

SCIENCE ET INGENIERIE DE DONNEE



DATA WAREHOUSE

Rapport TP 3 : Création et Alimentation du Magasin de Données

Participants

1- TALL Ousmane 22409896

2- OULARE Sékou 22212634

Introduction

Dans le cadre de la mise en place d'un système décisionnel performant, le TP3 se concentre sur la création d'un **magasin de données** (Data Mart). Ce composant clé est une structure optimisée pour l'analyse multidimensionnelle des données, permettant d'extraire des informations utiles pour la prise de décision. Après avoir établi un entrepôt de données dans les TP précédents, ce TP vise à construire un magasin de données qui repose sur des tables dimensionnelles et une table de faits.

L'objectif principal de ce TP est de :

Créer les différentes tables du magasin de données à partir de l'entrepôt déjà alimenté (DM08_MAGASINS, DM08_PRODUITS, DM08_TEMPS et DM08_VENTES).

Alimenter ces tables en transformant et en adaptant les données issues de l'entrepôt via **Talend Open Studio**.

Mettre en œuvre des techniques spécifiques telles que l'agrégation, les jointures et la gestion des dates pour enrichir et structurer les données dans un format prêt pour l'analyse.

Ce rapport détaille les étapes suivies pour la réalisation de ces tâches, les résultats obtenus, ainsi que les défis rencontrés et les solutions apportées. Il illustre également la manière dont ce magasin de données s'intègre dans une chaîne de traitement plus large pour répondre aux besoins d'analyse décisionnelle.

Objectifs du Travail à Réaliser

Créer le **magasin de données** dans la base de données compte_dm08.

Récupérer les **métadonnées** des tables nouvellement créées via Talend en établissant une connexion avec compte_dm08.

Structurer et alimenter les tables avec les schémas correspondants.

1. **DM08_D_MAGASINS** (dimension des magasins)
2. **DM08_D_PRODUITS** (dimension des produits)
3. **DM08_D_TEMPS** (dimension du temps)
4. **DM08_F_VENTES** (table de fait des ventes)

Création des quatre (4) Tables :

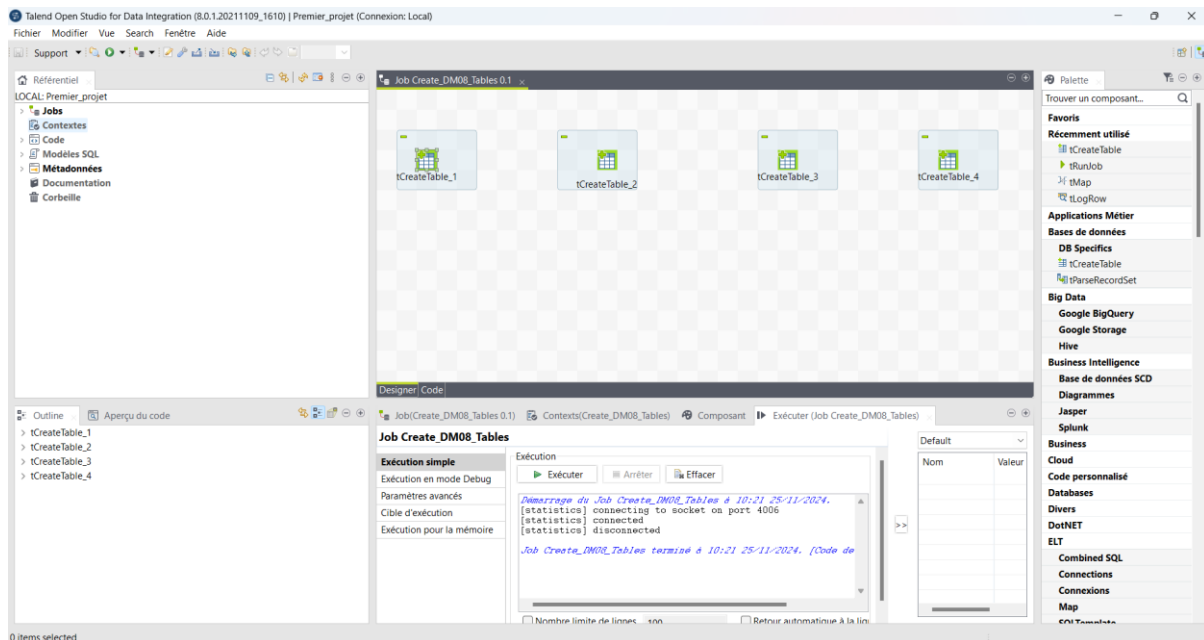
Nous avons cette fois-ci, décidé de créer les 4 tables dans un seul Job qui est le suivant :

Pour optimiser la gestion et le déploiement des structures du magasin de données, les quatre tables nécessaires ont été créées dans un seul **Job**. Cette approche centralisée permet de rationaliser le processus de création, d'assurer la cohérence des étapes et de minimiser les erreurs potentielles.

Le Job contient les composants **tCreateTable**, configurés pour chaque table cible :

- **DM08_D_MAGASINS** : Table dimensionnelle pour stocker les informations sur les magasins.
- **DM08_D_PRODUITS** : Table dimensionnelle pour les produits.
- **DM08_D_TEMPS** : Table dimensionnelle pour la gestion des dates.
- **DM08_F_VENTES** : Table de faits pour les données relatives aux ventes.

Chaque composant **tCreateTable** a été configuré avec les schémas correspondant à sa table respective, incluant les clés primaires et les types de données. L'exécution du Job assure la création automatique des tables dans la base de données associée, prêtes à recevoir les données transformées et chargées dans les étapes suivantes. Cette organisation garantit un processus fluide et cohérent pour la mise en place du magasin de données.



Après la création des tables, nous avons ajouté une nouvelle connexion à la base de données compte_dm08 puis procédé à la Récupération des métadonnées.

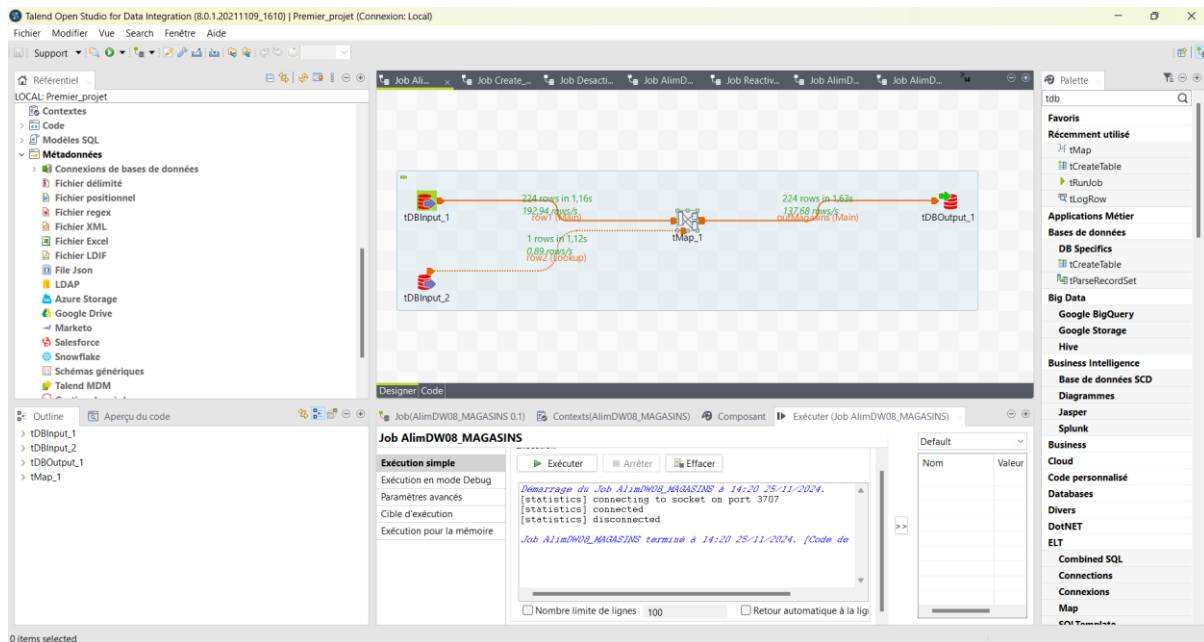
ALIMENTATION DES TABLES :

1 - Alimentation de la Table DM08_D_MAGASINS

L'alimentation de la table **DM08_D_MAGASINS** a été réalisée à l'aide d'un Job Talend dédié. Ce Job utilise plusieurs composants clés pour extraire, transformer et charger les données dans la table cible. Voici les étapes principales de ce processus :

1. Extraction des données :

- a. Les données source sont récupérées via le composant **tDBInput**, configuré pour interroger les tables de production contenant les informations nécessaires sur les magasins.
- b. La requête SQL utilisée permet de sélectionner les champs essentiels tels que le **Code Magasin**, le **Nom du Magasin**, et les informations géographiques associées.

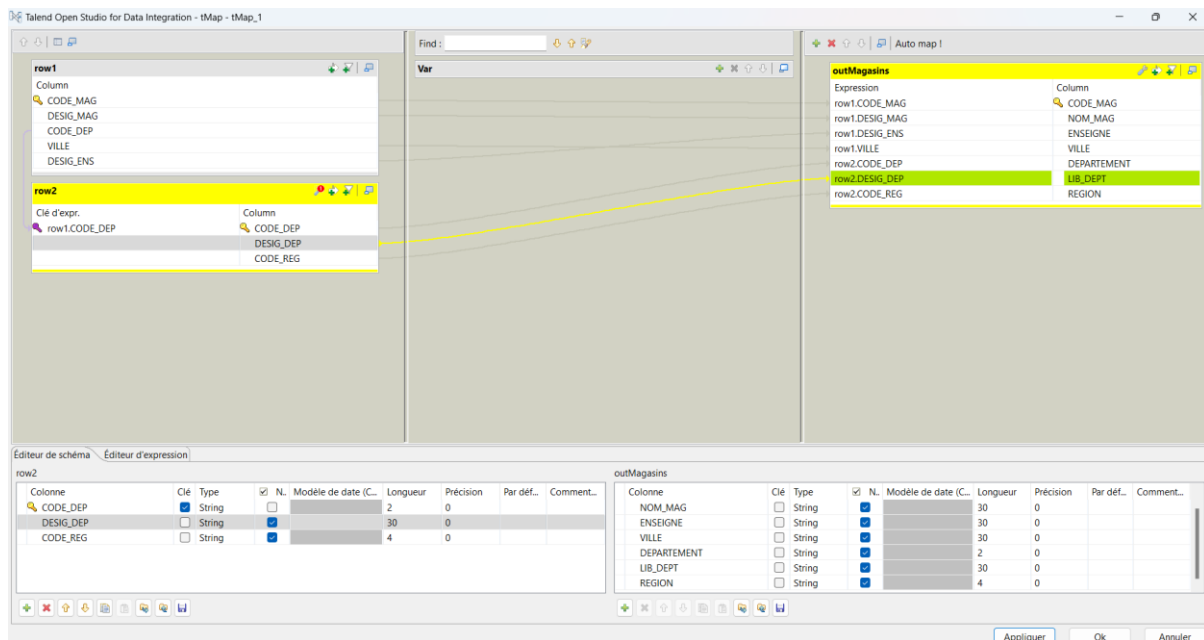


2. Transformation des données :

- Le composant **tMap** a été utilisé pour appliquer les transformations nécessaires, comme le formatage des champs ou la gestion des valeurs par défaut.
- Cette étape garantit que les données sont conformes au schéma attendu de la table **DM08_D_MAGASINS**.

3. Chargement des données :

- Les données transformées sont ensuite chargées dans la table cible via le composant **tDBOutput**.
- Ce composant est configuré pour insérer les enregistrements dans la base de données en respectant les contraintes définies, notamment les clés primaires et les relations avec d'autres tables.
- L'exécution de ce Job a permis de peupler la table **DM08_D_MAGASINS** avec les données nécessaires, prêtes à être utilisées pour des analyses décisionnelles. Ce processus garantit une alimentation fiable et structurée, essentielle pour la cohérence du data warehouse.



2 - Alimentation de la Table DM08_D_PRODUITS

La table **DM08_D_PRODUITS** a été alimentée en suivant un processus structuré pour garantir l'intégrité et la fiabilité des données. Voici les étapes principales de ce processus :

1. Extraction des données :

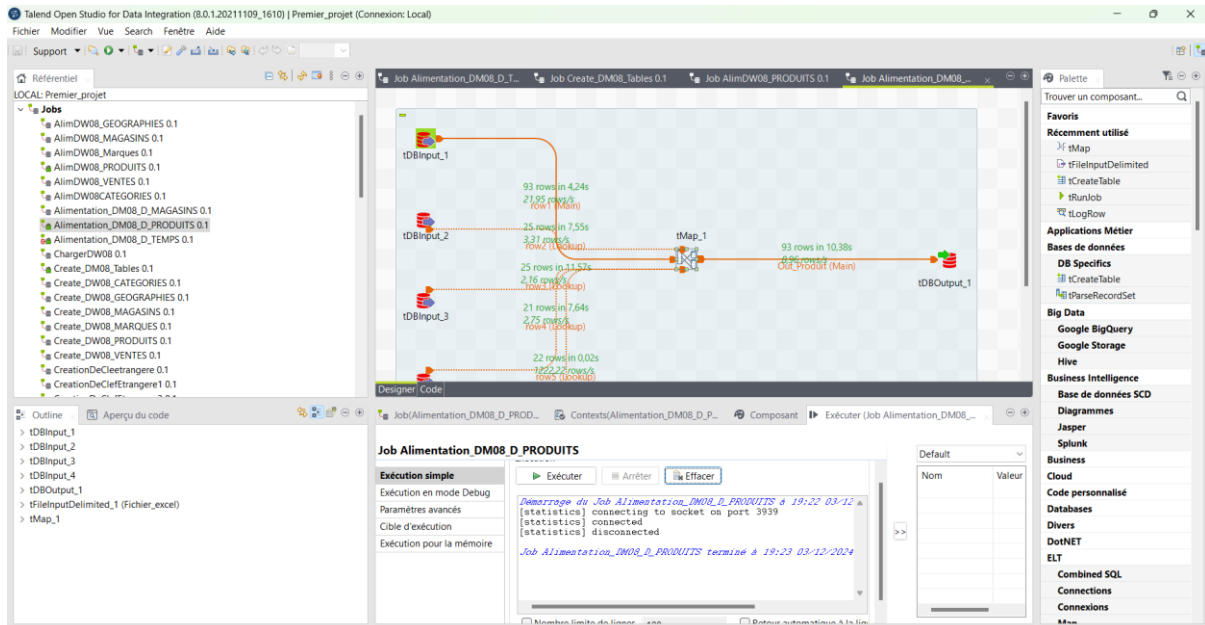
- Les informations relatives aux produits ont été extraites à l'aide du composant **tDBInput**, configuré pour interroger les tables sources de production.
- La requête SQL utilisée a permis de récupérer des données essentielles telles que le **Code Produit**, la **Description Produit**, le **Prix Unitaire**, ainsi que les relations avec les catégories et marques via les clés étrangères **Code Catégorie** et **Code Marque**.

2. Chargement des données :

- Le composant **tDBOutput** a permis d'insérer les données transformées dans la table **DM08_D_PRODUITS**.
- Une attention particulière a été accordée au respect des contraintes de clés étrangères pour garantir la cohérence avec les tables **DM08_D_CATEGORIES** et **DMXX_D_MARQUES**.

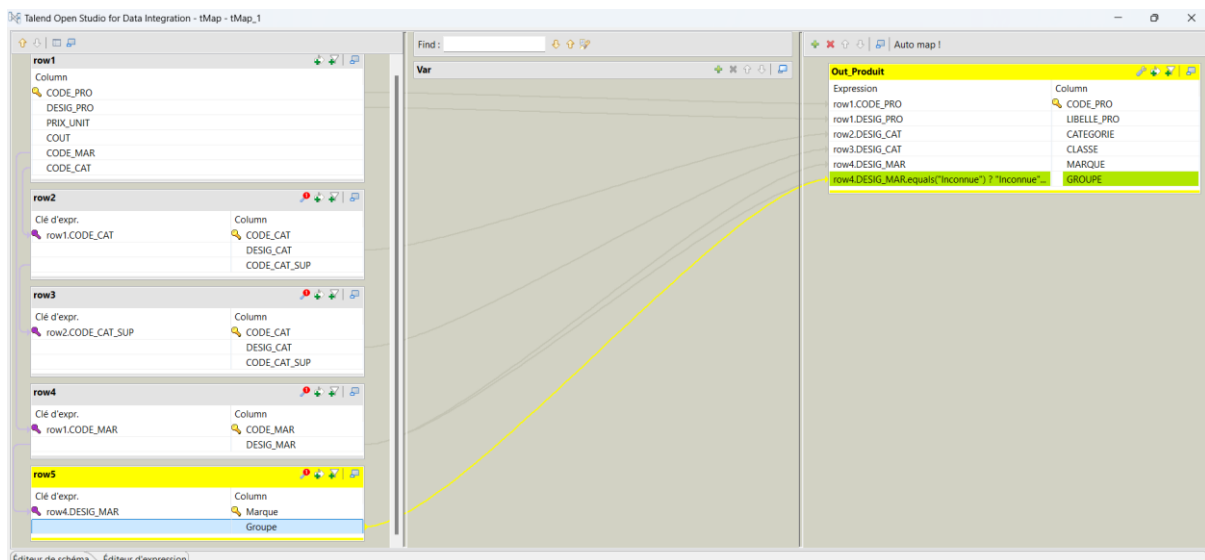
Ce processus a permis de constituer une table produits complète, intégrée et prête à être **exploitable dans le cadre des analyses décisionnelles**. Elle constitue désormais

un élément clé de l'entrepôt de données, assurant une représentation fidèle et cohérente des produits dans le système global.



3. Transformation des données :

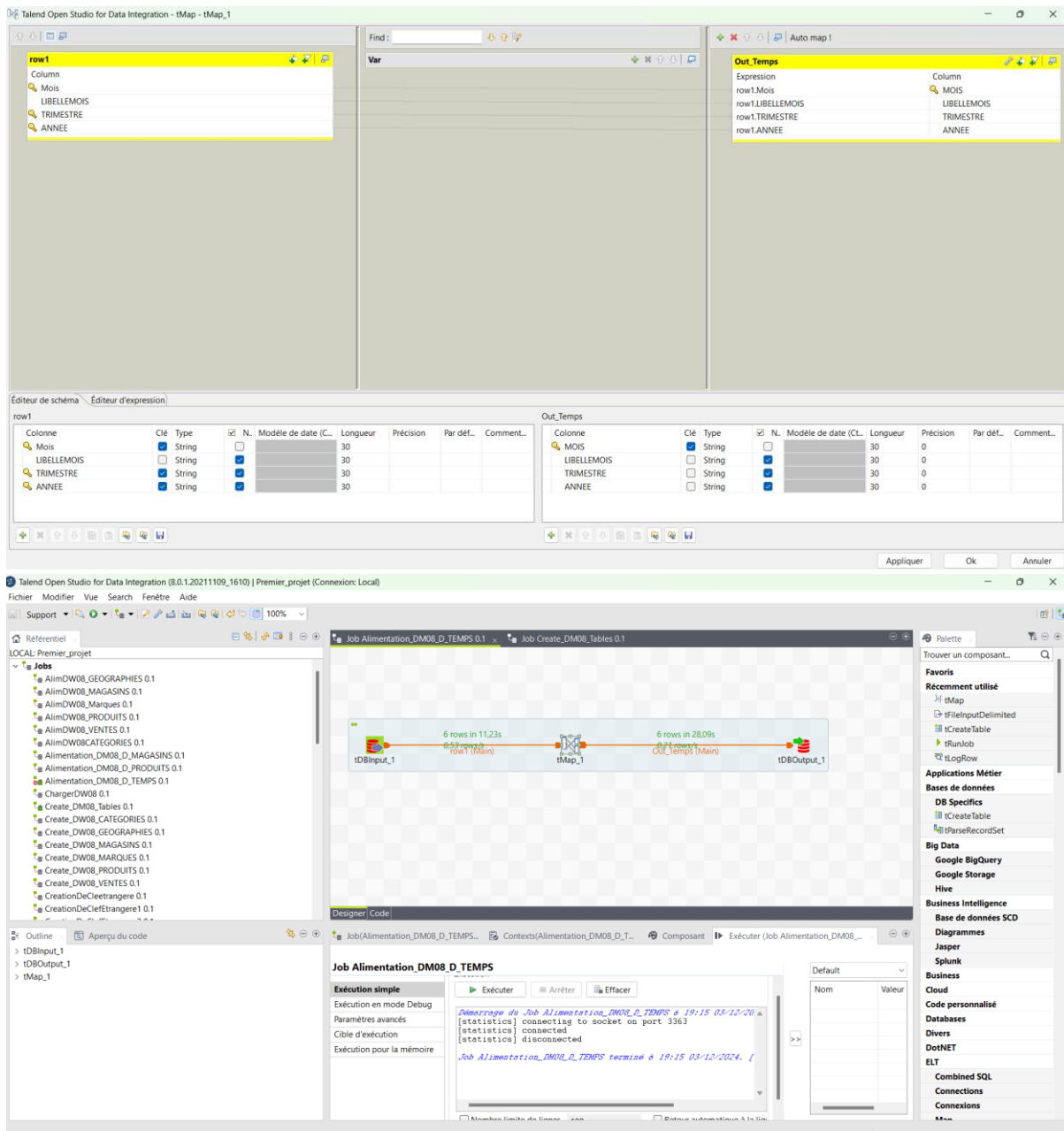
- a. Un composant **tMap** a été employé pour gérer les transformations nécessaires. Cette étape a inclus :
 - i. La correspondance entre les colonnes source et cible.
 - ii. La vérification des valeurs nulles ou incohérentes.
 - iii. La création de relations claires entre les clés étrangères, notamment entre les catégories et les marques.



3 - Alimentation de la Table DM08_D_TEMPS

Pour la gestion de date nous avons utilisé la requête suivante :

```
SELECT DISTINCT  
  
    TO_CHAR(DATE_VT, 'MM-YYYY') AS MOIS,  
  
    TRIM(TO_CHAR(DATE_VT, 'Month', 'NLS_DATE_LANGUAGE=FRENCH')) AS  
LIBELLEMOIS,  
  
    CEIL(TO_NUMBER(TO_CHAR(DATE_VT, 'MM')) / 3) AS TRIMESTRE,  
  
    TO_CHAR(DATE_VT, 'YYYY') AS ANNEE  
  
FROM COMPTE_DW08.DW08_VENTES  
  
WHERE DATE_VT IS NOT NULL ORDER BY ANNEE, MOIS
```

4 - Table DM08_F_VENTES

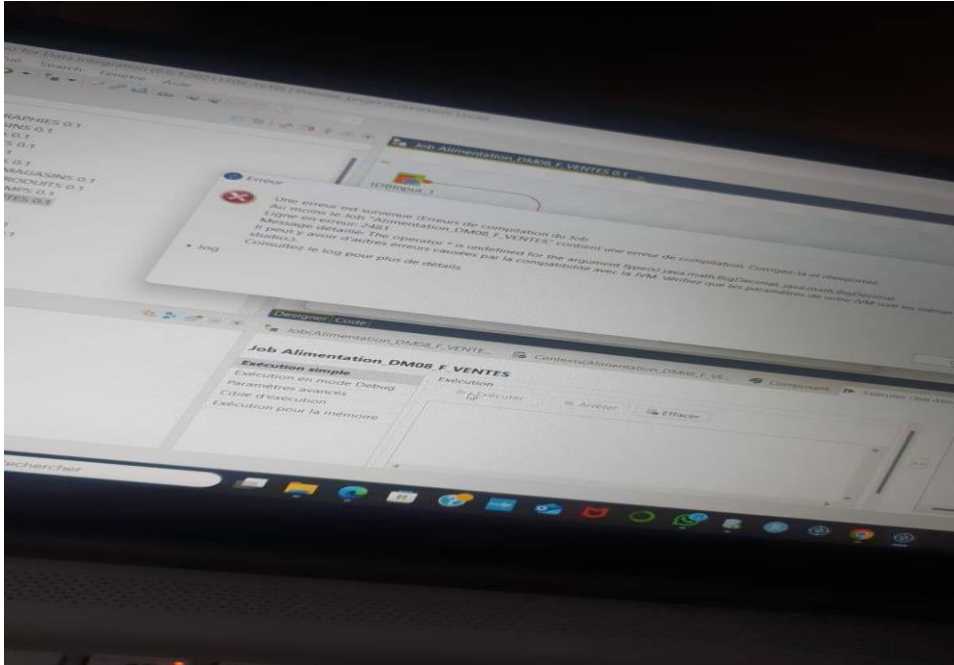
Lors de l'exécution du job d'alimentation de la table **DM08_F_VENTES**, une erreur a été détectée. Le message d'erreur indiquait :

"The operator '*' is undefined for the argument types java.math.BigDecimal, java.math.BigDecimal".

Analyse de l'Erreur

- L'erreur était liée à une incompatibilité de types dans une opération de calcul dans le composant **tMap**.

- Le problème provenait de la tentative de multiplication de valeurs stockées sous le type **BigDecimal** sans conversion explicite.

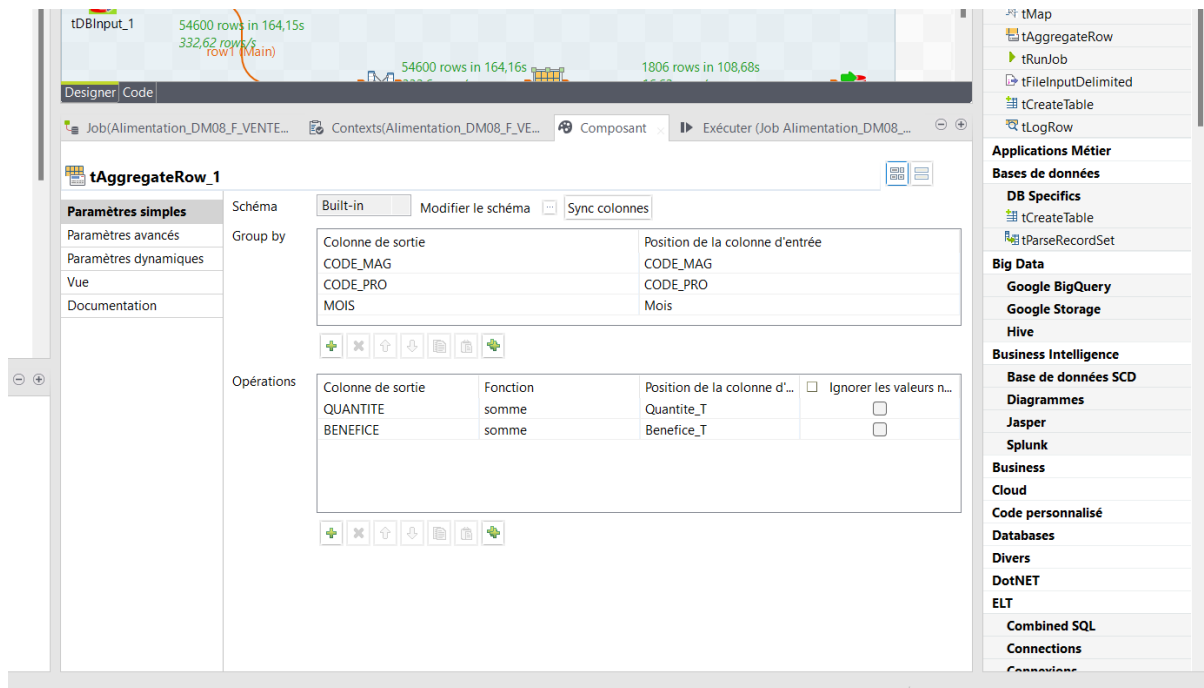
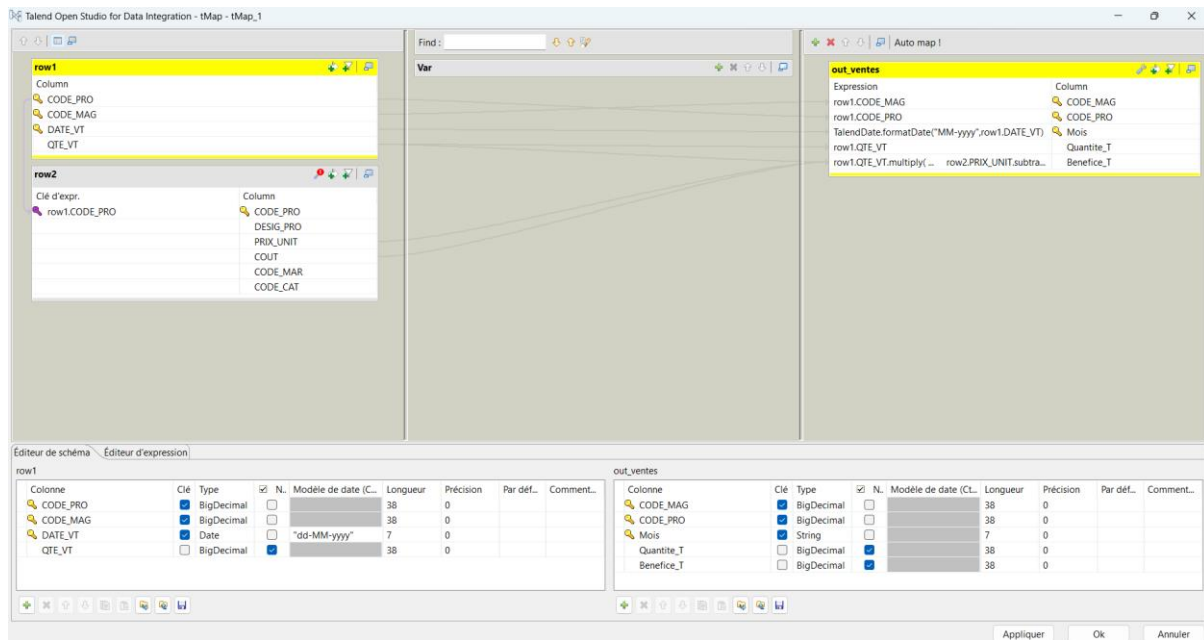


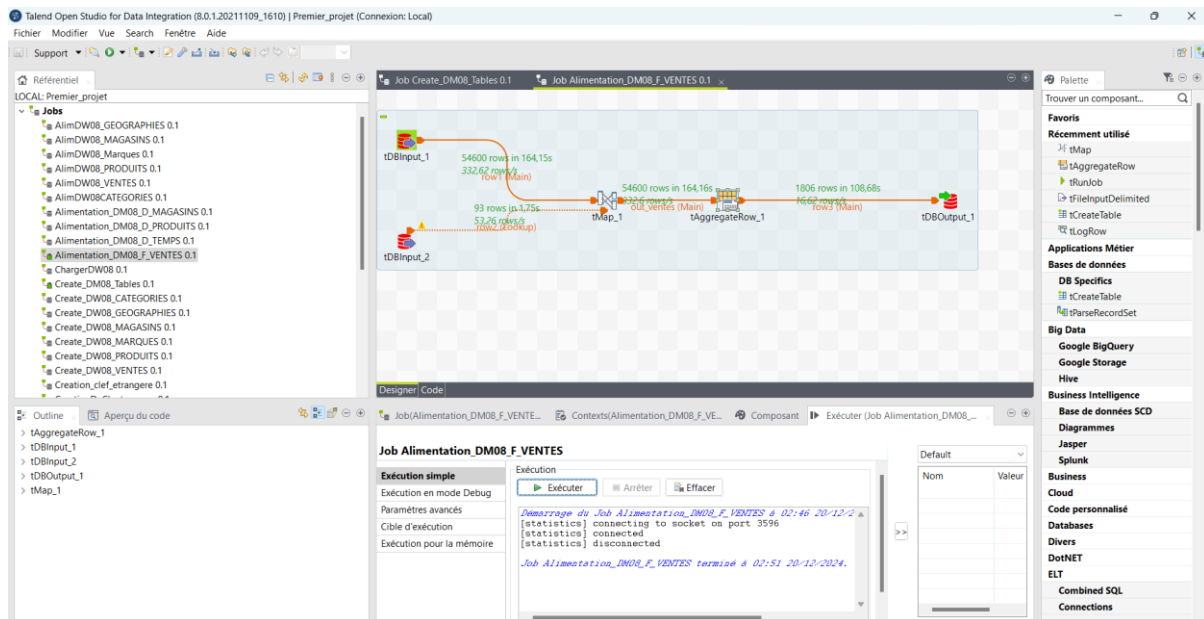
Solution Apportée

Pour résoudre cette erreur :

1. Les colonnes impliquées dans le calcul (PRIX_UNT, QTE_VT, etc.) ont été vérifiées et converties explicitement en types compatibles via la fonction **BigDecimal**.
2. Une expression appropriée a été utilisée dans le composant **tMap** pour effectuer le calcul du bénéfice sans erreurs.

Après avoir effectué ces corrections, le job a été exécuté avec succès, et les données ont été intégrées correctement dans la table **DM08_F_VENTES**.





Conclusion

Ce TP a permis de réaliser avec succès la **création d'un magasin de données** en utilisant les outils SQL et Talend. La connexion et l'importation des métadonnées des tables du magasin ont été validées, mais aussi nous a permis de mettre en œuvre un processus ETL complet dans Talend, illustrant les étapes d'extraction, transformation et chargement