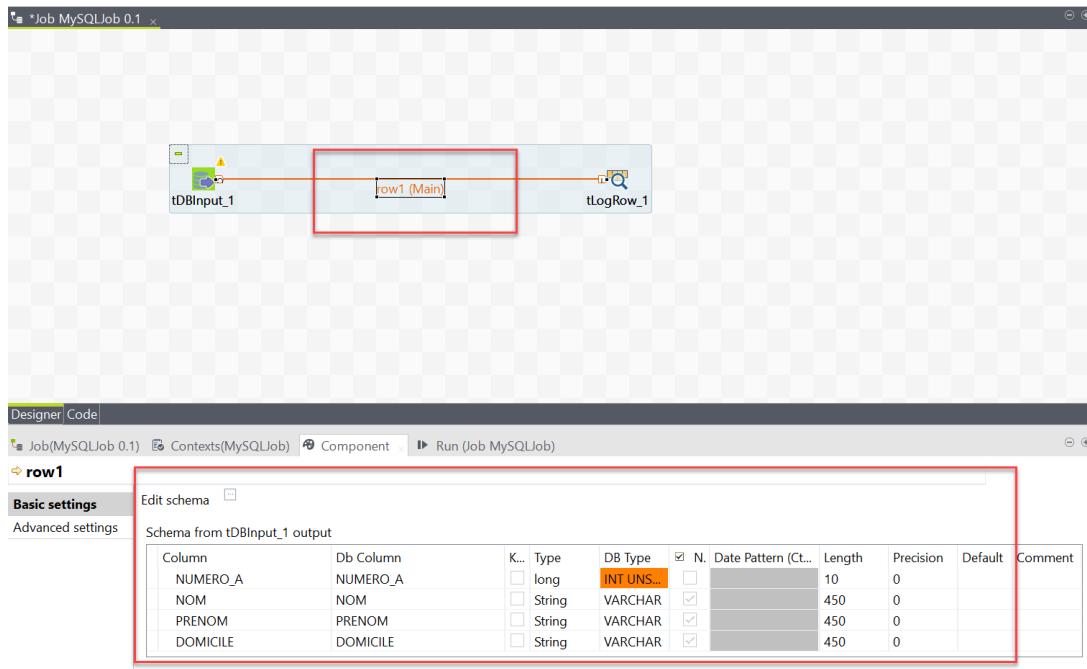
- Cliquer sur le composant tbDinput(MySQL) avec le bouton droit et afficher le menu contextuel



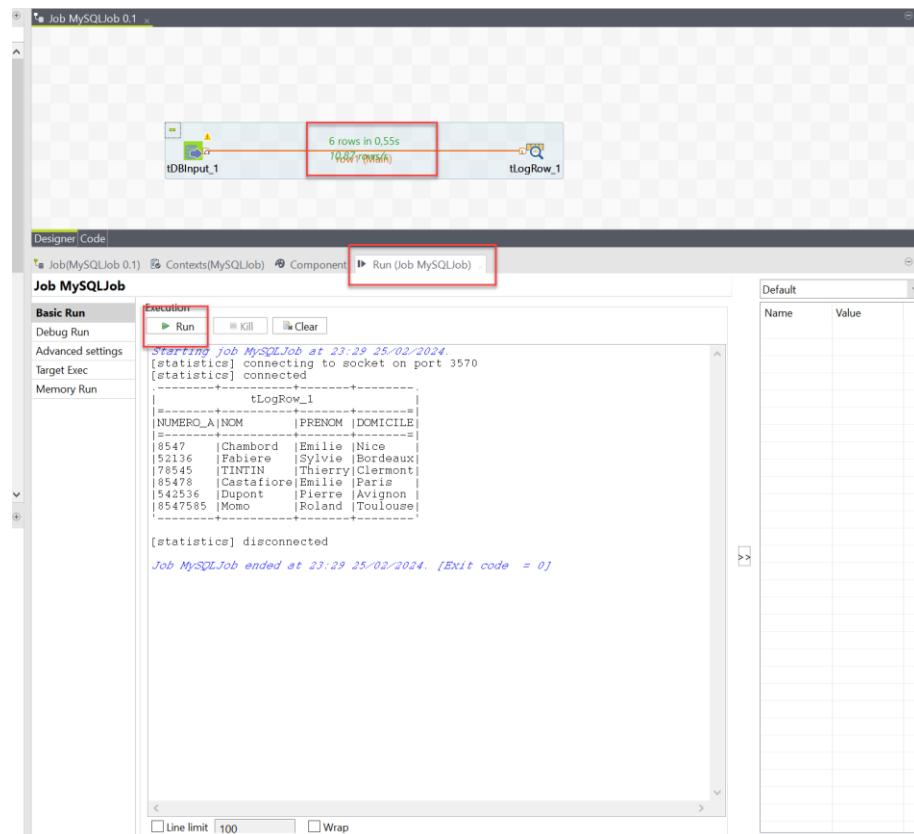
- Choisir Row -> Main et cliquer sur le composant tLogRow_1 pour établir la liaison



- La liaison est établie en fournissant l'output attendu

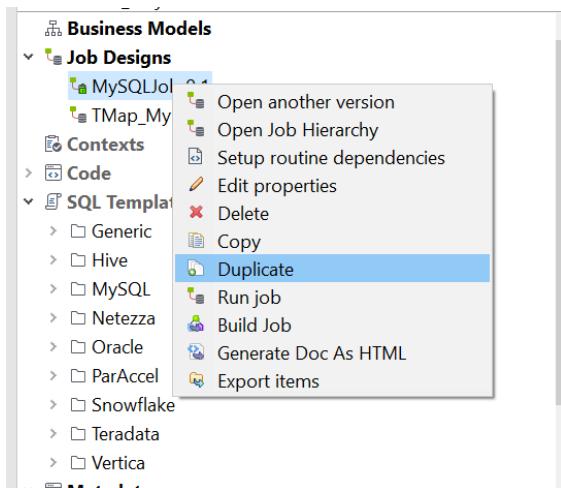


21. Exécuter le Job

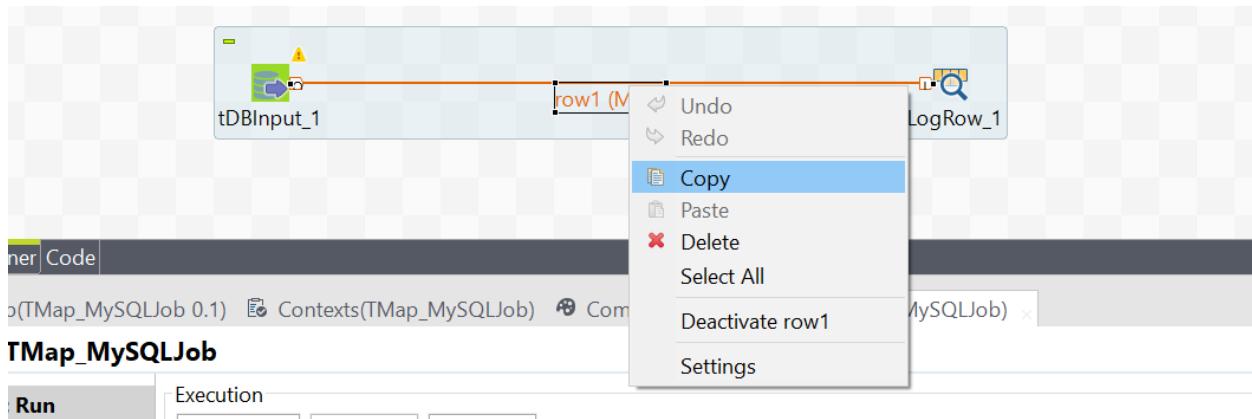


22. Modification de ce job : ajout d'un composant d'aiguillage tMap

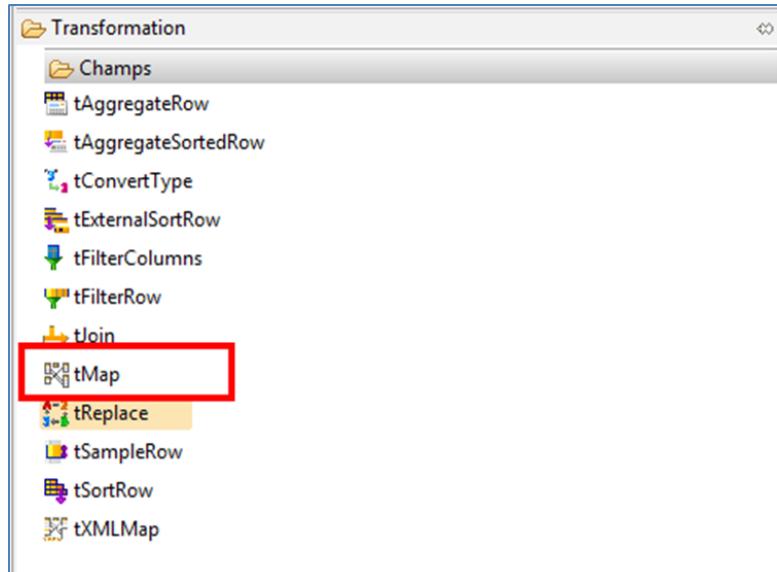
- Duplicer le projet



- Supprimer la liaison dans le nouveau job



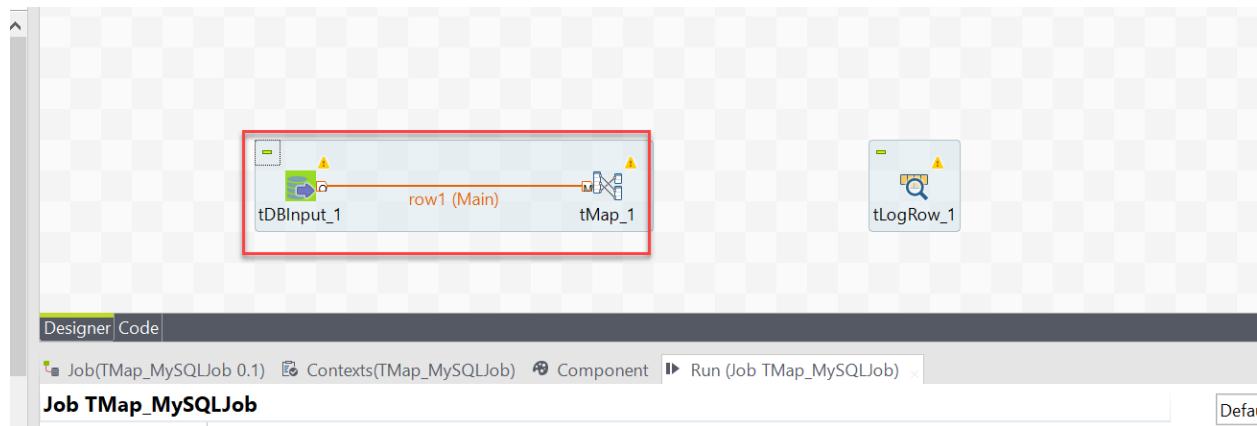
- Ajouter un composant tMap



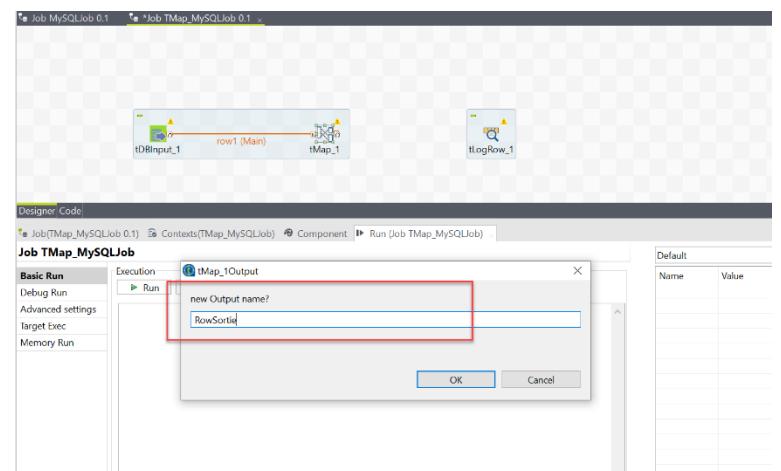
- Modification de ce job : liaison avec les autres composants

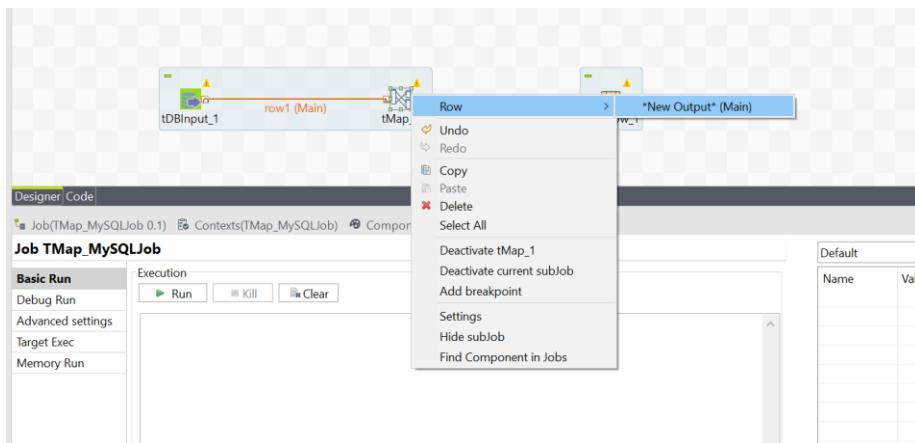


- Modification de ce job : Ajout liaison entre liaison Input(MySQL) entre tMap

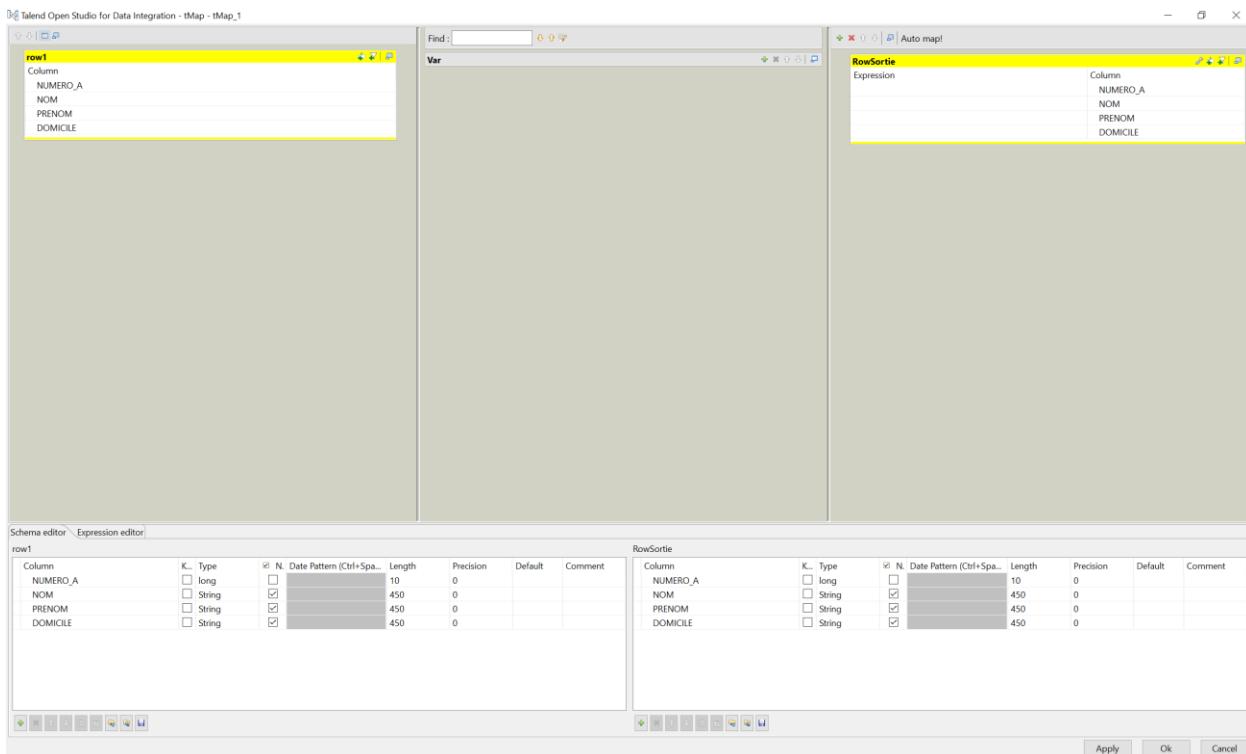


- Etablir liaison entre tMapr et LogRow en spécifiant le nom de l'output



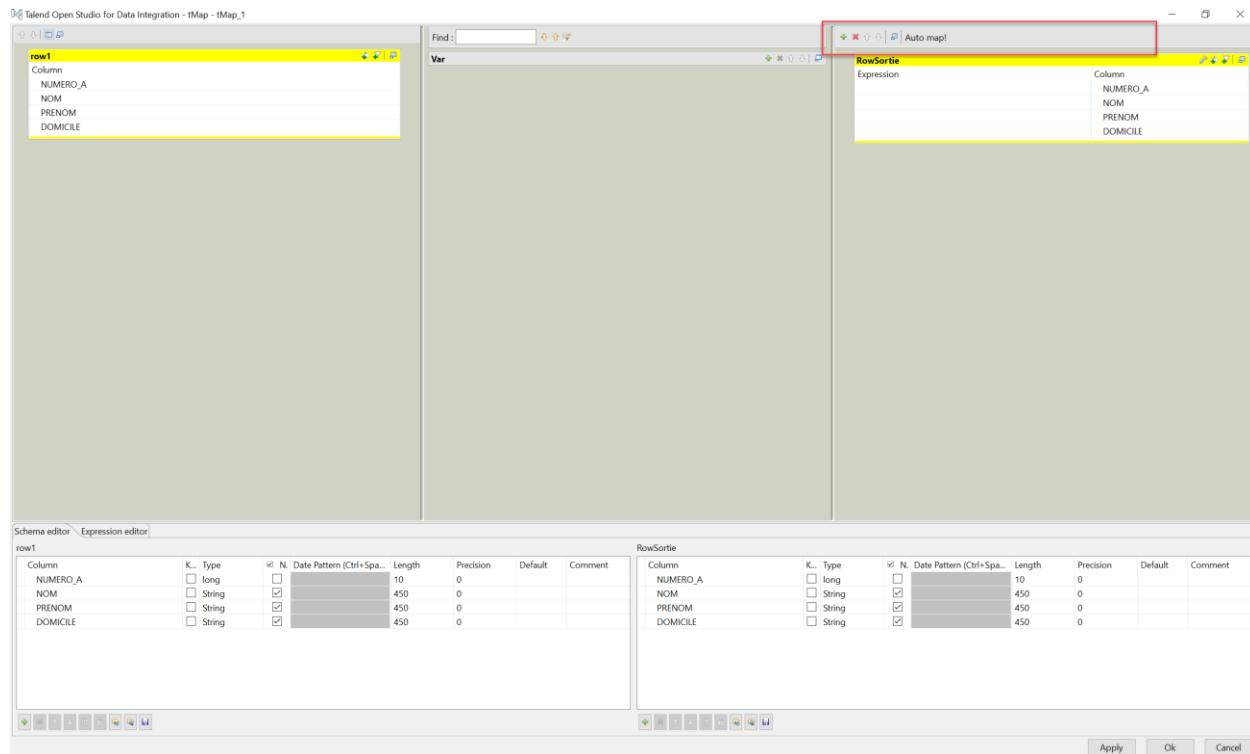


- Ouvrir le tMap

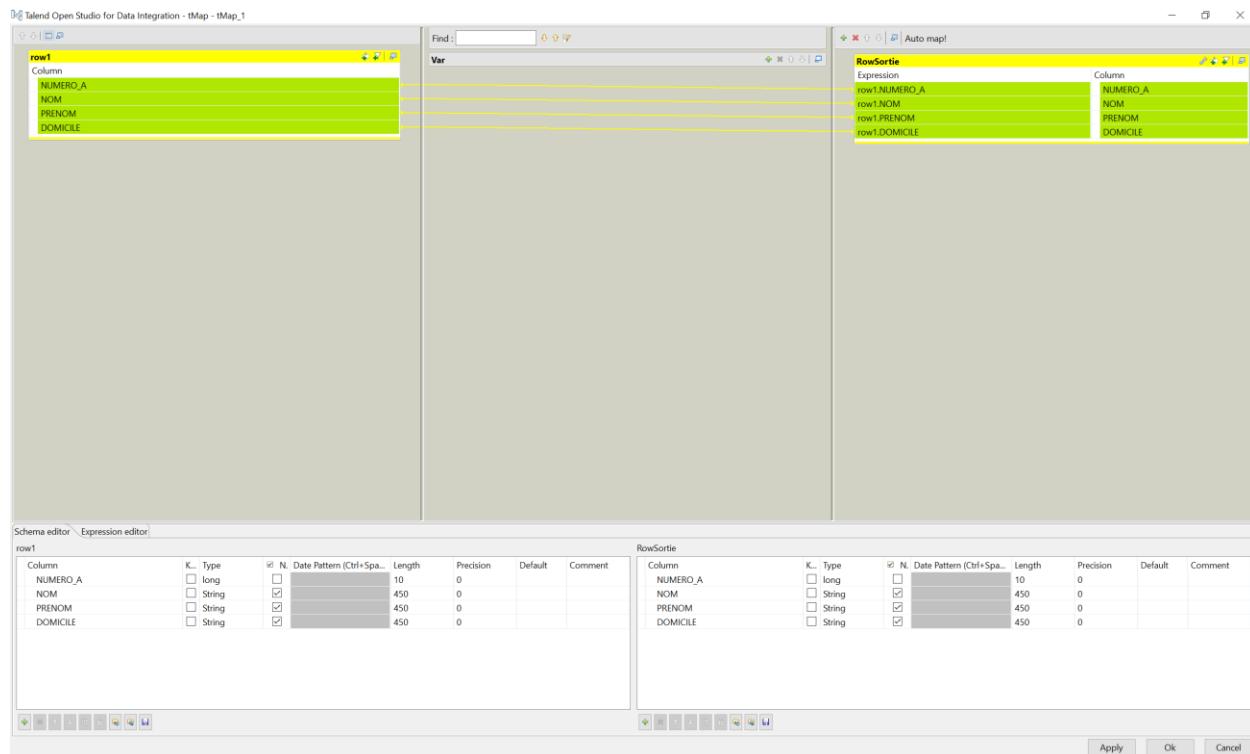


- Puis appuyer sur mapping auto en haut à droite

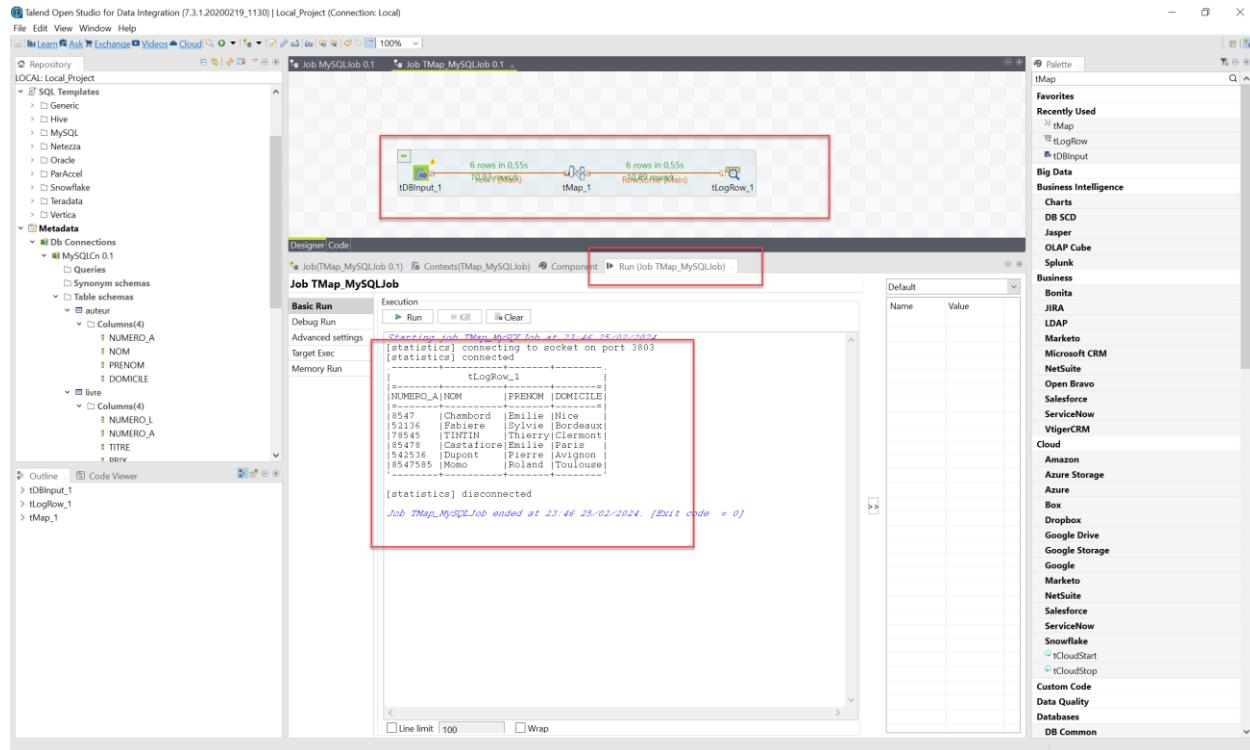
Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



- Une correspondance sera établie automatiquement

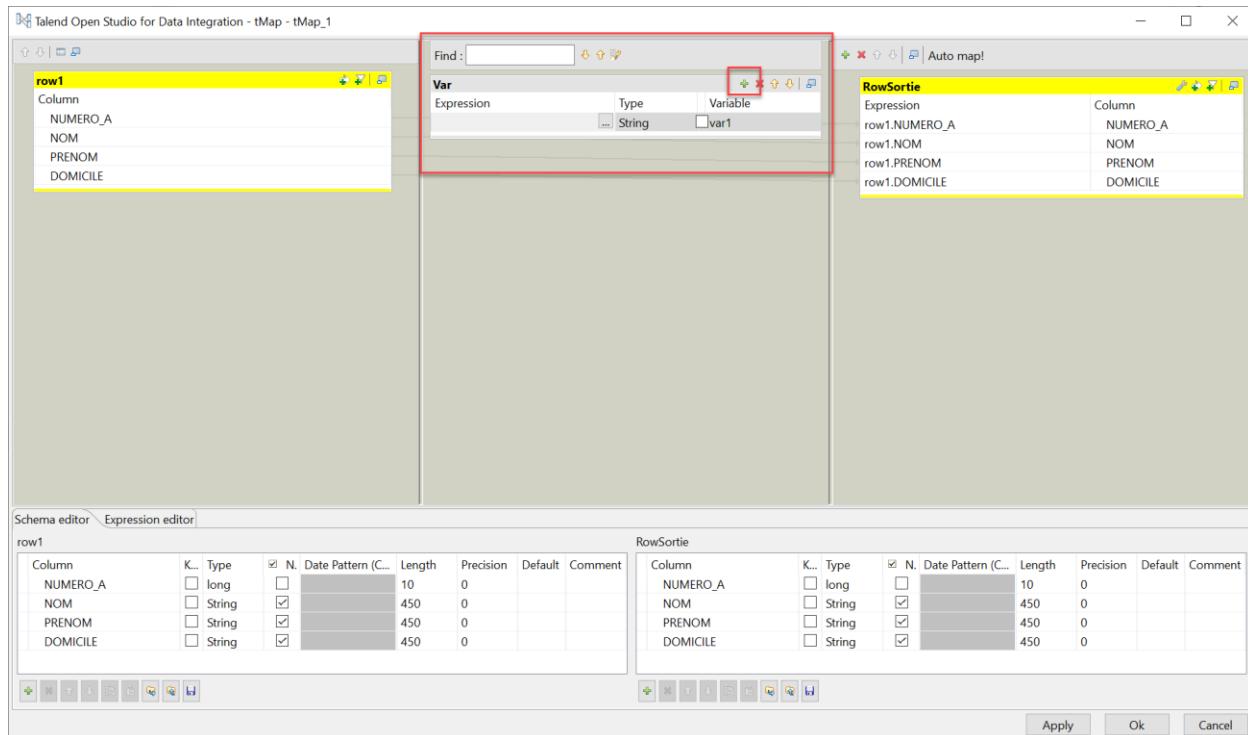


- Exécuter le job à nouveau (normalement il n'y a pas de changement)



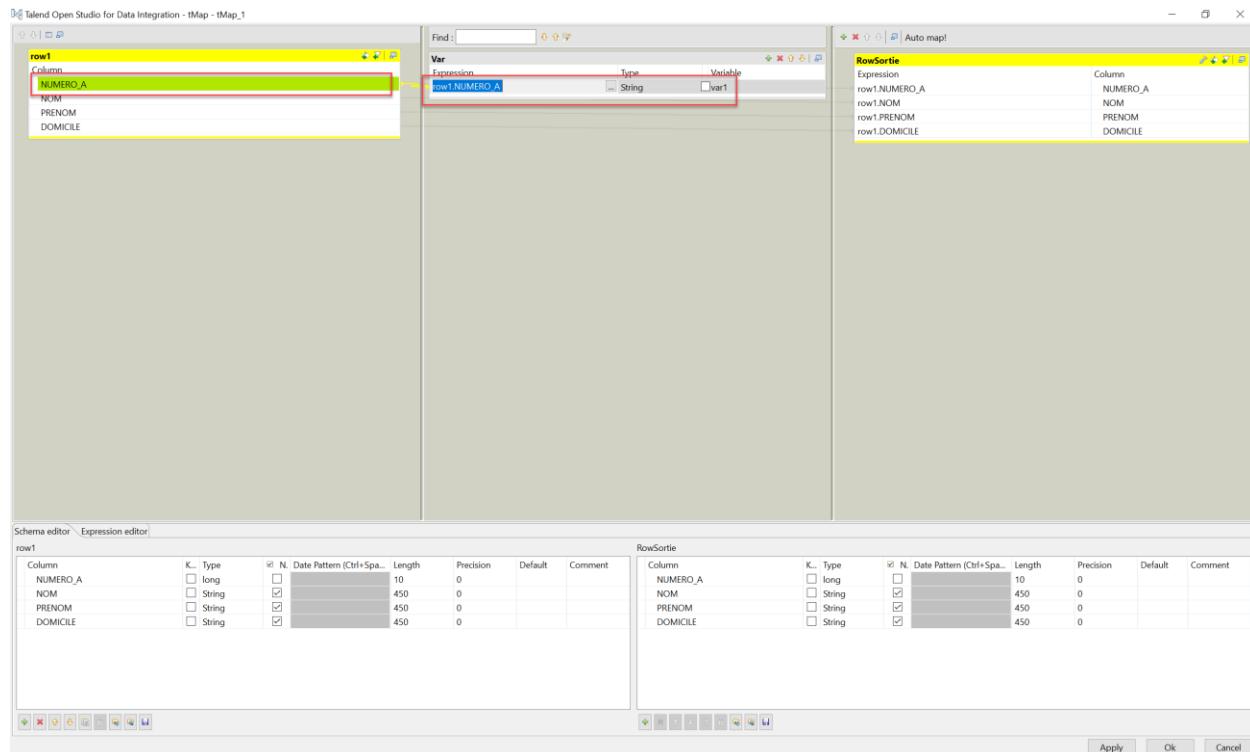
23. Modification de ce job :

- – Ouvrir à nouveau le tMap et créer le champ var1 dans la partie droite.



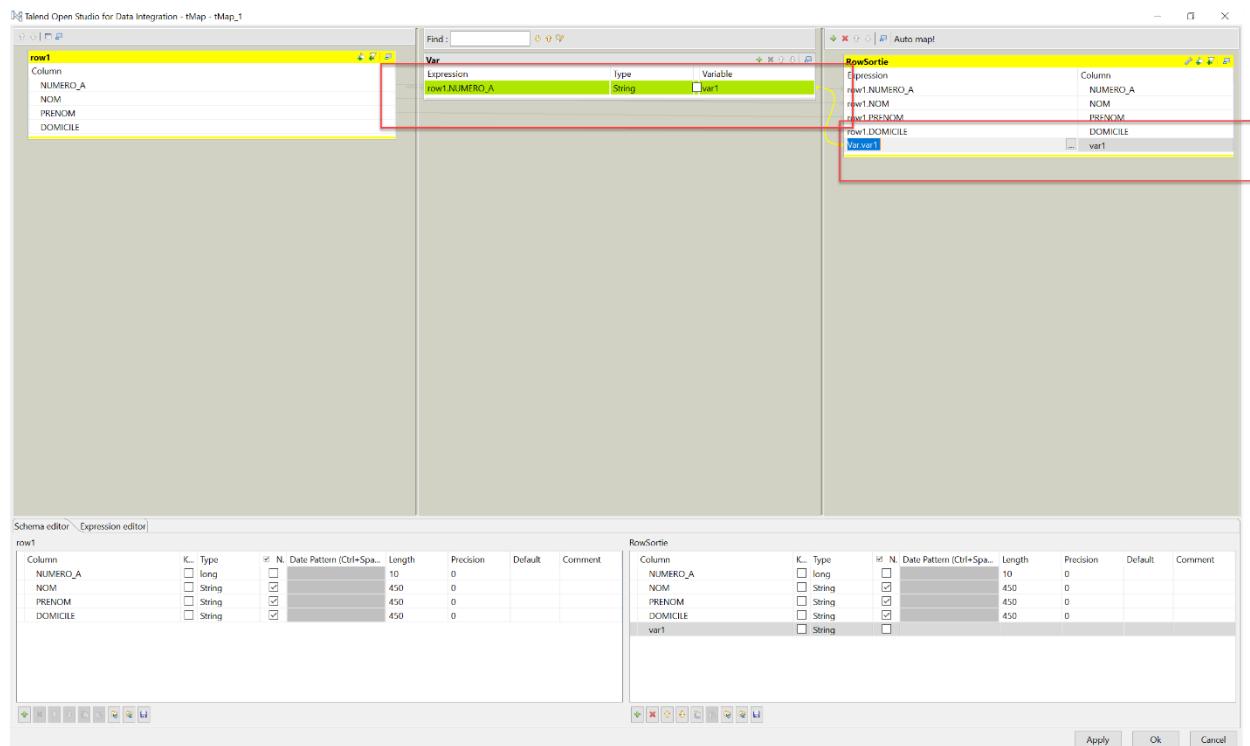
- Etablir une liaison entre la variable et Numéro en glissant le Numero_A vers la variable

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



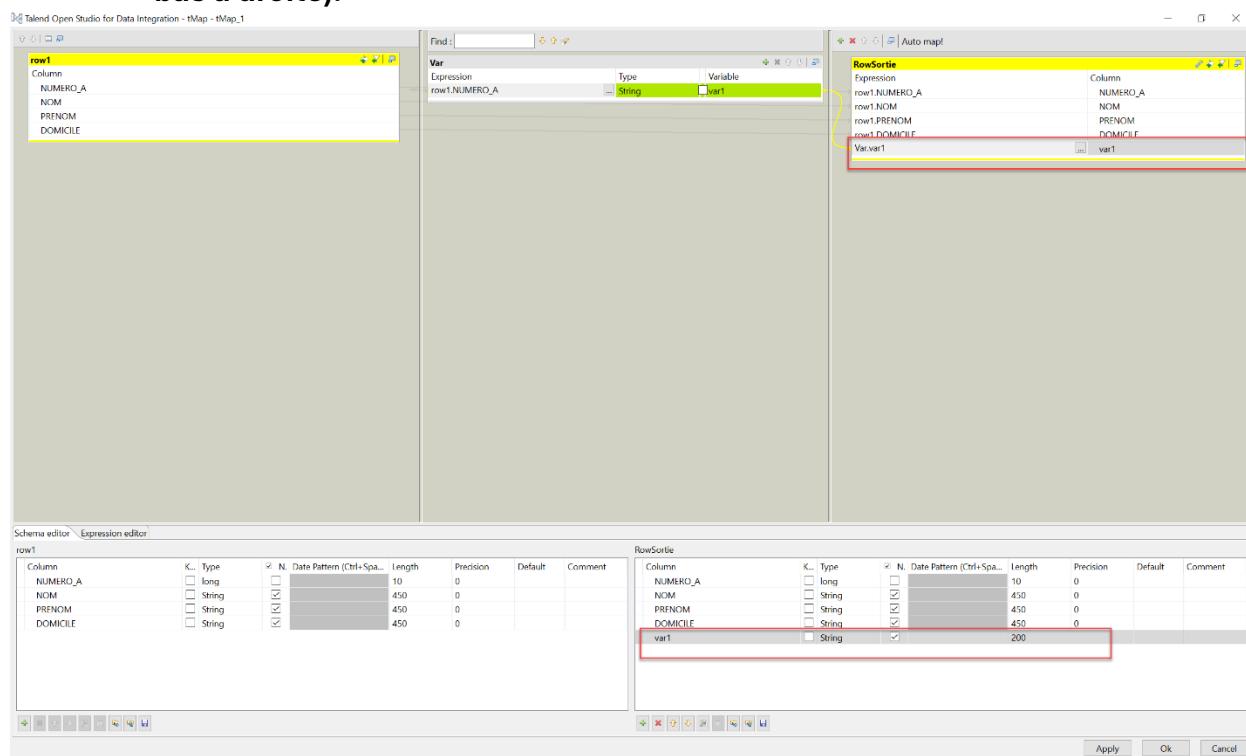
The screenshot shows the Talend Open Studio interface for a tMap component. The 'RowSortie' panel on the right displays a variable 'var1' mapped to the output columns: NUMERO_A, NOM, PRENOM, and DOMICILE. The 'Var' section in the center panel has 'row1.NUMERO_A' selected and mapped to 'var1'. The 'RowType' and 'RowSortie' schema editors at the bottom show the mapping details.

- Glisser cette variable vers la sortie



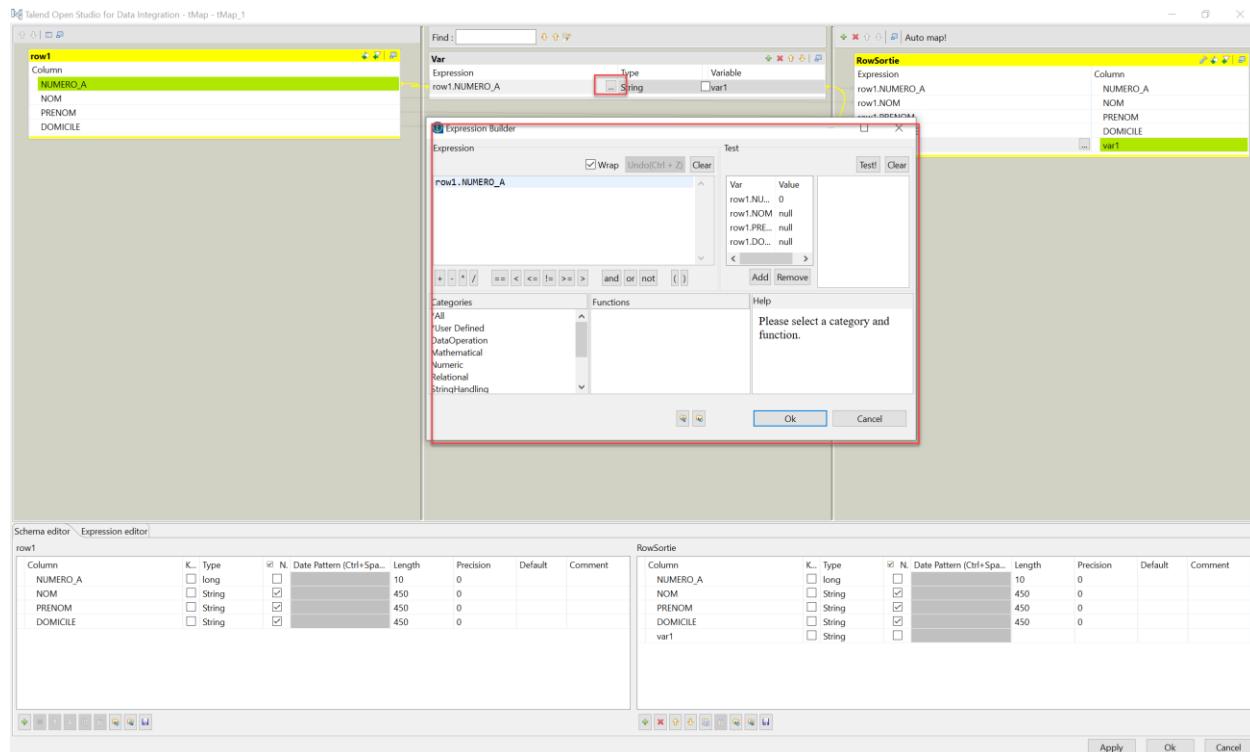
The screenshot shows the Talend Open Studio interface for a tMap component. The 'RowSortie' panel on the right now lists 'var1' as a mapped variable, indicating it has been successfully moved from the 'Var' section. The 'Var' section in the center panel still shows the original expression 'row1.NUMERO_A'.

- Cocher que celui-ci peut être null et qu'il peut avoir une taille de 200 (cf. partie en bas à droite).**

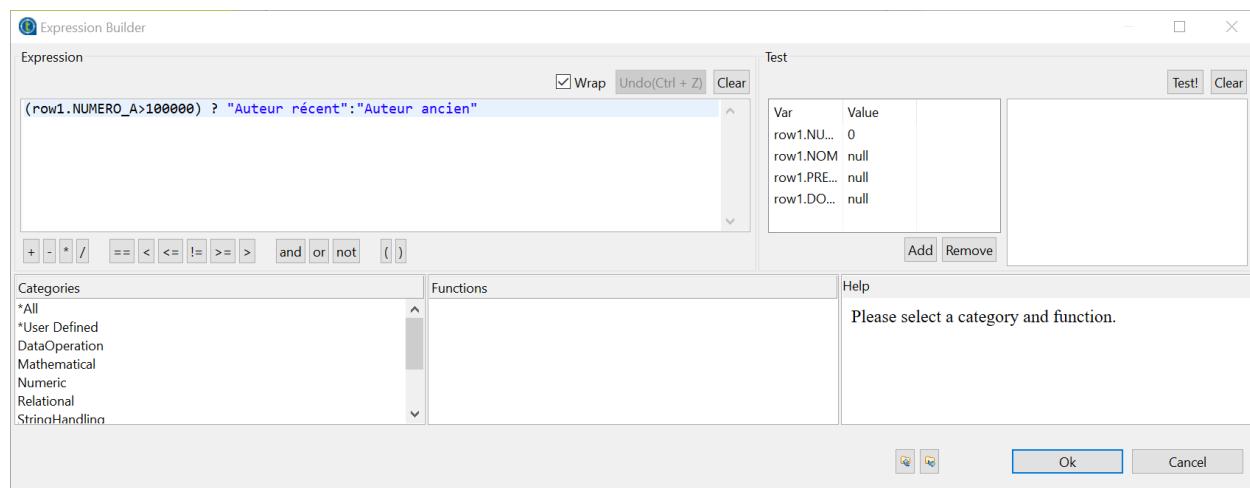


- Créer un nouveau champ dans la colonne du milieu puis cliquer sur les « ... » pour éditer l'expression :**

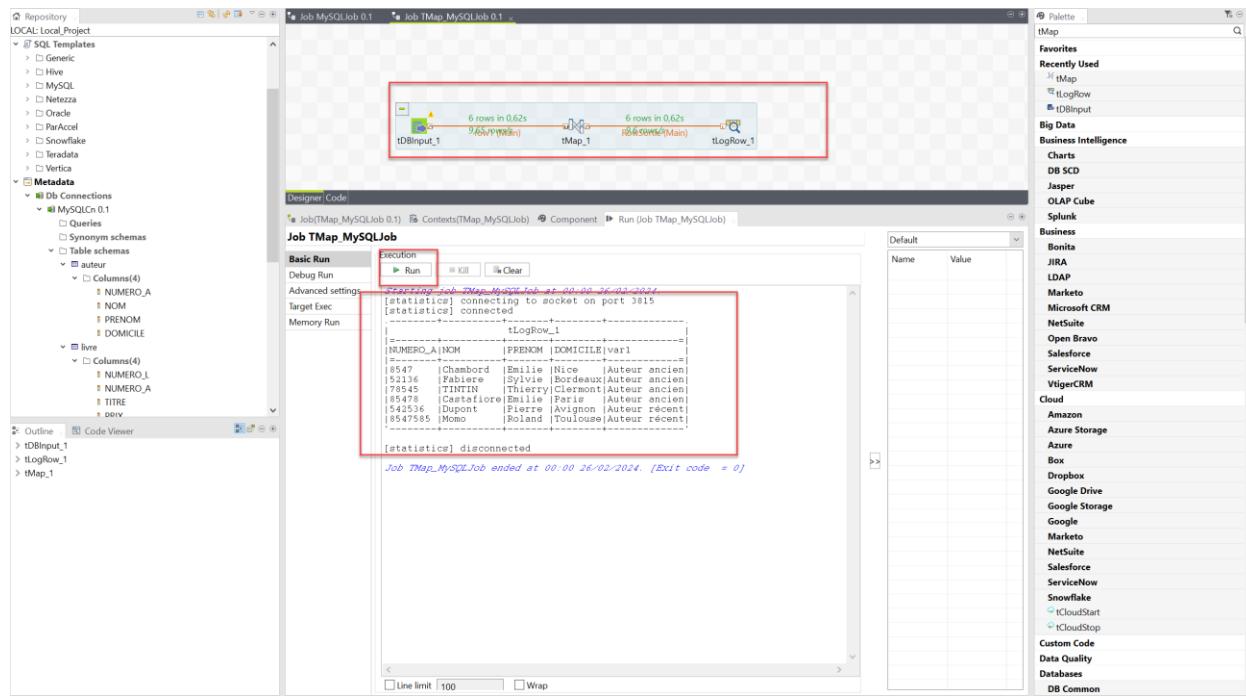
Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



- Ajouter le test ternaire dans le nouveau champ :
- `(row1.NUMERO_A>100000)?"Auteur récent": "Auteur ancien"`

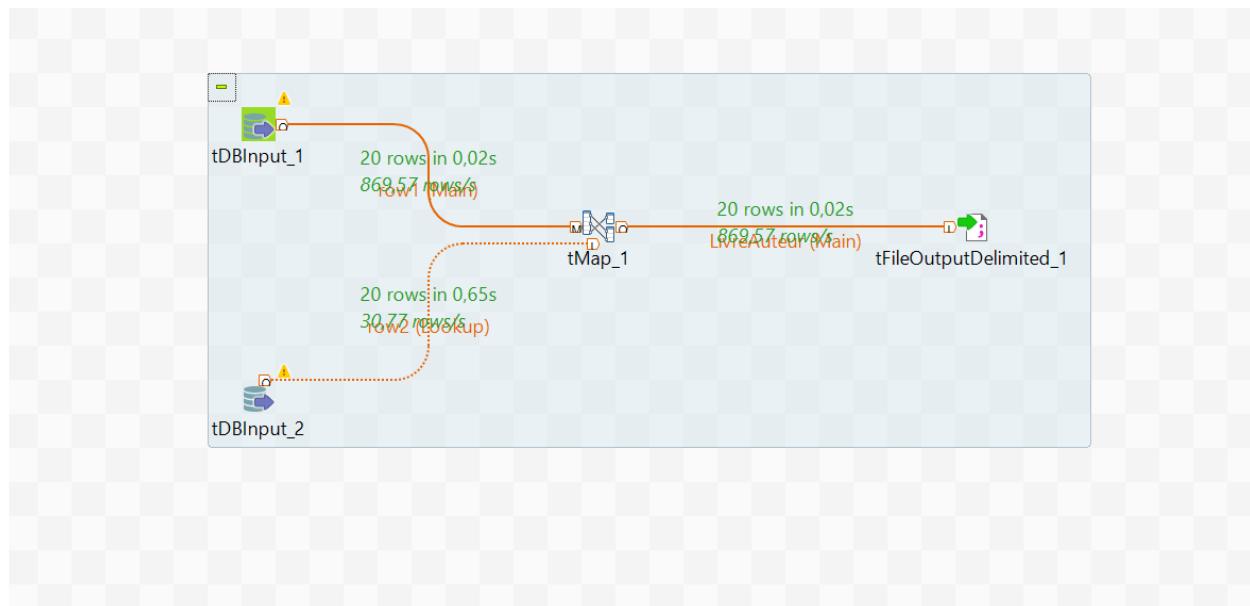


- Exécuter à nouveau le job, vous devriez obtenir ce type de résultat



Exercice à faire

L'objectif de cet exercice est de créer une ETL permettant de fusionner les tables "auteurs" et "livres" afin de stocker les données dans un fichier CSV nommé "BookAuteur.csv".



Votre job Talend devrait ressembler à quelque chose comme cela :

tMySQLInput ---(main)--> tMap ---(main)--> tFileOutputDelimited | ---(lookup)--> (Lookup sur la table auteur)

1. Creation de la base de données :

Pour créer une nouvelle base de données en utilisant le fichier SQL fourni, vous pouvez suivre les étapes suivantes :

- Modifier le fichier docker compose
- Exécution du script SQL : Copiez le contenu du fichier SQL dans un fichier nommé, par exemple, schema.sql.
- Utilisation de Docker pour MySQL : Utilisez Docker pour créer un conteneur MySQL dans lequel vous exécuterez le script SQL pour créer la nouvelle base de données et ses tables.

La nouvelle base de données à créer est décrite dans le fichier SQL fourni.

```
-- Create database
CREATE DATABASE IF NOT EXISTS `bookdb` DEFAULT CHARACTER SET utf8;
USE `bookdb`;

-- Table `auteur`
DROP TABLE IF EXISTS `auteur`;
CREATE TABLE `auteur` (
  `NUMERO_A` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NOM` varchar(450) DEFAULT NULL,
  `PRENOM` varchar(450) DEFAULT NULL,
  `DOMICILE` varchar(450) DEFAULT NULL,
  PRIMARY KEY (`NUMERO_A`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

-- Insert sample authors
INSERT INTO `auteur` (`NOM`, `PRENOM`, `DOMICILE`) VALUES
('Smith', 'John', 'New York'),
('Johnson', 'Emily', 'San Francisco'),
('Garcia', 'Miguel', 'Madrid'),
('Brown', 'Jessica', 'London'),
('Chen', 'Wei', 'Beijing'),
('Dubois', 'Marie', 'Paris'),
('Kim', 'Seong', 'Seoul'),
('Müller', 'Hans', 'Berlin'),
('Rossi', 'Marco', 'Rome'),
```

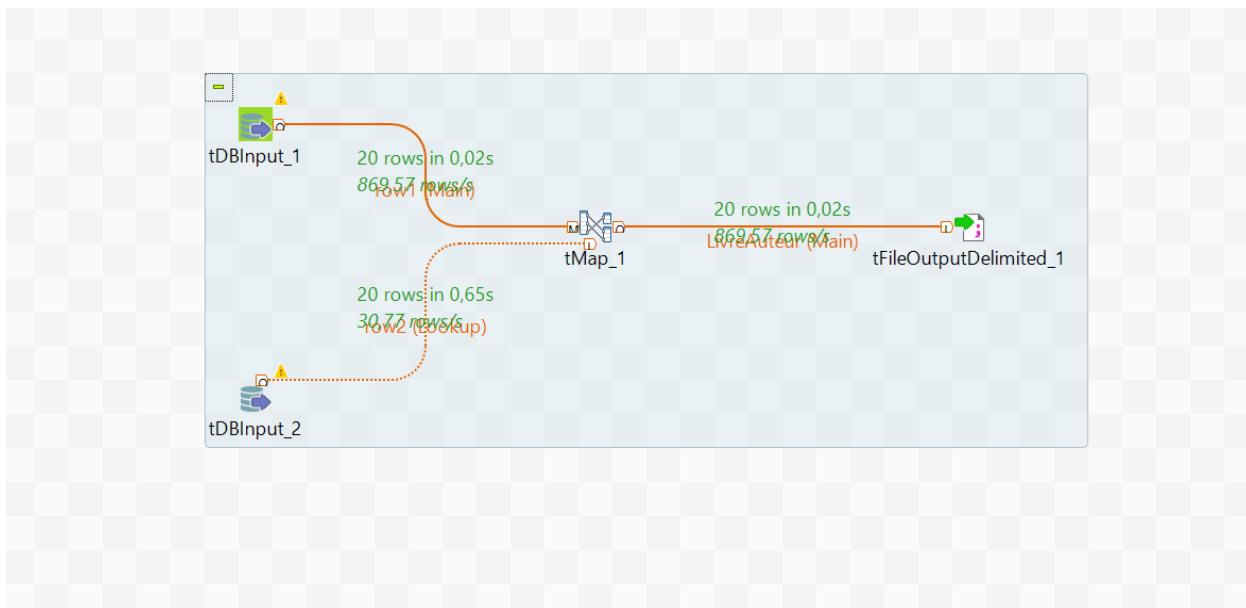
```
('Sato', 'Takashi', 'Tokyo');

-- Table `livre`
DROP TABLE IF EXISTS `livre`;
CREATE TABLE `livre` (
  `NUMERO_L` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NUMERO_A` int(10) unsigned NOT NULL,
  `TITRE` varchar(128) DEFAULT NULL,
  `PRIX` float DEFAULT NULL,
  PRIMARY KEY (`NUMERO_L`),
  CONSTRAINT `fk_auteur_numero` FOREIGN KEY (`NUMERO_A`) REFERENCES `auteur`(`NUMERO_A`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

-- Insert sample books
INSERT INTO `livre` (`NUMERO_A`, `TITRE`, `PRIX`) VALUES
(1, 'Data Engineering for Dummies', 19.99),
(2, 'Practical Data Engineering', 29.99),
(3, 'Mastering Data Engineering', 34.99),
(4, 'Advanced Techniques in Data Engineering', 39.99),
(5, 'Big Data Infrastructure', 49.99),
(6, 'Data Pipelines and ETL Processing', 24.99),
(7, 'Data Science Essentials', 29.99),
(8, 'Machine Learning in Practice', 39.99),
(9, 'Cloud Data Solutions', 44.99),
(10, 'Data Governance and Compliance', 34.99),
(1, 'Data Warehousing Fundamentals', 29.99),
(2, 'Real-time Analytics', 49.99),
(3, 'Data Quality Management', 39.99),
(4, 'Data Mining Techniques', 34.99),
(5, 'Introduction to Hadoop', 29.99),
(6, 'Spark in Action', 49.99),
(7, 'Python for Data Analysis', 39.99),
(8, 'R Programming for Data Science', 34.99),
(9, 'SQL Mastery', 29.99),
(10, 'NoSQL Databases', 49.99);
```

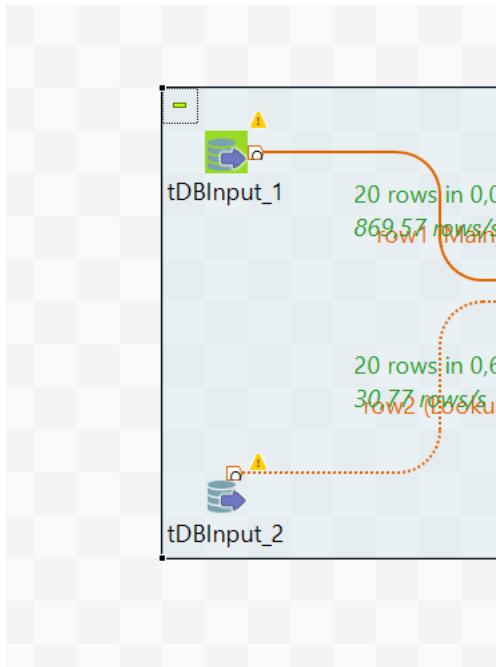
2. Creation de l'ETL :

Pour générer un fichier CSV avec les données fournies, vous pouvez utiliser Talend Data Integration. Voici comment vous pouvez le faire :



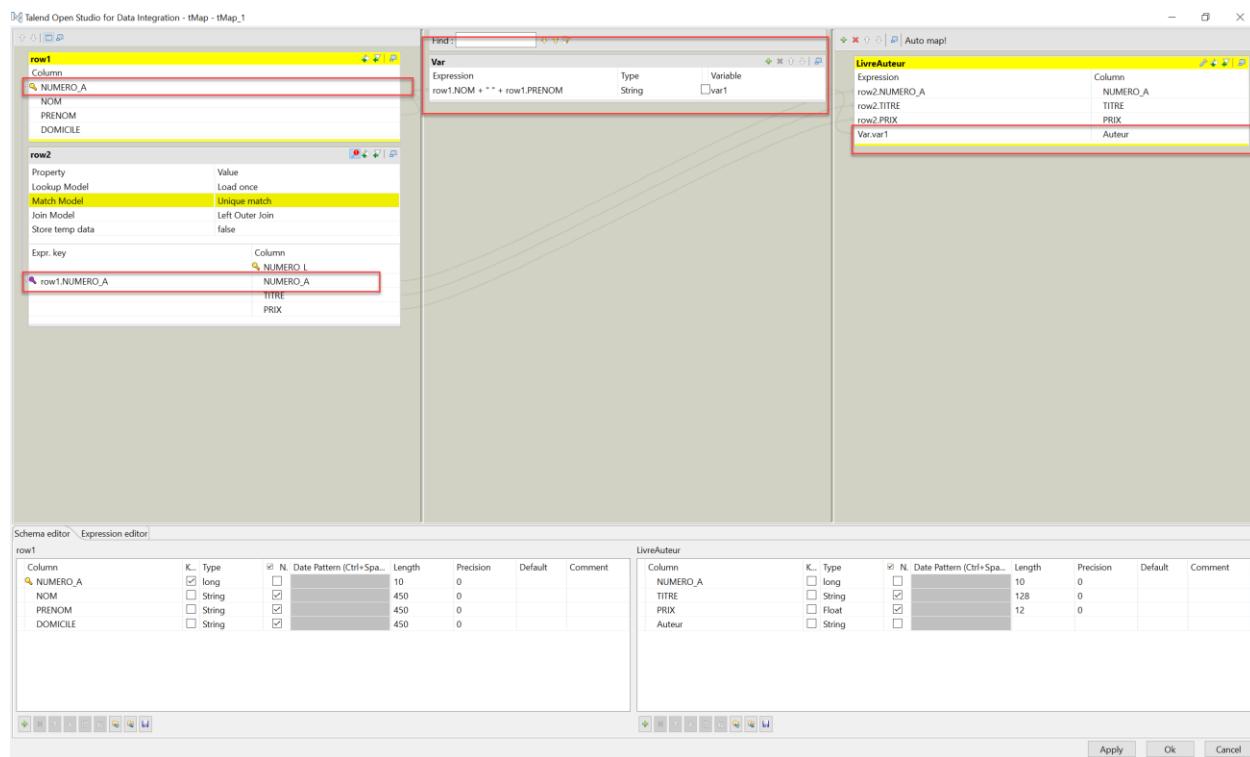
3. Extraction des données :

- Utilisez le composant **tMySQLInput** pour extraire les données des tables **livre** et **auteur** de la base de données **bookdb**.



4. Transformation des données :

- Dans le composant **tMap**, créez une colonne supplémentaire contenant le nom complet de l'auteur en concaténant les colonnes **NOM** et **PRENOM**.
- Réorganisez les colonnes selon le format requis pour le fichier CSV : **NUMERO_L, TITRE, PRIX, Auteur**.



Dans le composant **tMap**, vous pouvez utiliser une expression Java pour concaténer les colonnes **NOM** et **PRENOM**. Par exemple :

```
row1.NOM + " " + row1.PRENOM
```

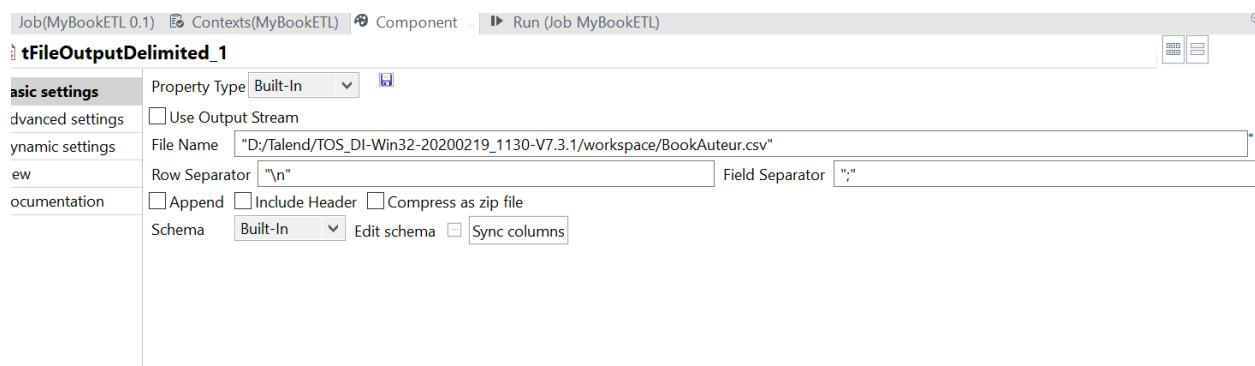
Assurez-vous de mapper les colonnes appropriées de la table **livre** et de la table **auteur** dans le composant **tMap**.

5. Chargement des données :

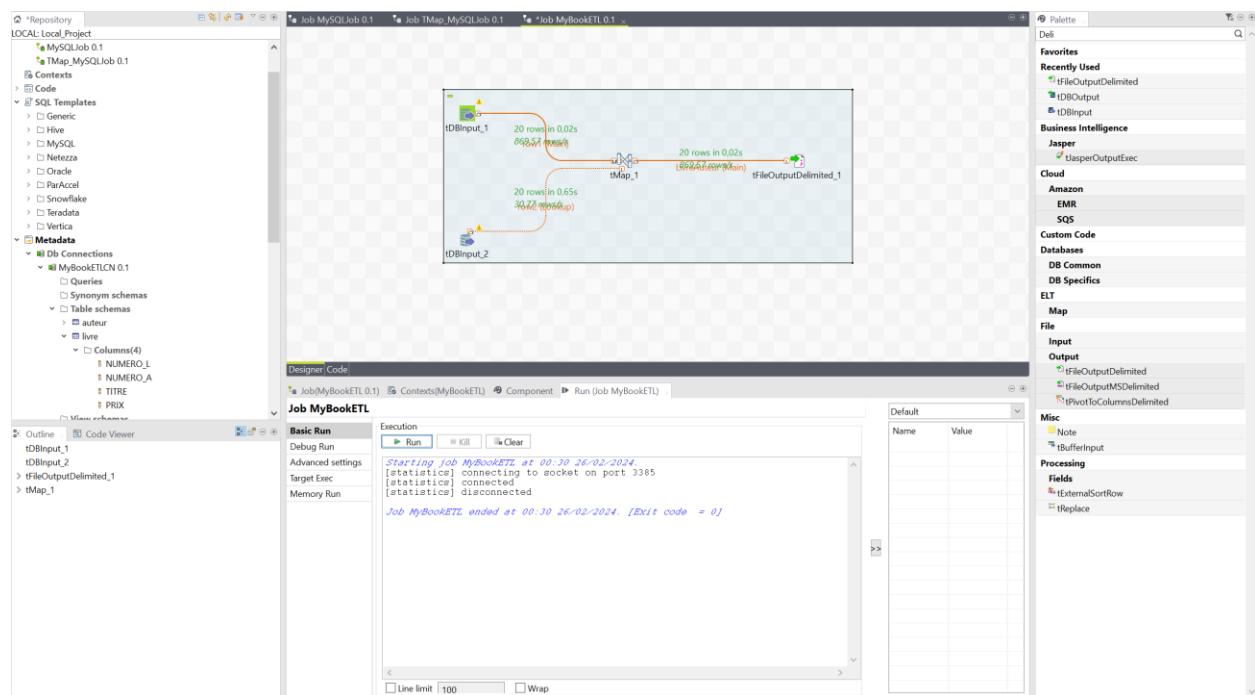
- Utilisez un composant **tFileOutputDelimited** pour écrire les données transformées dans un fichier CSV.

- Configurez le composant pour spécifier le chemin d'accès et le nom de fichier pour le fichier CSV de sortie.

Enfin, dans le composant **tFileOutputDelimited**, configurez le chemin d'accès et le nom de fichier pour votre fichier CSV de sortie, ainsi que les délimiteurs appropriés pour séparer les champs.



Exécuter le job et vérifier par la suite la bonne exécution ;



En suivant ces étapes, vous pourrez générer un fichier CSV avec les données fournies,



Talend for Data Integration



Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

BookAuteur.csv - Saved to this PC

File Home Insert Page Layout Formulas Data Review View Automate Help Analytic Solver Data Mining

Cut Copy Format Painter

Font Alignment Number Styles

Conditional Formatting

Cells Editing

Sort & Filter

Add-ins Analyze Data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	1	Data Ware	29.99	Smith John																						
2	2	Real-time	49.99	Johnson Emily																						
3	3	Data Qual	39.99	Garcia Miguel																						
4	4	Data Miner	34.99	Brown Jessica																						
5	5	Introductic	29.99	Chen Wei																						
6	6	Spark in A	49.99	Dubois Marie																						
7	7	Python for	39.99	Kim Seong																						
8	8	R Program	34.99	Mller Hans																						
9	9	SQL Mast	29.99	Rossi Marco																						
10	10	NoSQL Da	49.99	Sato Takashi																						
11	0			Smith John																						
12	0			Johnson Emily																						
13	0			Garcia Miguel																						
14	0			Brown Jessica																						
15	0			Chen Wei																						
16	0			Dubois Marie																						
17	0			Kim Seong																						
18	0			Mller Hans																						
19	0			Rossi Marco																						
20	0			Sato Takashi																						
21																										
22																										
23																										
24																										
25																										
26																										
27																										
28																										
29																										
30																										
31																										
32																										
33																										
34																										
35																										
36																										
37																										
38																										
39																										

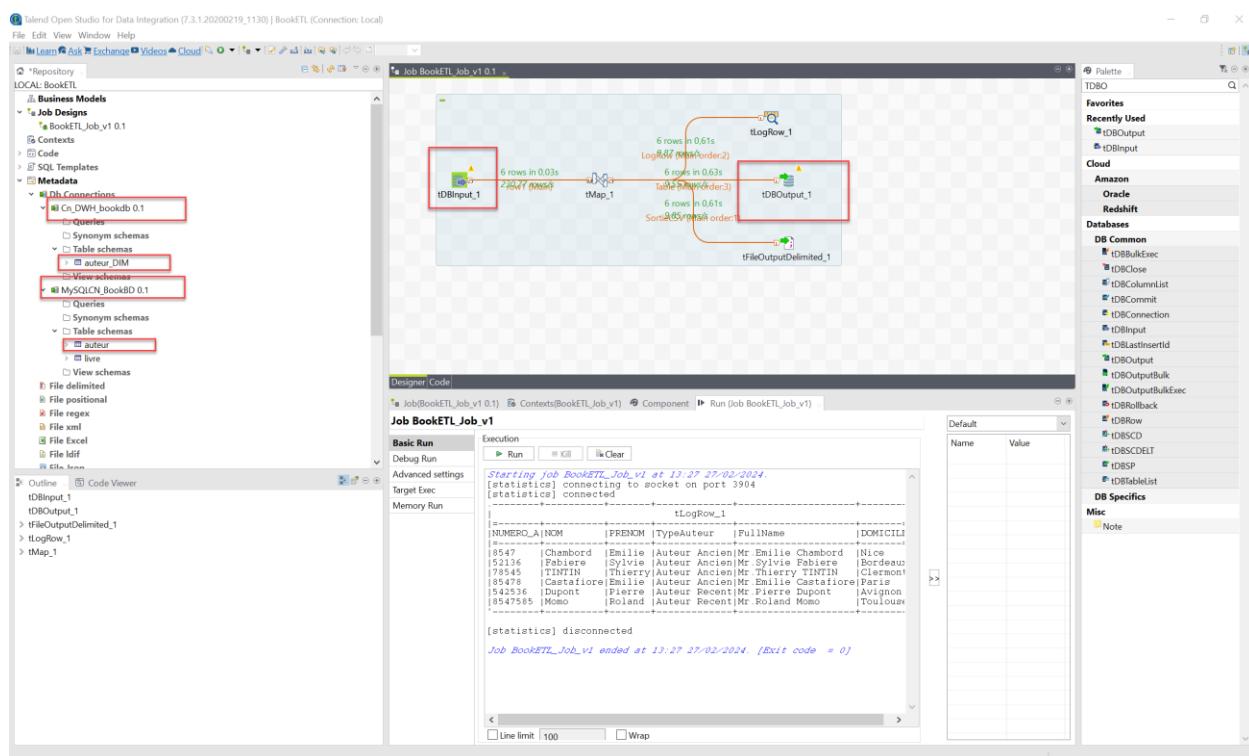
BookAuteur

Ready Accessibility: Unavailable

Exercice 2 :

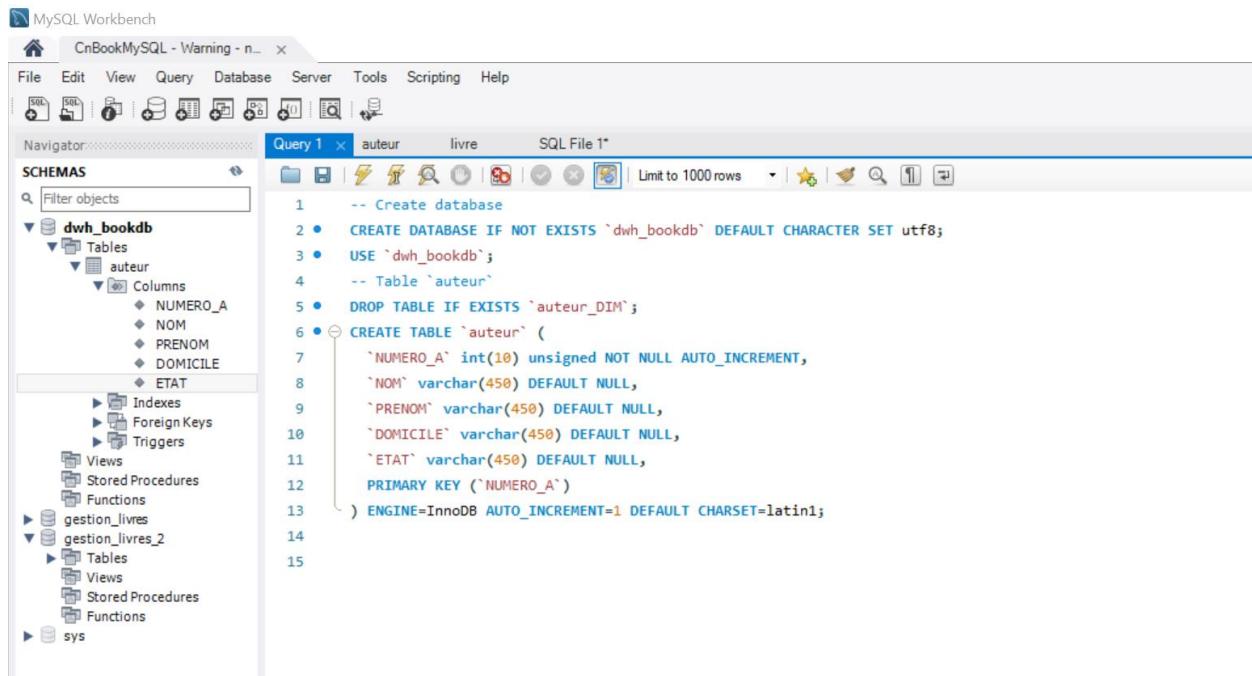
Chargement des données dans une bases de données MSYQL

Dans cet exemple on duplique le job et on ajoute un autre composant pour insérer des données dans la base de données cible



6. Préparation de la bases cibles :

Créer la base de données Target « dwh_books » avec le script suivant en utilisant workbench ou Docker Desktop

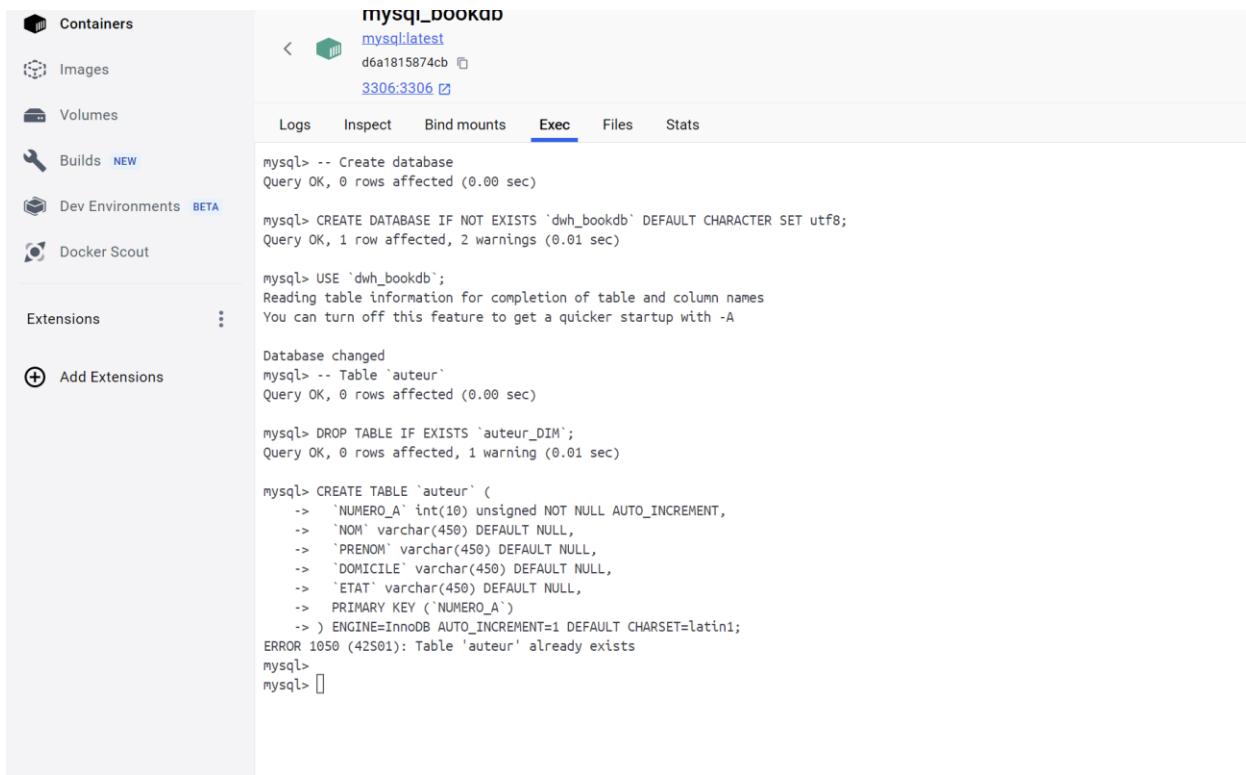


```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator: CnBookMySQL - Warning - n... ×
SCHEMAS
Filter objects
dwh_bookdb
  Tables
    auteur
      Columns
        NUMERO_A
        NOM
        PRENOM
        DOMICILE
        ETAT
      Indexes
      ForeignKeys
      Triggers
    Views
    Stored Procedures
    Functions
  gestion_livres
  gestion_livres_2
    Tables
    Views
    Stored Procedures
    Functions
sys
Query 1 × auteur livre SQL File 1*
1 -- Create database
2 • CREATE DATABASE IF NOT EXISTS `dwh_bookdb` DEFAULT CHARACTER SET utf8;
3 • USE `dwh_bookdb`;
4 -- Table `auteur`
5 • DROP TABLE IF EXISTS `auteur_DIM`;
6 • CREATE TABLE `auteur` (
7   `NUMERO_A` int(10) unsigned NOT NULL AUTO_INCREMENT,
8   `NOM` varchar(450) DEFAULT NULL,
9   `PRENOM` varchar(450) DEFAULT NULL,
10  `DOMICILE` varchar(450) DEFAULT NULL,
11  `ETAT` varchar(450) DEFAULT NULL,
12  PRIMARY KEY (`NUMERO_A`)
13 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
14
15

```

Avec Docker

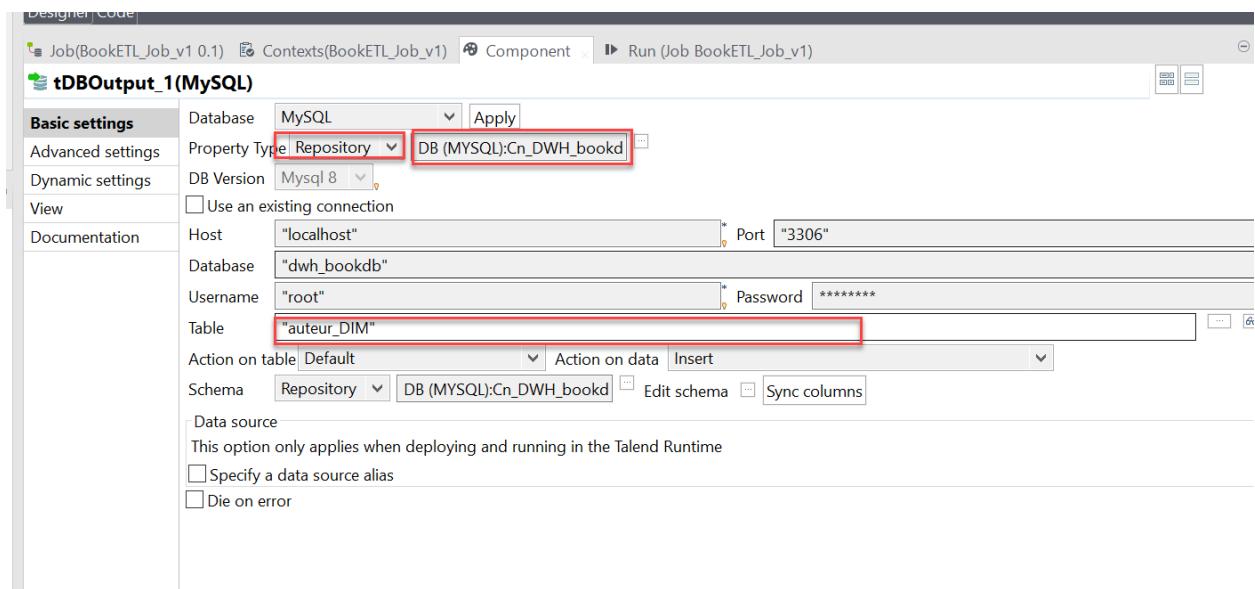


```
-- Create database
CREATE DATABASE IF NOT EXISTS `dwh_bookdb` DEFAULT CHARACTER SET utf8;
USE `dwh_bookdb`;
-- Table `auteur`
DROP TABLE IF EXISTS `auteur_DIM`;
CREATE TABLE `auteur` (
  `NUMERO_A` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NOM` varchar(450) DEFAULT NULL,
  `PRENOM` varchar(450) DEFAULT NULL,
  `DOMICILE` varchar(450) DEFAULT NULL,
  `ETAT` varchar(450) DEFAULT NULL,
  PRIMARY KEY (`NUMERO_A`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
ERROR 1050 (42S01): Table 'auteur' already exists
mysql>
mysql> []
```

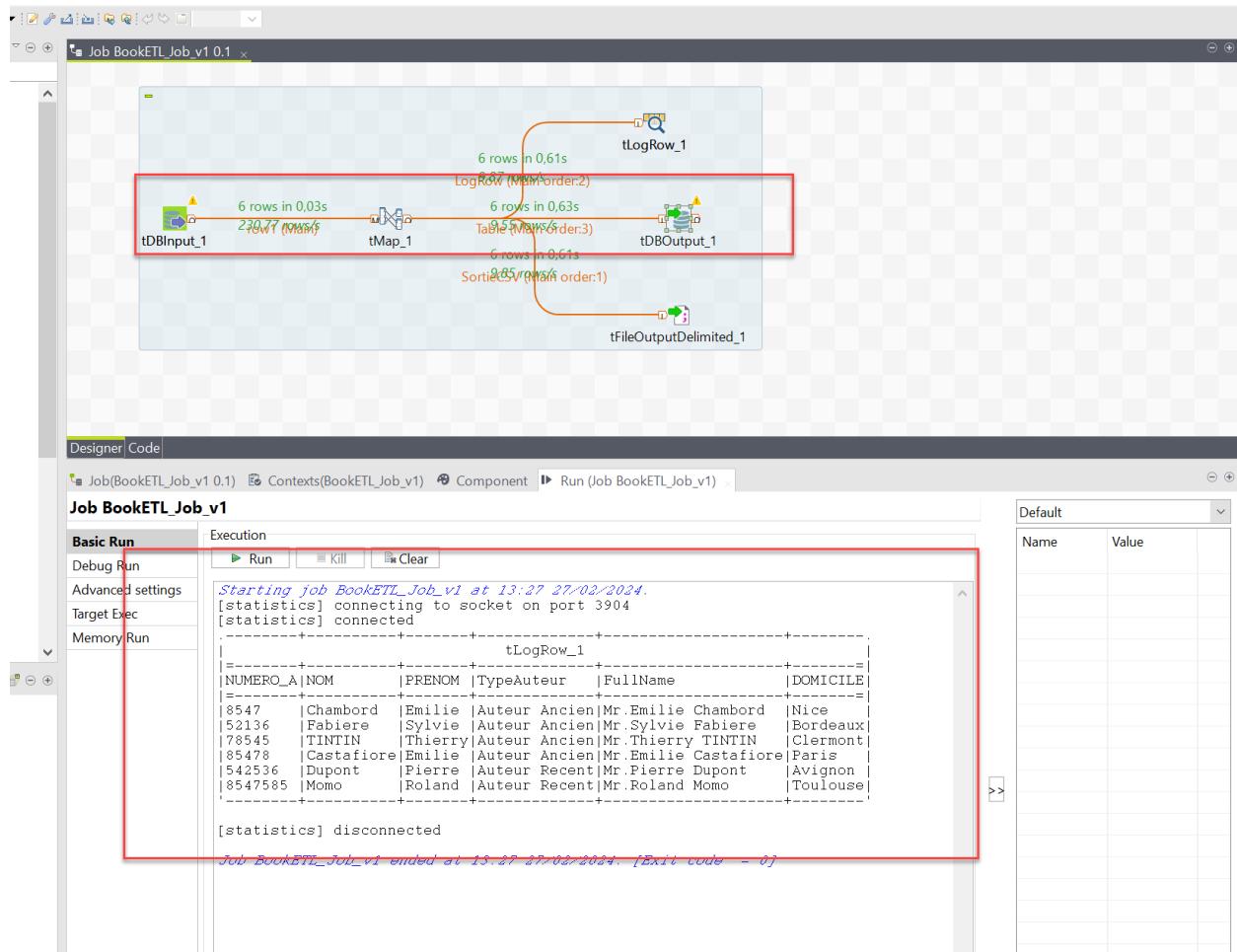
1. Chargement des données dans une table “Auteur_DIM”:

- Utilisez un composant **tDBOutput** pour écrire les données transformées dans un une table mysql « auteur_dim » ».

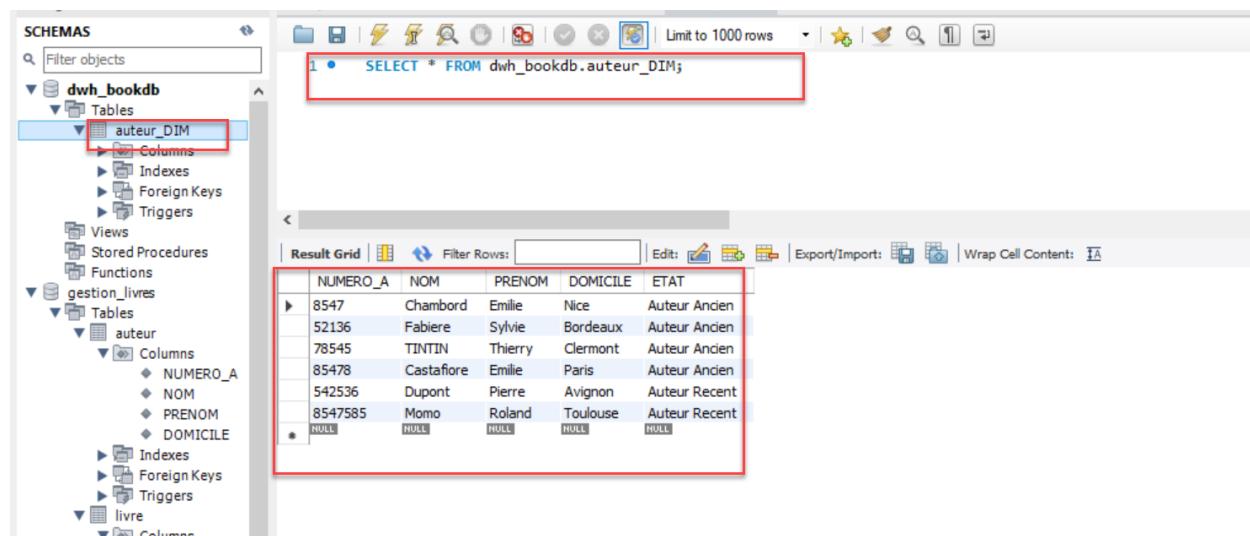
Enfin, dans le composant **tDBOutput**, configurez les paramètres de l'accès à la bases de données et de sortie, ainsi que le nom de la table « Auteur_DIM » appropriée.



Exécuter le job et vérifier par la suite la bonne exécution ;

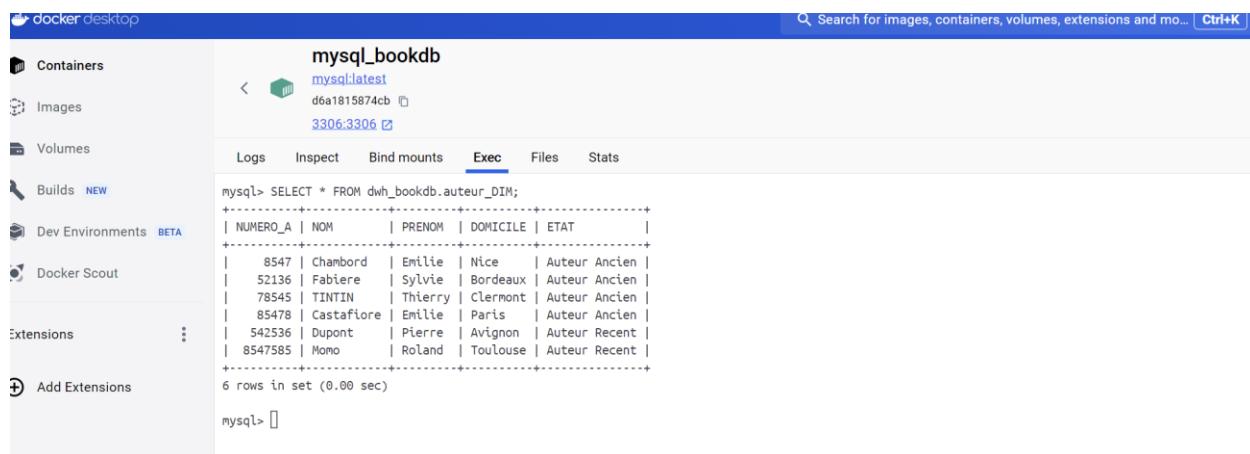


En suivant ces étapes, vous pourrez charger dans la table « Auteur_DIM » avec les données fournies, avec Workbench ou Docker Desktop « Exec »



The screenshot shows the Talend Studio interface. On the left, the 'SCHEMAS' tree view displays two databases: 'dwh_bookdb' and 'gestion_livres'. Under 'dwh_bookdb', the 'Tables' section contains 'auteur_DIM' (highlighted with a red box), 'Columns', 'Indexes', 'Foreign Keys', and 'Triggers'. Under 'gestion_livres', it contains 'livre' and 'auteur' (which also has 'Columns', 'Indexes', 'Foreign Keys', and 'Triggers'). On the right, a query editor window shows a single SQL statement: '1 • SELECT * FROM dwh_bookdb.auteur_DIM;'. Below the query is a 'Result Grid' displaying the data from the 'auteur_DIM' table:

	NUMERO_A	NOM	PRENOM	DOMICILE	ETAT
▶	8547	Chambord	Emilie	Nice	Auteur Ancien
	52136	Fabiere	Sylvie	Bordeaux	Auteur Ancien
	78545	TINTIN	Thierry	Clermont	Auteur Ancien
	85478	Castafiore	Emilie	Paris	Auteur Ancien
	542536	Dupont	Pierre	Avignon	Auteur Recent
	8547585	Momo	Roland	Toulouse	Auteur Recent
●	NULL	NULL	NULL	NULL	NULL



The screenshot shows the Docker Desktop interface. On the left, there's a sidebar with 'Containers', 'Images', 'Volumes', 'Builds', 'Dev Environments', 'Docker Scout', and 'Extensions'. The main area shows a MySQL container named 'mysql_bookdb' (image: mysql:latest, ID: d6a1815874cb, port: 3306:3306). The 'Logs' tab is active, showing the following SQL query and its results:

```
mysql> SELECT * FROM dwh_bookdb.auteur_DIM;
+-----+-----+-----+-----+-----+
| NUMERO_A | NOM | PRENOM | DOMICILE | ETAT |
+-----+-----+-----+-----+-----+
| 8547 | Chambord | Emilie | Nice | Auteur Ancien |
| 52136 | Fabiere | Sylvie | Bordeaux | Auteur Ancien |
| 78545 | TINTIN | Thierry | Clermont | Auteur Ancien |
| 85478 | Castafiore | Emilie | Paris | Auteur Ancien |
| 542536 | Dupont | Pierre | Avignon | Auteur Recent |
| 8547585 | Momo | Roland | Toulouse | Auteur Recent |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Job ETL_FACT_Empreunts

Objectif

Ce document décrit l'ajout de filtres, tris et agrégations dans le flux d'intégration des emprunts afin de générer la table de faits Emprunt_FACT.

Source

Fichier : Emprunts.xlsx

Colonnes :

- NUMERO_E (Identifiant Emprunt)
 - NUMERO_AB (Identifiant Abonné)
 - NUMERO_L (Identifiant Livre)
 - DATE_EMPRUNT
 - DATE_RETUR
-

Composants Talend à ajouter

1. tFilterRow — Validation des emprunts valides

But : Éliminer les lignes où la date de retour est antérieure à la date d'emprunt.

- **Position dans le flux :** après tFileInputExcel
 - **Configuration :**
 - Condition :
 - Colonne : DATE_RETUR
 - Fonction : Greater than or equal
 - Colonne de référence : DATE_EMPRUNT
-

2. tSortRow — Tri des données

But : Ordonner les emprunts par abonné, puis par date d'emprunt.

- **Position dans le flux :** après tFilterRow
 - **Configuration :**
 - Sort key 1 : NUMERO_AB, ordre croissant
 - Sort key 2 : DATE_EMPRUNT, ordre croissant
-

3. tAggregateRow — Agrégation des emprunts

But : Calculer le nombre d'emprunts identiques (même abonné, livre, date).

- **Position :** après tSortRow
- **Group By :**
 - NUMERO_AB
 - NUMERO_L
 - DATE_EMPRUNT
 - DATE_RETOUR
- **Operations :**
 - Nom : NombreEmprunts
 - Fonction : count
 - Champ : NUMERO_E (ou toute colonne non nulle)

4. tMap — Génération des clés calendrier

But : Générer les clés DateEmpruntKey et DateRetourKey au format yyyyMMdd

- **Expressions :**
 - DateEmpruntKey = Integer.parseInt(TalendDate.formatDate("yyyyMMdd", row1.DATE_EMPRUNT))
 - DateRetourKey = Integer.parseInt(TalendDate.formatDate("yyyyMMdd", row1.DATE_RETOUR))

5. tOutputMySQL — Chargement vers Emprunt_FACT

Champs insérés :

Colonne cible Source (via tMap + agrégation)

NUMERO_AB row1.NUMERO_AB

NUMERO_L row1.NUMERO_L

DateEmpruntKey champ calculé

Colonne cible Source (via tMap + agrégation)

DateRetourKey champ calculé

NombreEmprunts résultat de tAggregateRow

Job ETL_FACT_Emprunts

Objectif

Ce document décrit l'ajout de filtres, tris et agrégations dans le flux d'intégration des emprunts afin de générer la table de faits Emprunt_FACT.

 **Source**

Fichier : Emprunts.xlsx

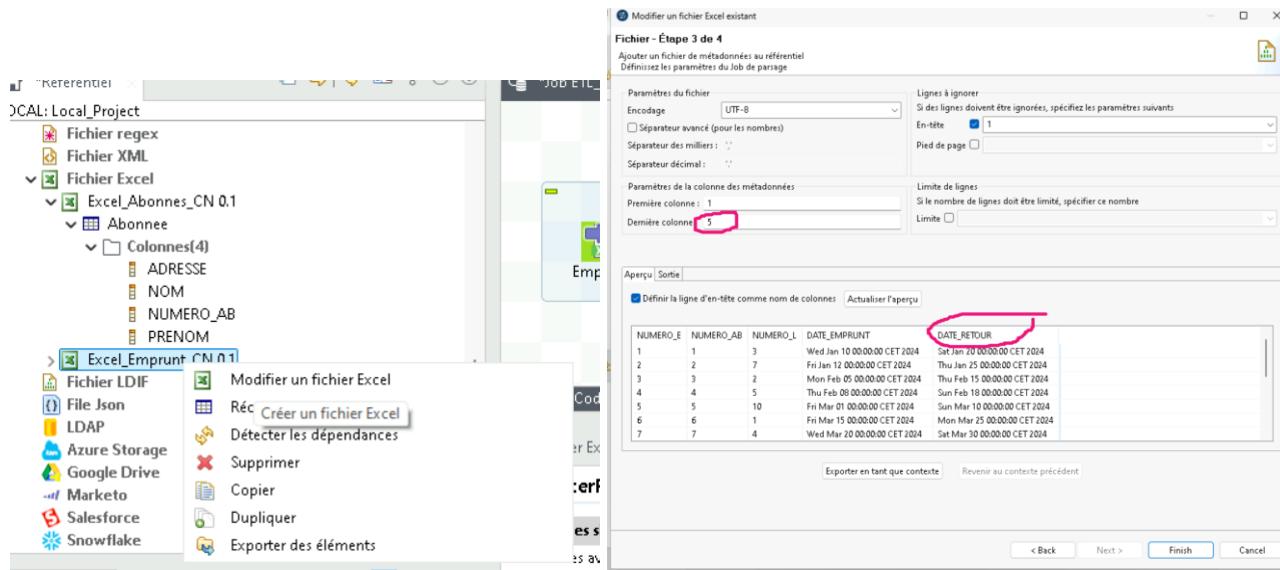
Colonnes :

- NUMERO_E (Identifiant Emprunt)
- NUMERO_AB (Identifiant Abonné)
- NUMERO_L (Identifiant Livre)
- DATE_EMPRUNT
- DATE_RETUR

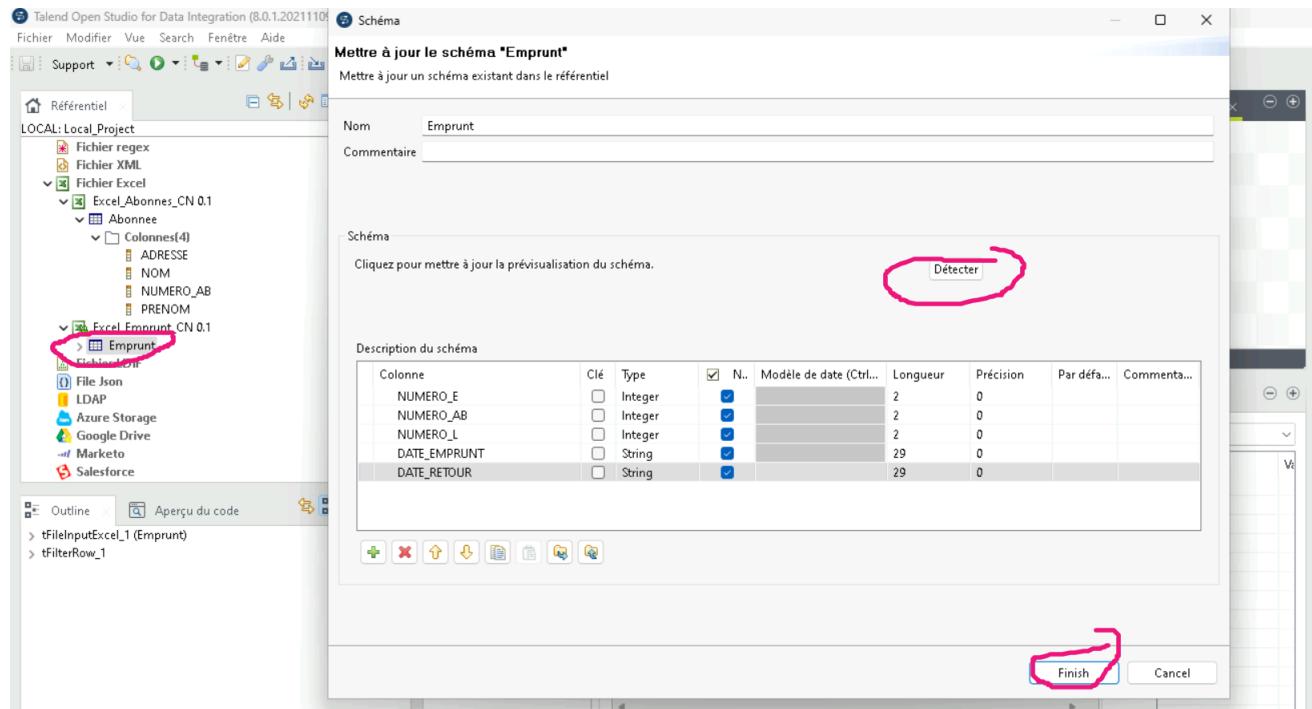
Editer la connexion au fichier Excel Emprunt pour intégrer la colonnes DATE_RETUR

- Cliquer avec le bouton droite su Excel_Emprunt -> Modifier un Fichier Excel

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



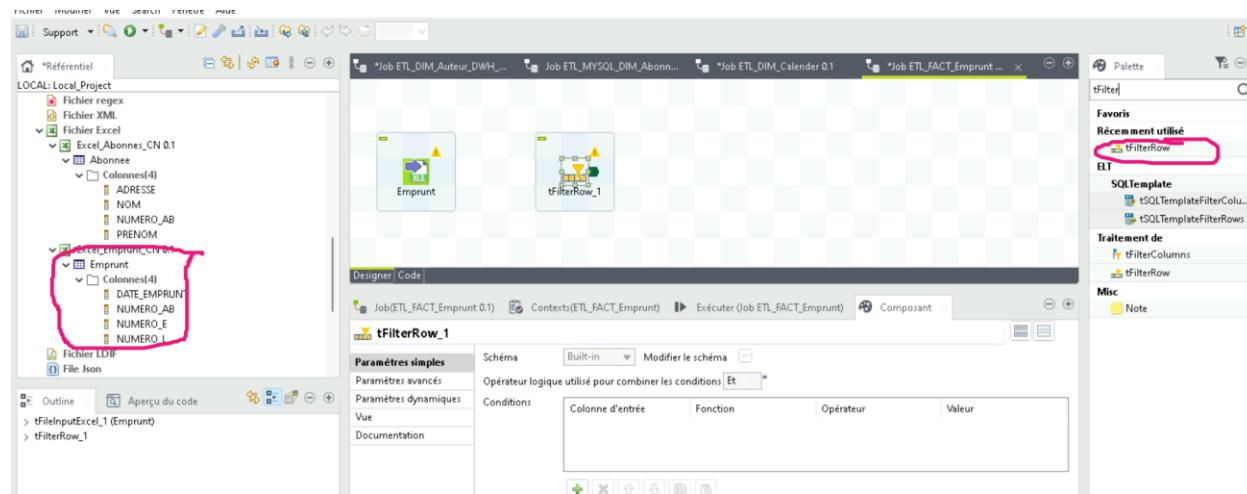
- Modifier le Schema pour ajouter la colonne Date Retour en cliquer avec bouton droit Modifier Schema



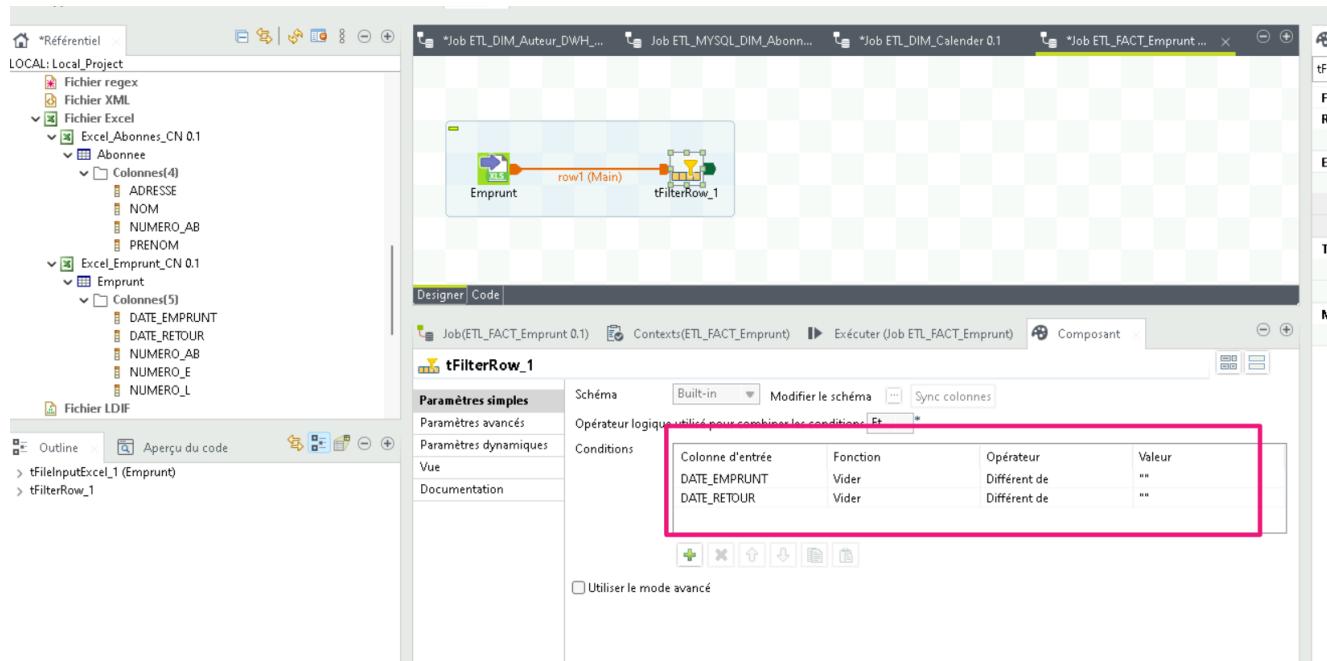
🔧 Composants Talend à ajouter

1. tFilterRow — Validation des emprunts valides

But : Éliminer les lignes où la date de retour est antérieure à la date d'emprunt.



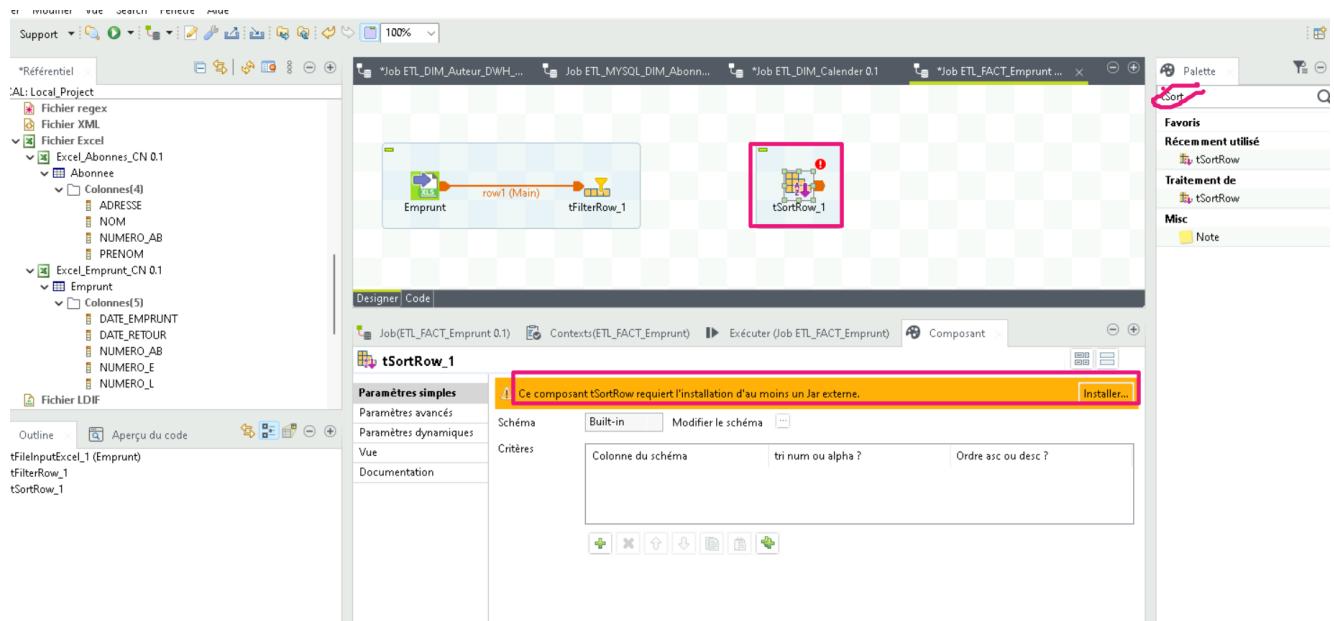
- **Position dans le flux :** après tFileInputExcel
- **Configuration :**
 - Conditionw :
 - Colonne : DATE_RETOUR et DATE_EMPRUNT
 - Fonction : Différent de
 - Valeur : “ ”



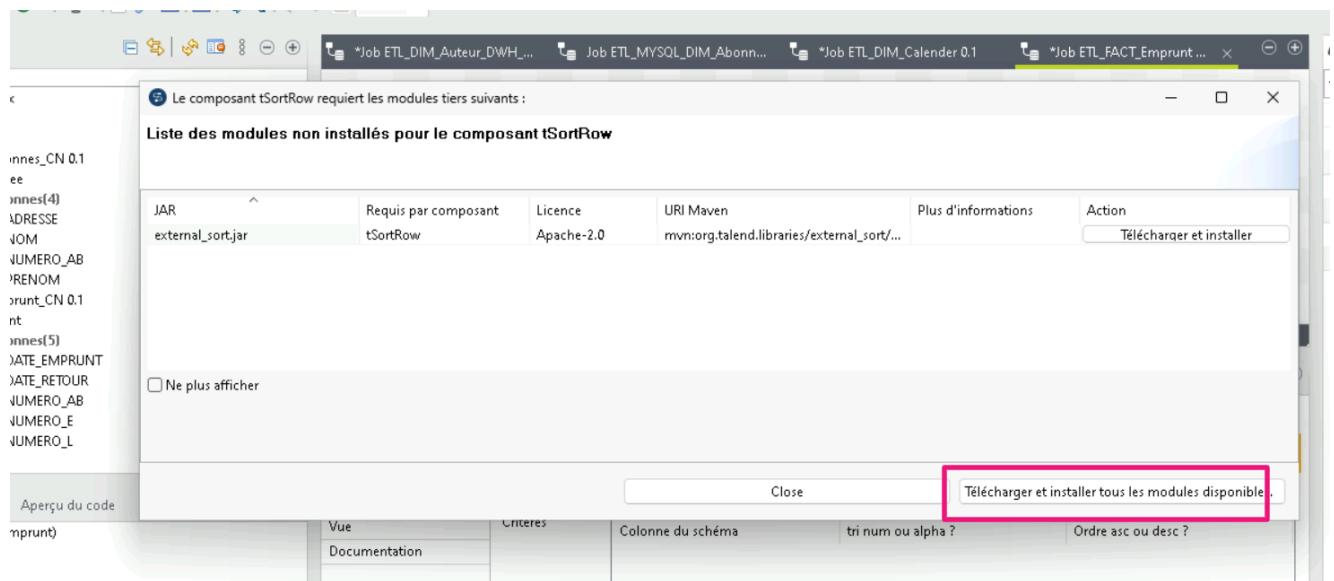
2. tSortRow — Tri des données

But : Ordonner les emprunts par abonné, puis par date d'emprunt.

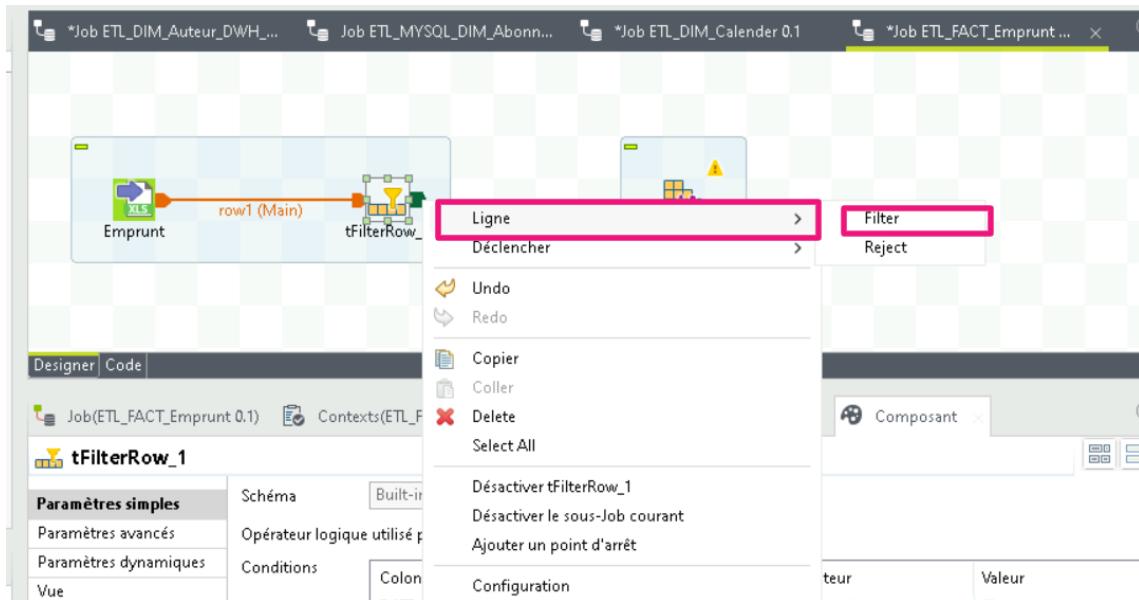
- **Position dans le flux :** après tFilterRow
- **Configuration :**
 - Sort key 1 : NUMERO_AB, ordre croissant
 - Sort key 2 : DATE_EMPRUNT, ordre croissant



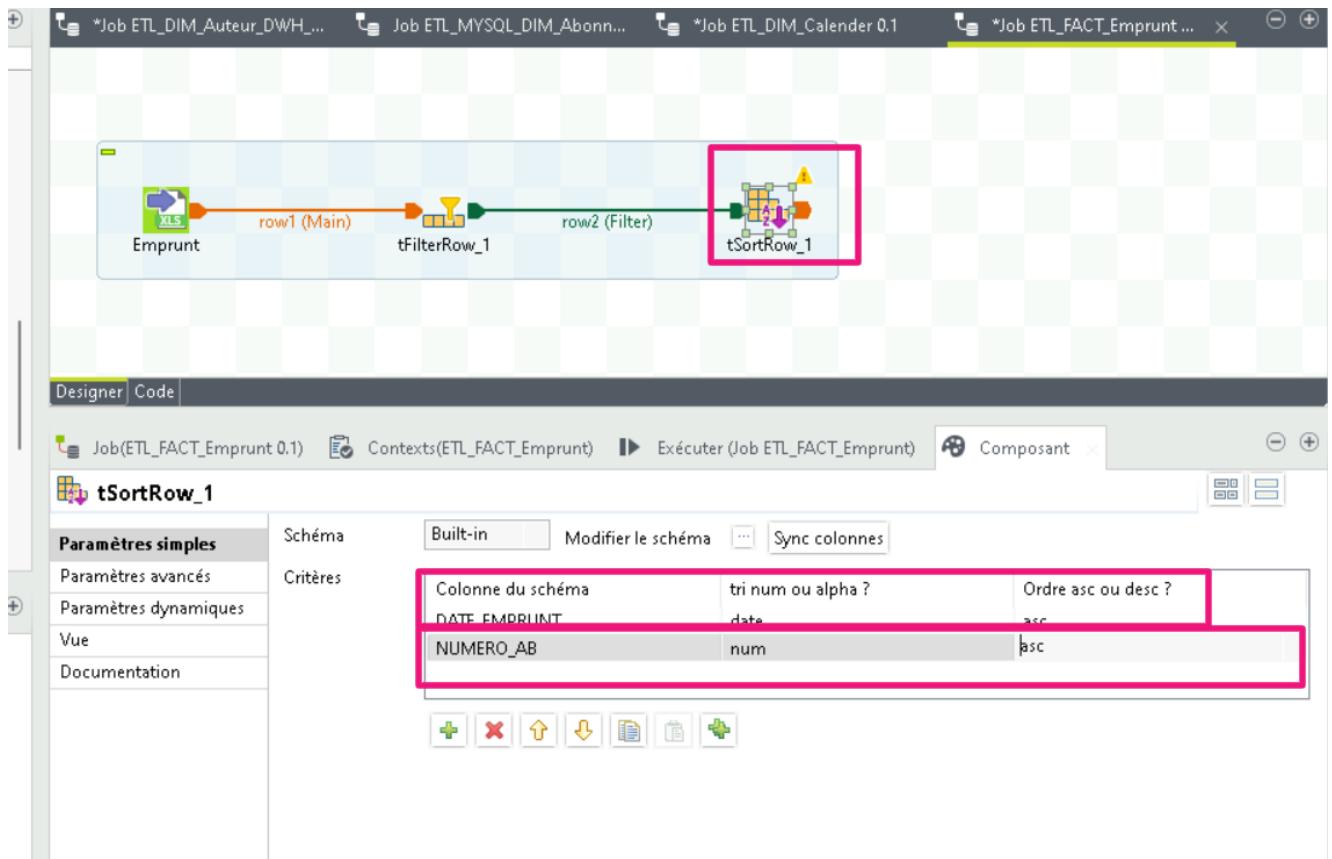
Cliquer sur Installer les modules disponibles



Relier les composants tFiltreRow avec tSortRow



Configurer les conditions de composant tSortRow en ajoutant avec « + » les colonnes et les modes de tri

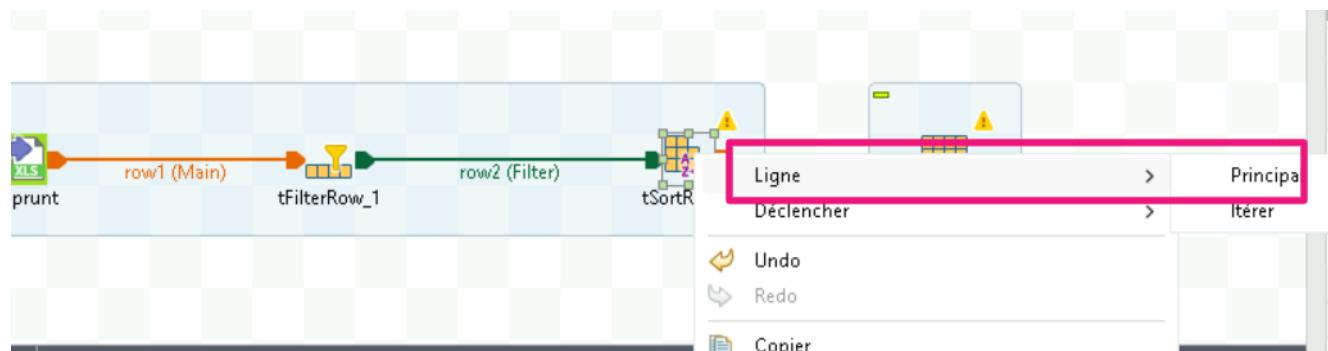


3. tAggregateRow — Agrégation des emprunts

But : Calculer le nombre d'emprunts identiques (même abonné, livre, date).

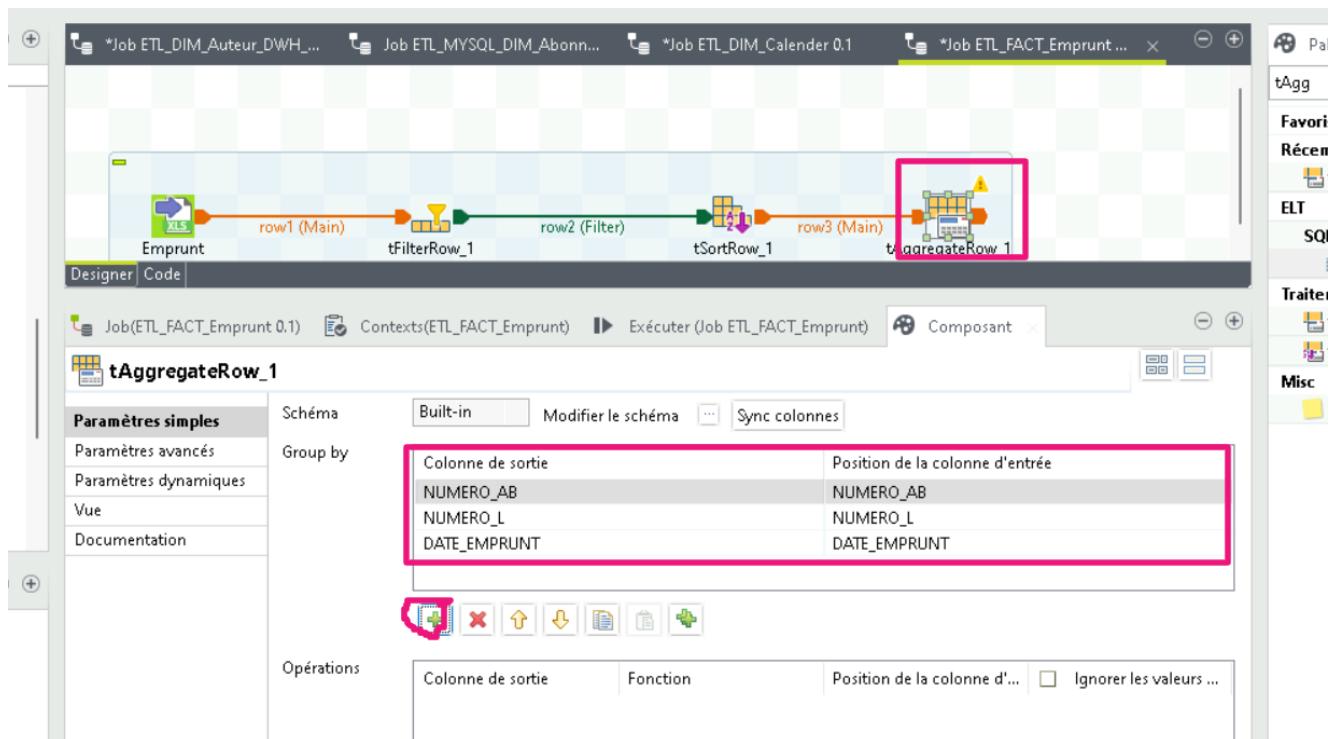


- Position** : après tSortRow (relier les composants)

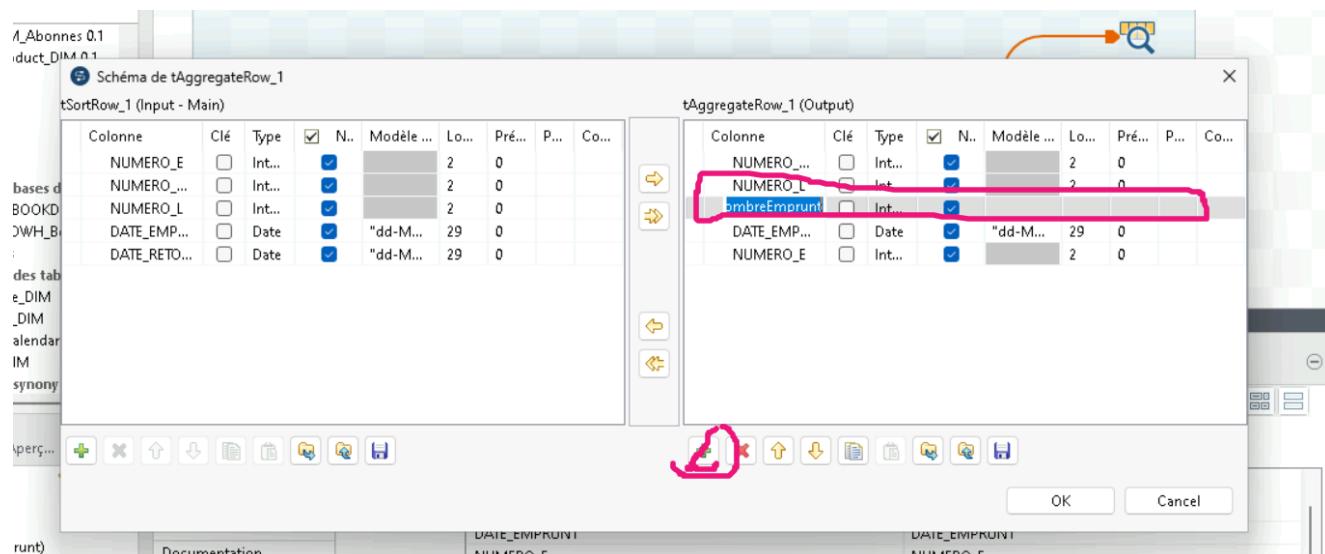


- Group By** : Ajouter les colonnes suivantes dans la conditions group by

- NUMERO_AB
- NUMERO_L
- DATE_EMPRUNT



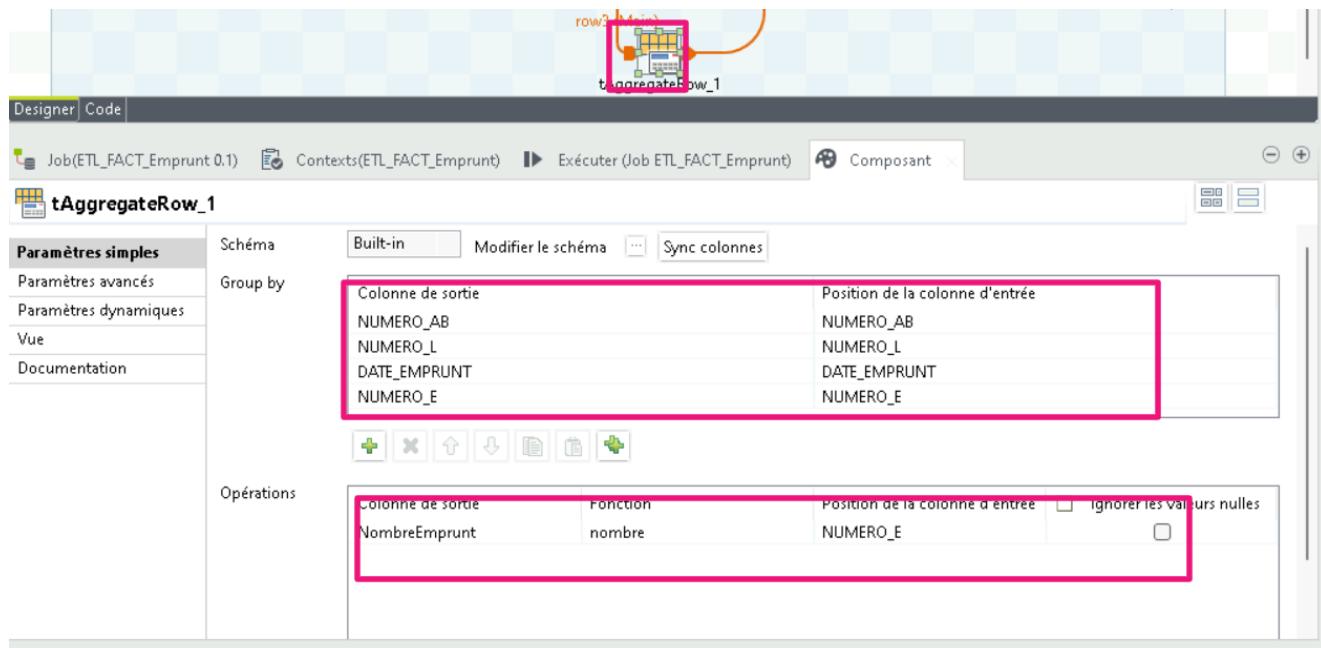
- Modifier le Schéma de Output Agrégation



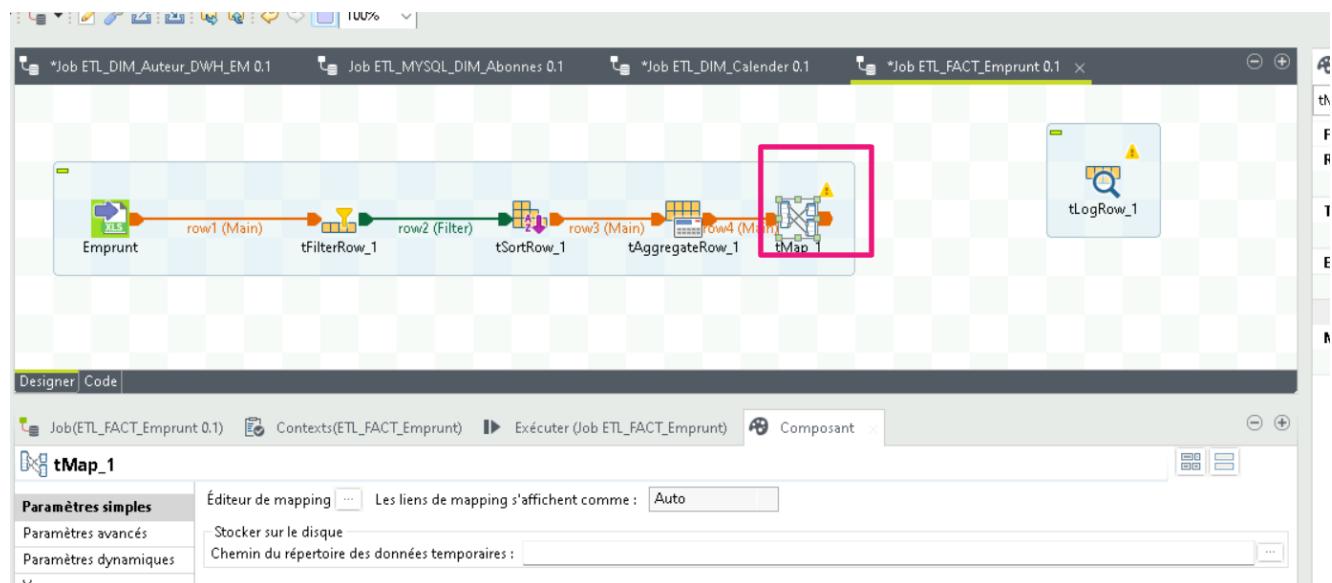
- Operations : Configurer l'opération d'agrégation

- Nom : NombreEmprunt
- Fonction : nombre

- Champ : NUMERO_E (ou toute colonne non nulle)



5. tMap — Génération des clés calendrier

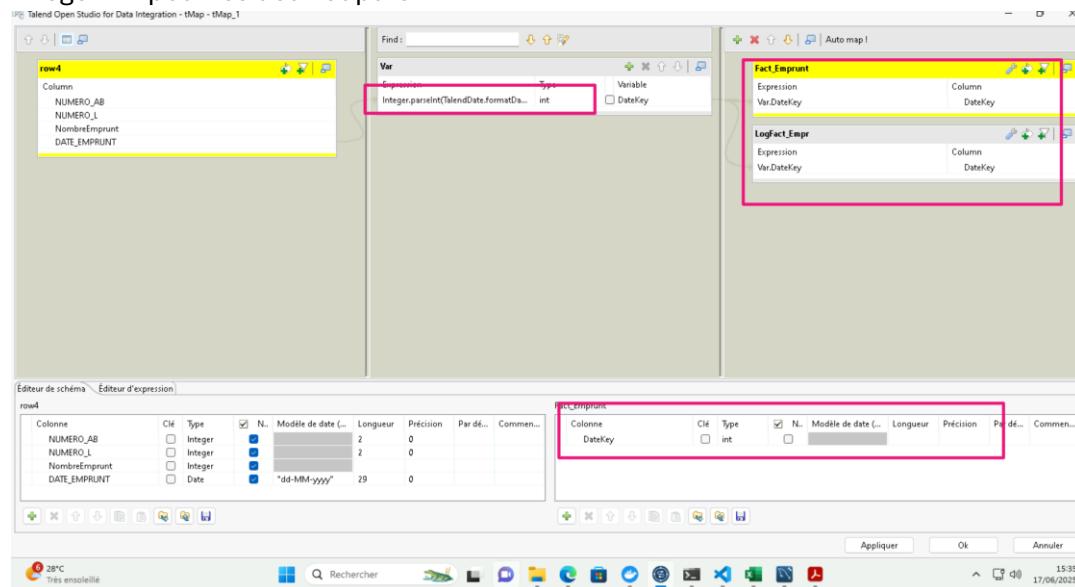


But : Générer les clés DateEmpruntKey et DateRetourKey au format yyyyMMdd

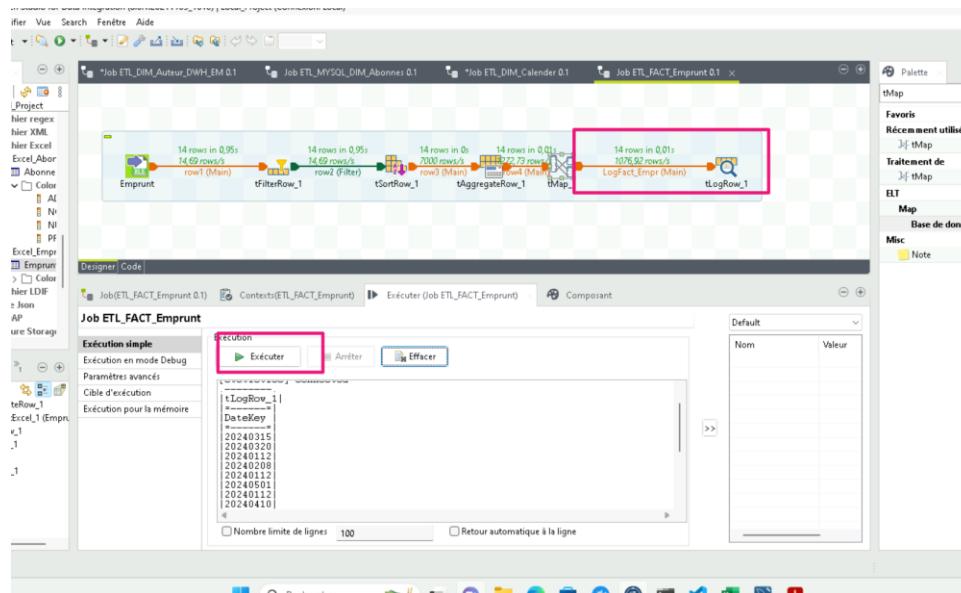
- **Expressions :**

- DateEmpruntKey = Integer.parseInt(TalendDate.formatDate("yyyyMMdd", row1.DATE_EMPRUNT))
- DateRetourKey = Integer.parseInt(TalendDate.formatDate("yyyyMMdd", row1.DATE_RETOUR))

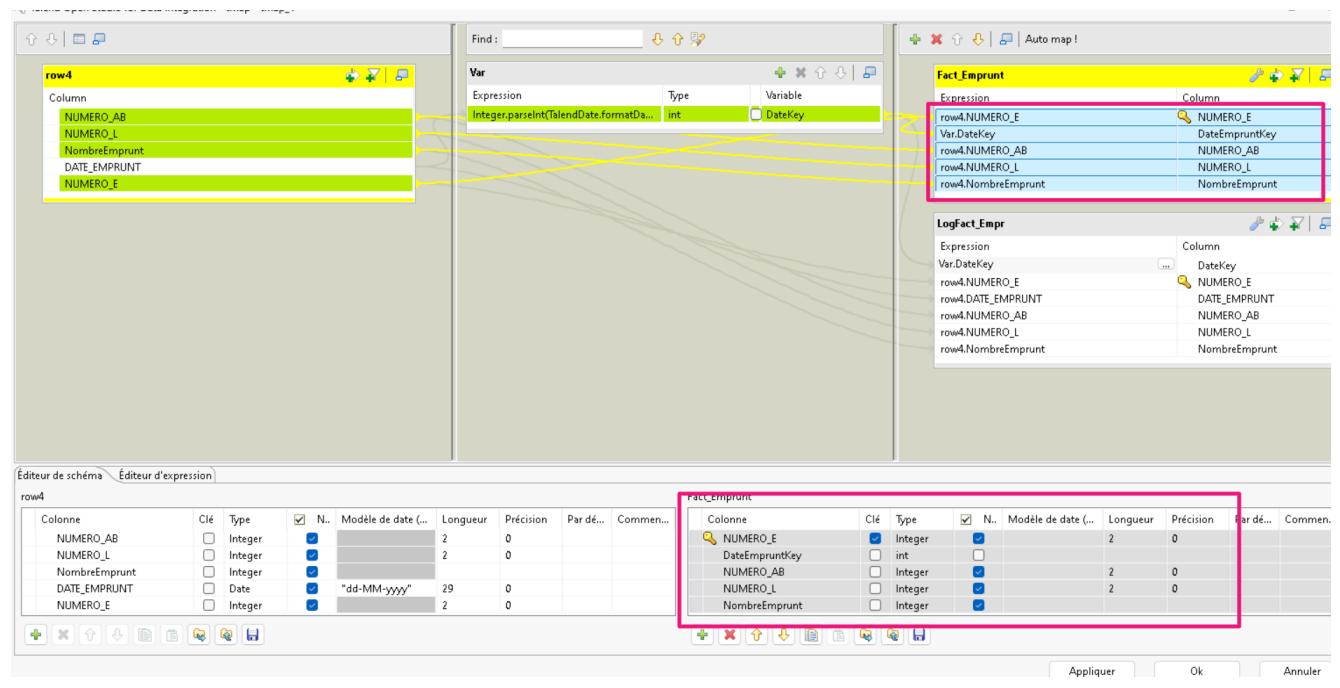
Configurer TMap avec en ajoutant les outputs nécessaires et les mappages des colonnes avec les types Integer/Int pour les deux ouputs



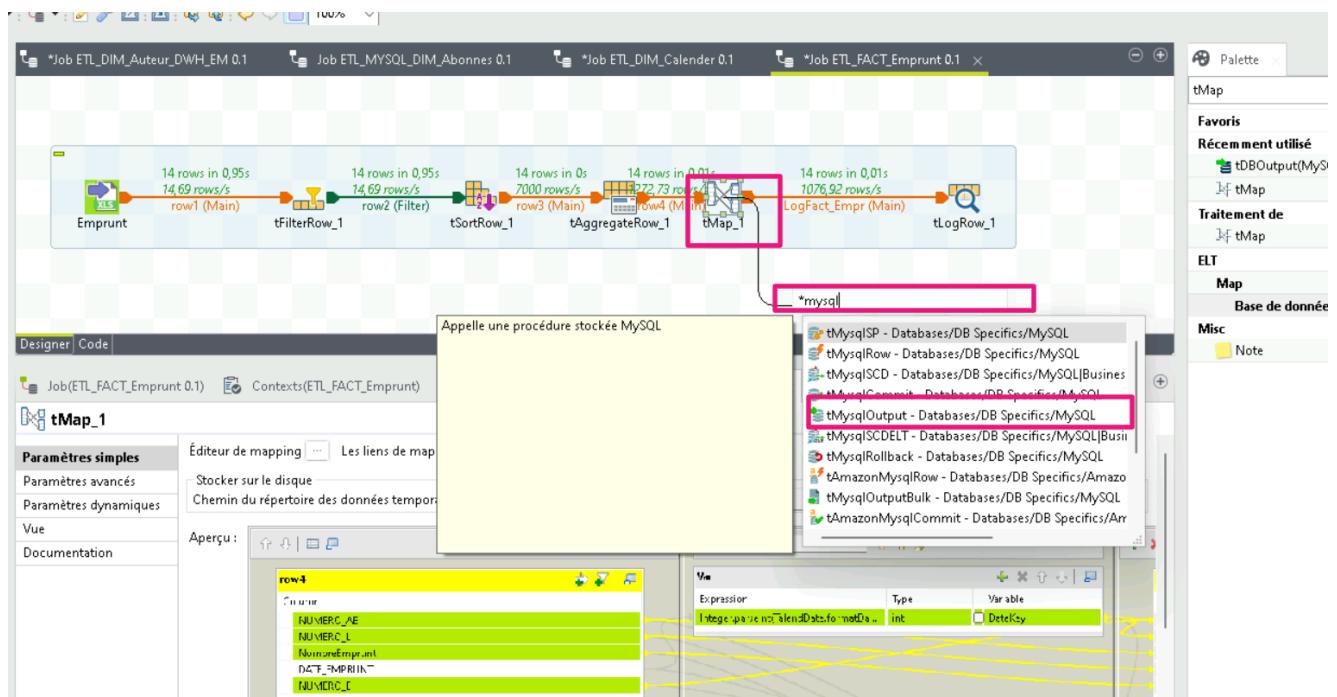
Ajouter un tLogRow pour vérifier la conversion des dates en clés entiers



Configurer les Tmap respectant les champs suivantes



6. tOutputMySQL — Chargement vers Emprunt_FACT



Champs insérés :

Colonne cible Source (via tMap + agrégation)

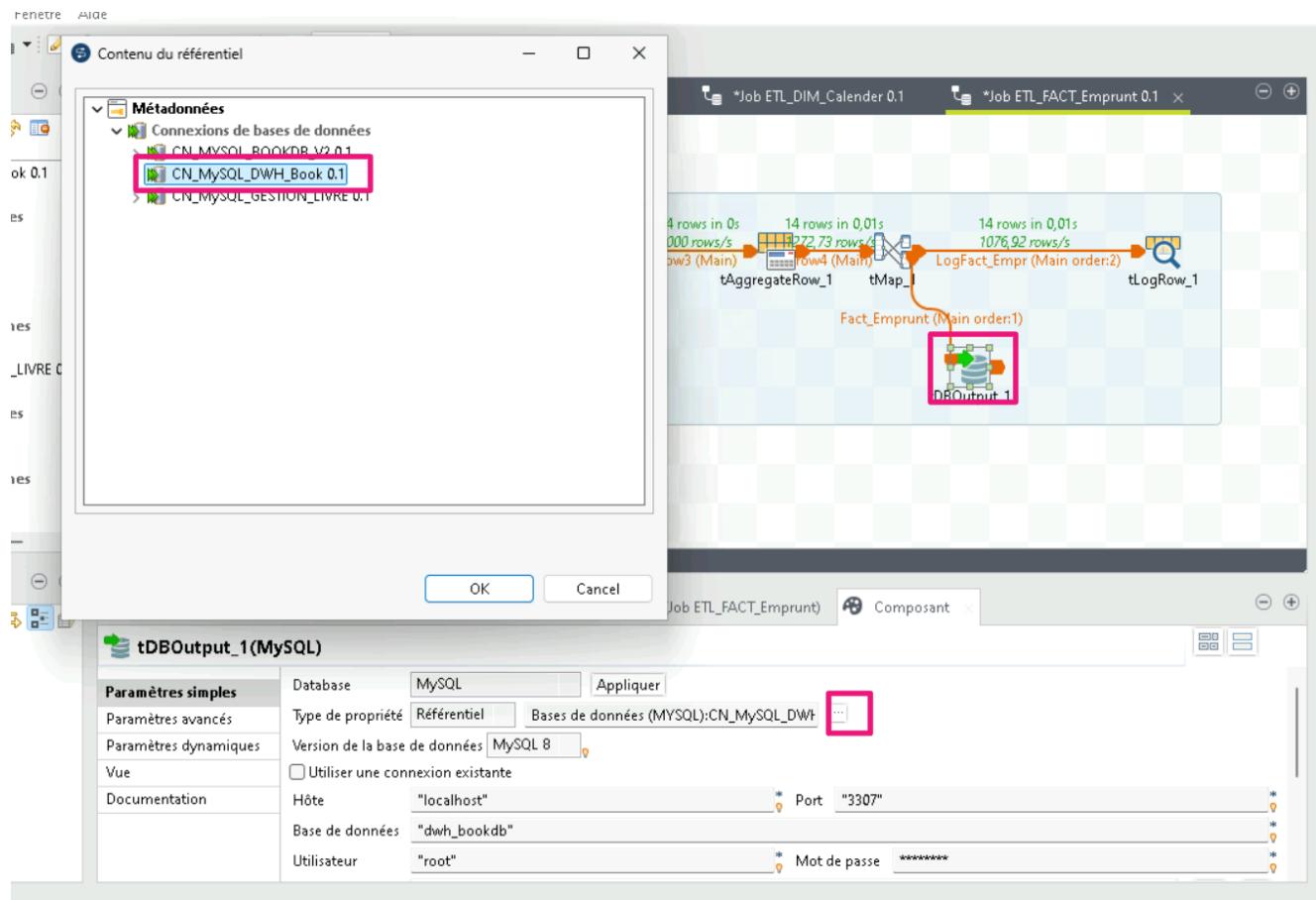
NUMERO_AB row1.NUMERO_AB

NUMERO_L row1.NUMERO_L

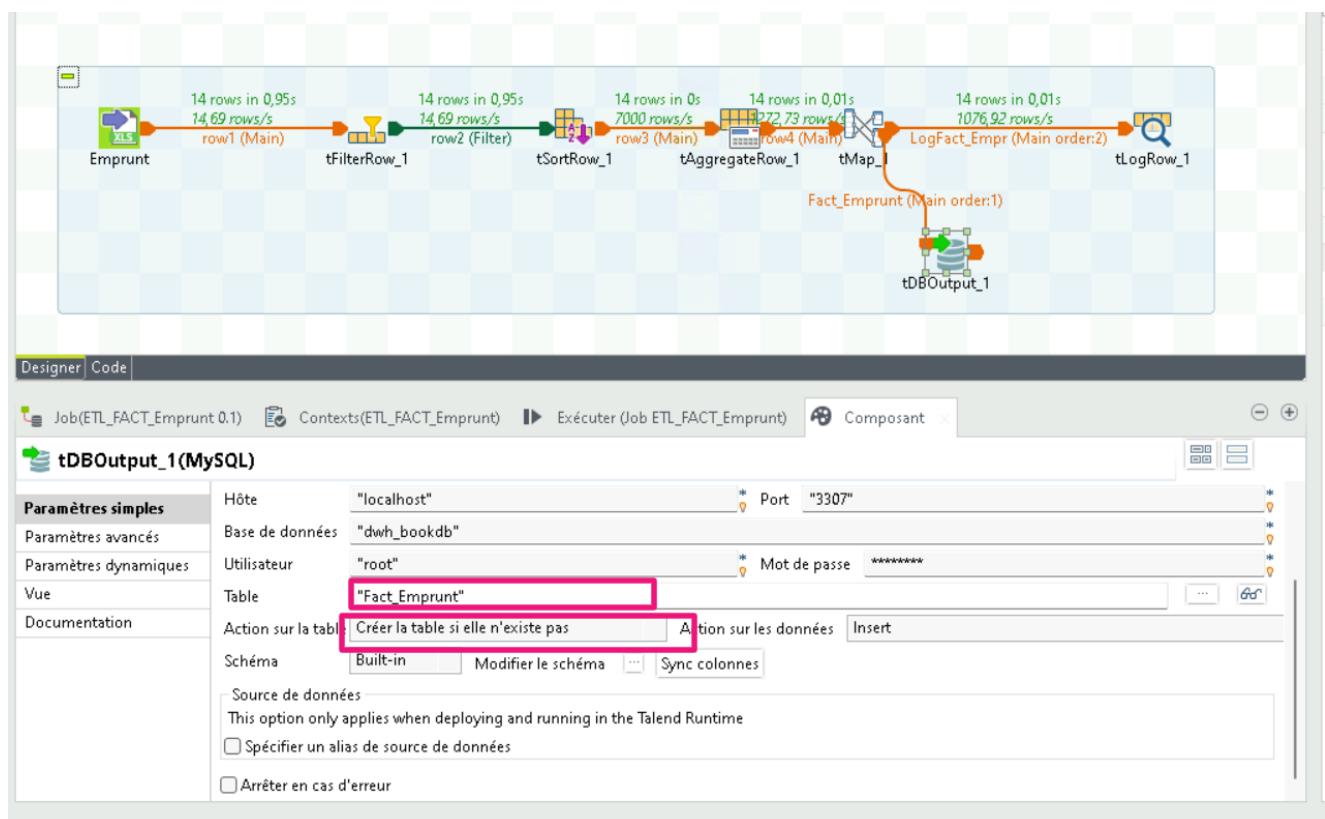
DateEmpruntKey champ calculé

NombreEmprunt résultat de tAggregateRow

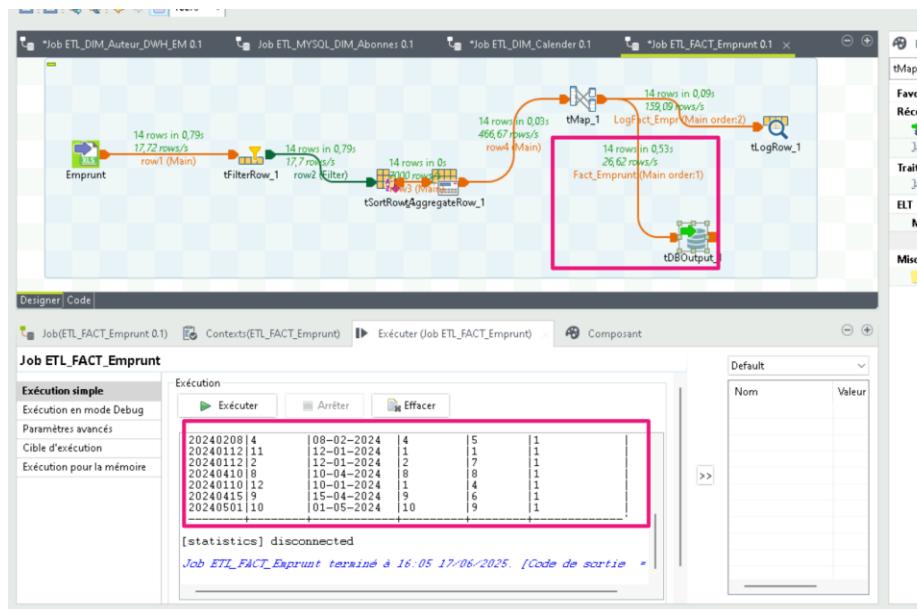
Configurer le composant **tOutputMSQL** en choisissant la connexion du référenciel



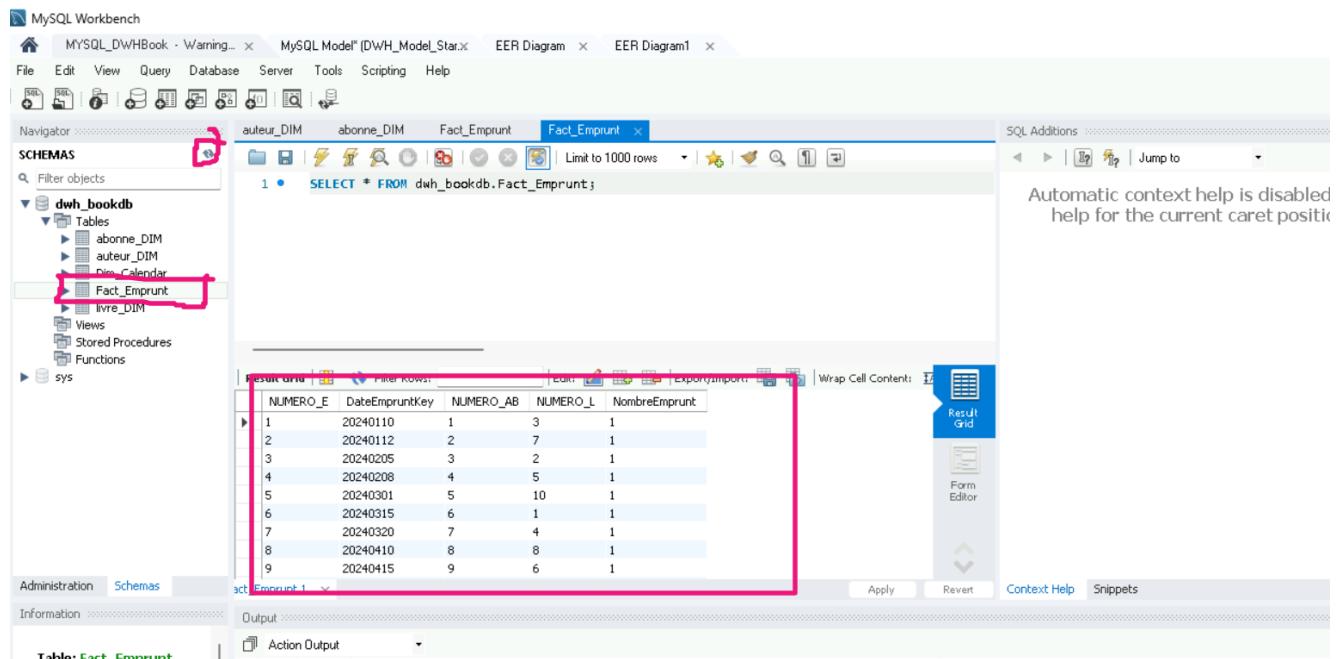
Configurer maintenant la connexion en spécifiant le nom de tables et l'action par défaut et valider



- Exécuter le job et vérifier



- Vérifier avec Workbench



Notifications : Étapes pour intégrer tSendMail dans un Job Talend

1. Activer la validation en deux étapes sur votre compte Gmail

- Rendez-vous sur <https://myaccount.google.com/security>
- Activez "Validation en deux étapes"
 - Obligatoire pour générer un mot de passe d'application.

2. Générer un mot de passe d'application

- Allez sur <https://myaccount.google.com/apppasswords>
- Choisissez :
 - Application : "Mail"
 - Appareil : "Autre" → donnez un nom (par ex. "TalendETL")
- Cliquez sur Générer
- Copiez le mot de passe généré (ex : abcx yzwt qhkm sxyz)

← Mots de passe des applications

Les mots de passe d'application vous permettent de vous connecter à votre compte Google sur les appareils compatibles avec les services Google.

Les mots de passe d'application sont des mots de passe spécifiques aux services à jour qui vous permettent de créer un mot de passe pour chaque application à besoin pour établir une connexion.

[En savoir plus](#)

Vos mots de passe

TalendETL

Pour créer un mot de passe

Nom de l'application

Mot de passe d'application généré

Mot de passe d'application pour votre appareil



Comment l'utiliser ?

Accédez aux paramètres de votre compte Google dans l'application ou l'appareil que vous essayez de configurer. Remplacez le mot de passe par celui de 16 caractères indiqué ci-dessus.

Tout comme votre mot de passe classique, ce mot de passe spécifique à une application permet d'accorder un accès complet à votre compte Google. Étant donné que vous n'avez pas besoin de le mémoriser, ne le notez nulle part ni ne le partagez avec personne.

Terminé

Créer

 3. Coller ce mot de passe dans Talend

Dans le composant tSendMail :

- **User :** votreadresse@gmail.com
- **Password :** le mot de passe d'application collé tel quel (sans espaces en trop)

Créer une application dans Gmail

[Google Compte](#)[← Mots de passe des applications](#)

Les mots de passe d'application vous permettent de vous connecter à votre compte Google sur des applis et des services plus anciens, non compatibles avec les normes de sécurité les plus récentes.

Les mots de passe d'application sont moins sécurisés que les applis et services à jour qui utilisent les normes de sécurité les plus récentes. Avant de créer un mot de passe d'application, vous devez vérifier si votre appli en a besoin pour établir la connexion.

[En savoir plus](#)

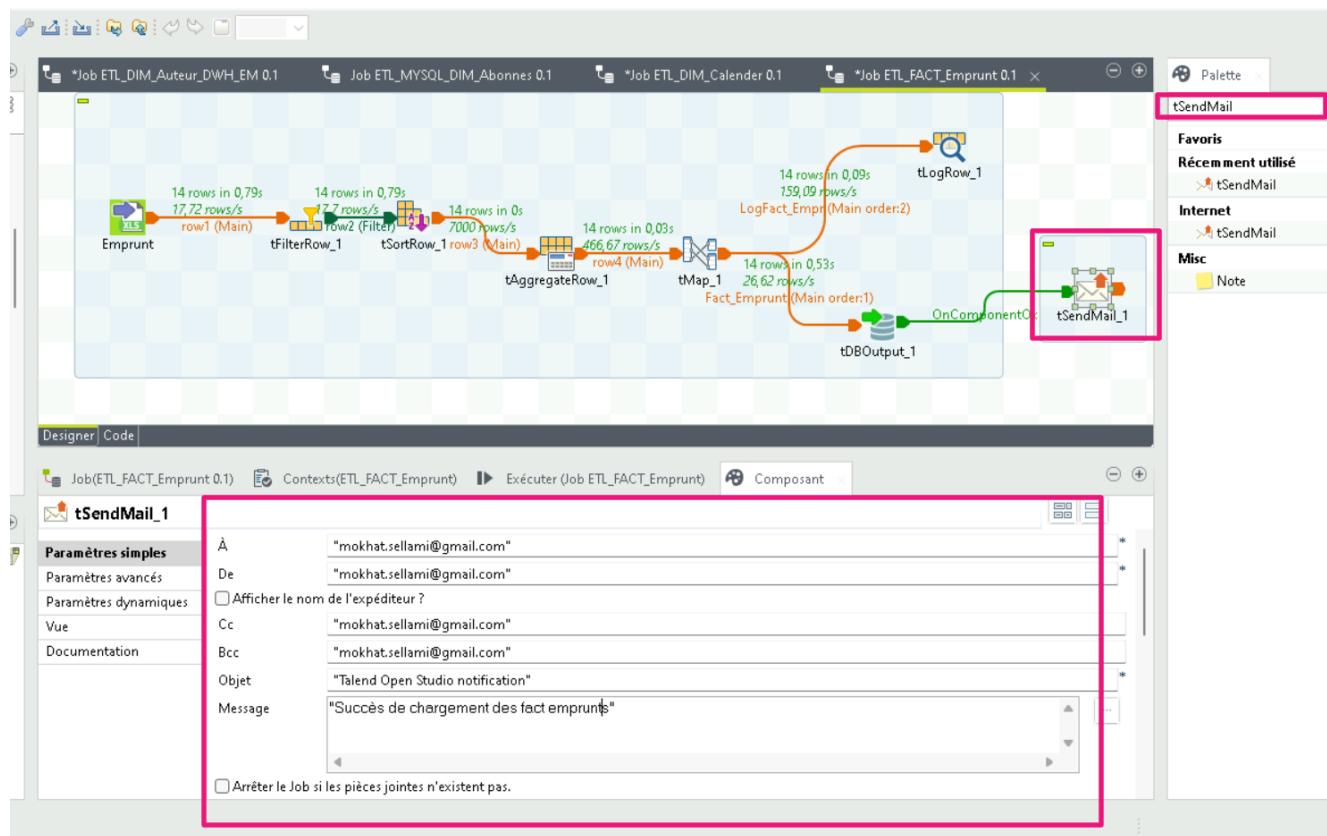
The screenshot shows a user interface for creating a new application password. At the top, a message says "Vous n'avez aucun mot de passe d'application." Below it, a note reads "Pour créer un mot de passe spécifique à une appli, indiquez son nom ci-dessous." A text input field is highlighted with a red border and contains the text "Nom de l'appli" followed by "TalendETL". At the bottom right of the form is a blue "Créer" button.

1. Ajouter tSendMail

Depuis la **Palette**, glissez le composant tSendMail dans l'espace de travail.

2. Configurer tSendMail (dans l'onglet **Basic settings**)

Remplissez les champs suivants :



◆ 1. Ajouter le composant

- Faites glisser **tSendMail** depuis la **Palette** vers l'espace de travail.
- Reliez-le avec un **lien de type OnSubjobOk** depuis le dernier composant (ex. **tOutputMySQL_1** ou **tLogRow_1**).

◆ 2. Paramétrer la connexion SMTP

Dans l'onglet **Component** de **tSendMail**, remplissez :

Champ	Valeur
To	votre@email.com (destinataire)
From	votreadresse@gmail.com (expéditeur)

Champ	Valeur
Subject	Notification ETL_FACT_Emprunt – Succès
Message	"Le job ETL_FACT_Emprunt s'est terminé avec succès à " TalendDate.getCurrentDate()
SMTP host	smtp.gmail.com
Port	587 (<i>SMTP avec TLS</i>)
Use authentication	<input checked="" type="checkbox"/> coché
User	votre adresse Gmail (ex: votreadresse@gmail.com)
Password	 un mot de passe d'application (pas celui de votre Gmail direct)
Use TLS	<input checked="" type="checkbox"/> coché

