PROJET QUALITÉ DE DONNÉES

Utilisation d'un ETL pour évaluer et améliorer la qualité des données

Réalisé par : Wafa ZARROUKI

Wafa OUNIS

Omar EL MIR

Achref BOUZID

Encadrés par : Mme Zoubida KEDAD

2019-2020

I- Scénario :

Le Crous de Versailles considère une aide financière pour aider les étudiants de l'université Paris Saclay qui ont des revenus faibles .

Notre travail consiste à créer un entrepôt de données pour l'Université Paris Saclay. Cet entrepôt regroupera trois bases de données réparties dans différentes universités accréditées de Paris Saclay (Université d'Evry , Université de Versailles , Université Paris Sud) .

Les bases de données possèdent la même structure mais sous différents formats.

Les bases de données source :

Notre travail comprend 3 sources de données :

- Fichier **CSV** : contient une table Etudiant (NumEtudiant, Nom, Prenom, Nationalité, NumTelephone, Email, Adresse, Revenus) représentant les données des étudiants inscrits à Paris Sud.

- Fichier XLS : contient une table Etudiant (numEtu, Nom, Prenom, Nationalité, Tel, Email, Salaire, codePostal) représentant les données des étudiants inscrits à Versailles.

- Fichier XML : contient une table Etudiant (etuID, Nom, Prenom, Nationalité, Telephone, Adresse, Email, Salaire) représentant les données des étudiants inscrits à Evry.

L'objet cible: un seul fichier représentant une table **ETUDIANT** (IDEtudiant,Nom, Prenom, Nationalité, NumTelephone, Email, Adresse, Revenus)

II- Objectif :

Notre objectif est de regrouper les trois sources dans un entrepôt de données cible en utilisant l'ETL Talend et en considérant quatre facteurs de qualités :

- **Conformité à un format, une codification**
- **Hétérogénéité des échelles, de la granularité**
- **Complétude des données**
- **Détection et élimination de doublons**.

III- Travail réalisé :

Le travail sera divisé en deux parties, détection des problèmes et amélioration de la qualité des données pour chaque facteur de qualité.

Certains cas exigent que les sources de données soient regroupés pour pouvoir faire la détection. Pour cela nous avons fusionné les sources en utilisant les composants tUnit et tMap (pour faire le mapping entre les attributs sources et cibles).

Détection des doublons :

Objet : les attributs : Nom , Prénom , Nationalité , Adresse

Problème : Le but est de déterminer que deux objets représentent le même objet . Un étudiant est inscrit dans deux universités différentes.

Évaluation : Deux tuples différents représentent le même étudiant si ils présentent la mème valeur dans les attributs : Nom , Prénom , Nationalité , Adresse.

Méthode : Sous Talend on utilise le composant tUniqRow qui compare les entrées et met les doublons du flux d'entrée dans un fichier de sortie séparé.

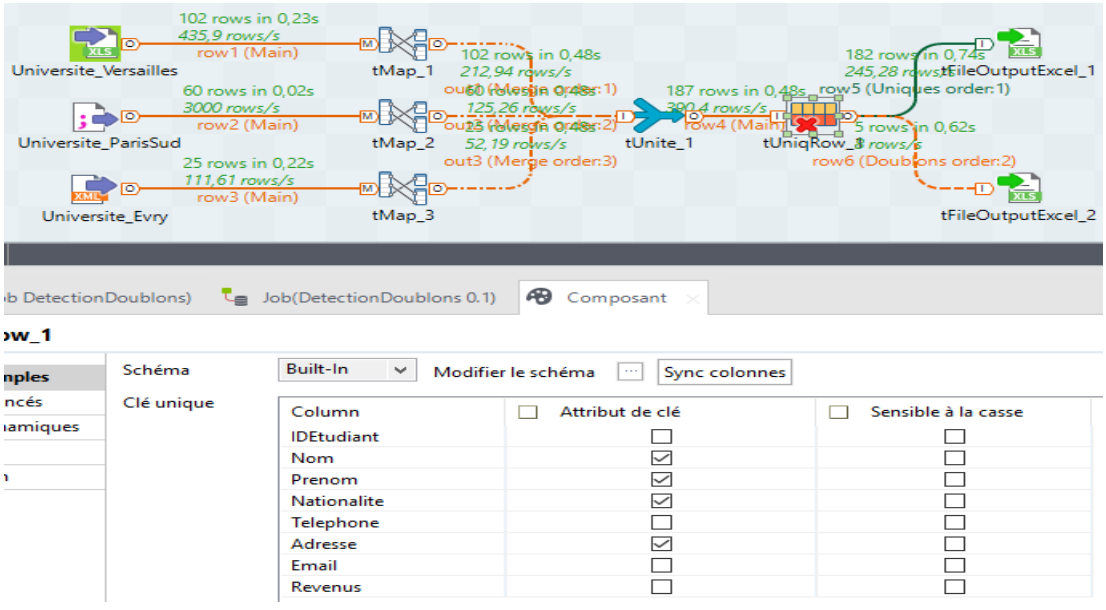

Figure : Workflow de detection du problème de doublons

Résultat : 5 tuples rejetés (information incomplète)

Amélioration : A la présence de deux tuples présentant le même IDEtudiant (Nom, Prénom, Nationalité, Adresse sont identiques), on garde que le tuple qui a le taux de revenus le plus élevé, on supprime l'autre tuple tout en conservant son IDEtudiant et l'ajouter a notre table cible a l'étudiant correspondant.

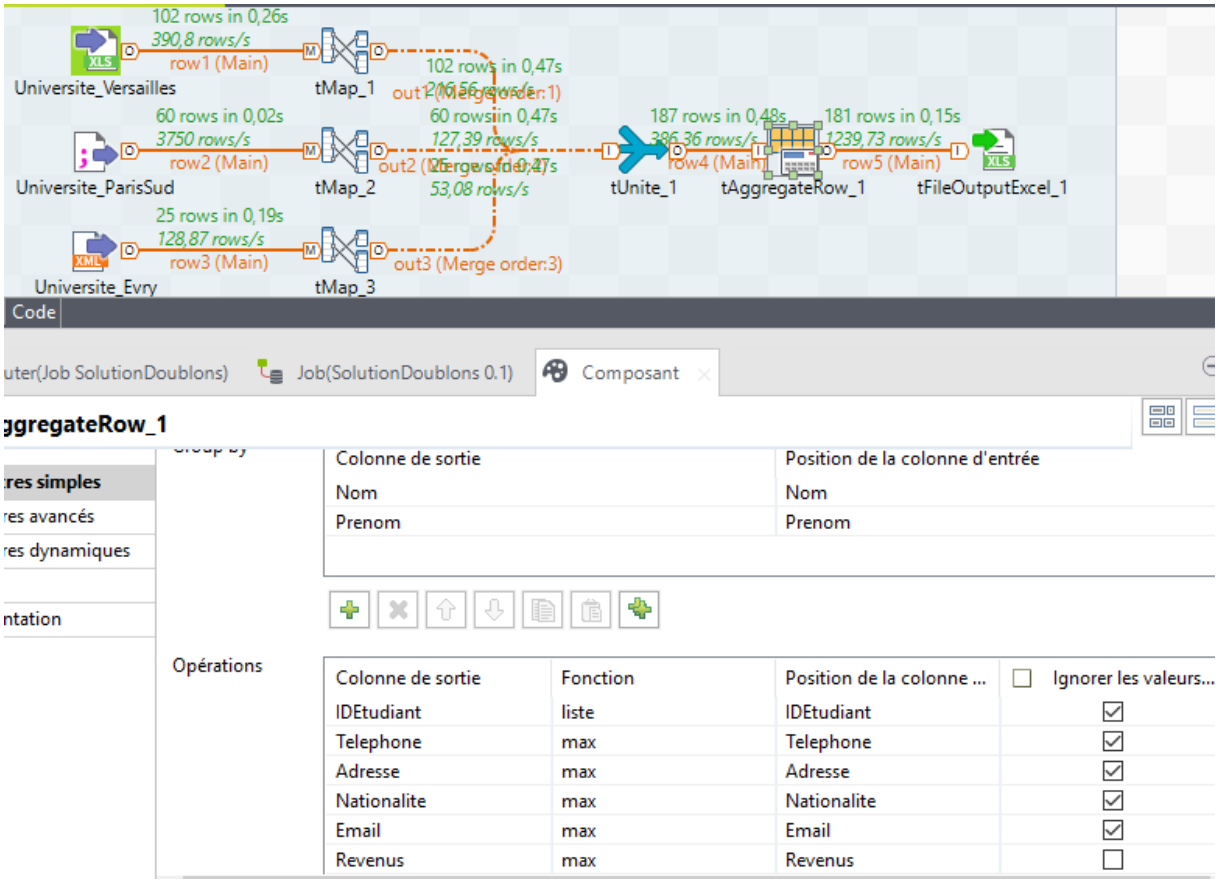

Figure : Workflow de résolution du problème de doublons

Non conformité à un format :

Objet : attributs email et numTelephone

Problème : les valeurs des attributs Email et NumTelephone ne correspondent pas aux formats valides.

Evaluation : vérifier la conformité des adresses Email et Numéros de téléphone aux expressions régulieres prédéfinies :

pour l Email : "^[a-zA-Z][a-zA-Z0-9-_.]+@[az]+.[a-zA-Z-.//-]{2,20}"

pour le numéro de téléphone : "(\\+33)([0-9])([0-9]{2}){3}[0-9]{2}||([0-9]{2}){4}[0-9]{2}"

Méthode : Nous avons utilisé le composant tFilterRow qui permet d'extraire les tuples ne vérifiant pas une condition donnée sur un attribut.

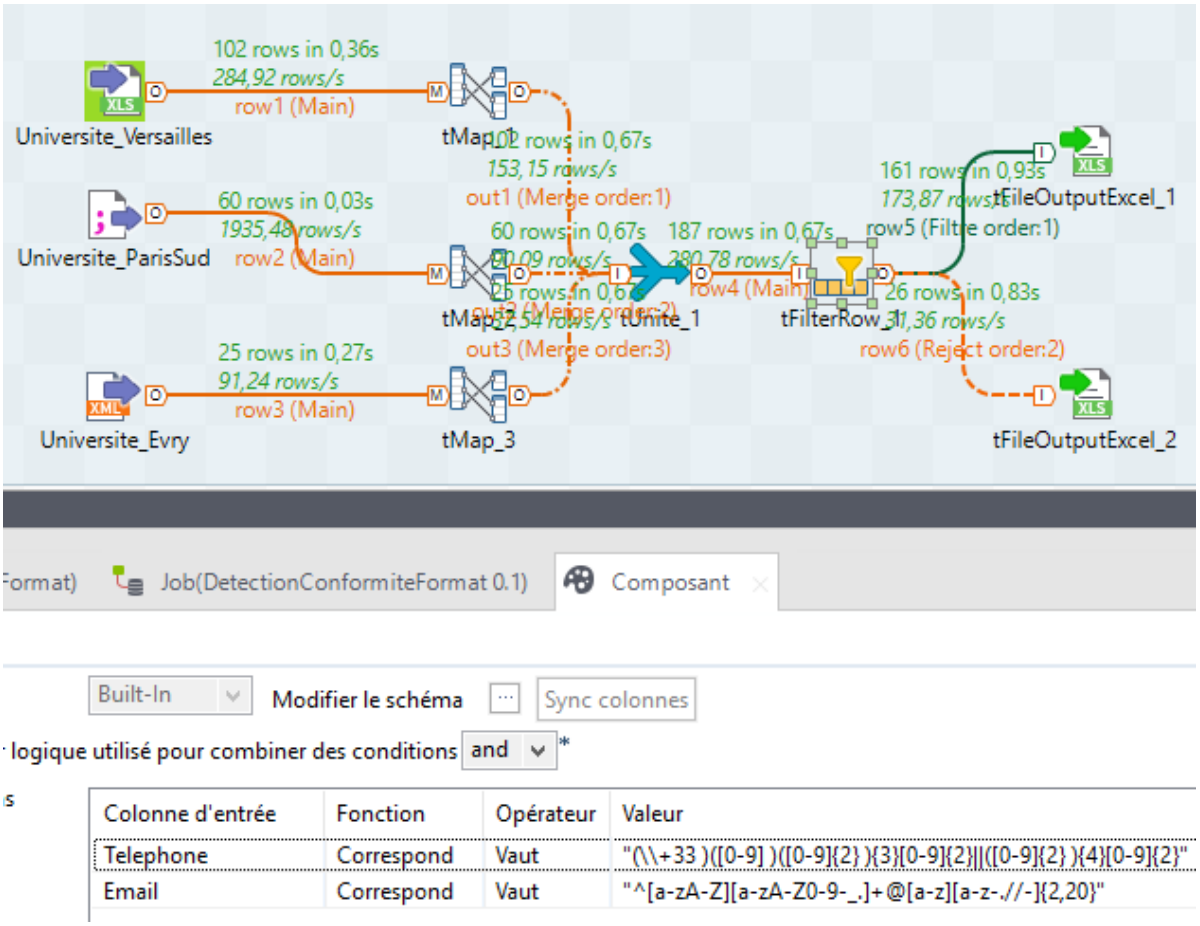

Figure : Workflow de detection du problème de conformité à un format

Résultat : 26 tuples extraits (Email ou Numero de téléphone non conforme).

Amélioration :

-Affecter a chaque étudiant n'ayant pas une adresse e-mail valide une adresse de la forme « nom.prenom@Domaine_Universite.fr »

-Supprimer les numéros de téléphone qui ne sont pas conformes aux expressions prédéfinies tout en gardant le tuple correspondant.

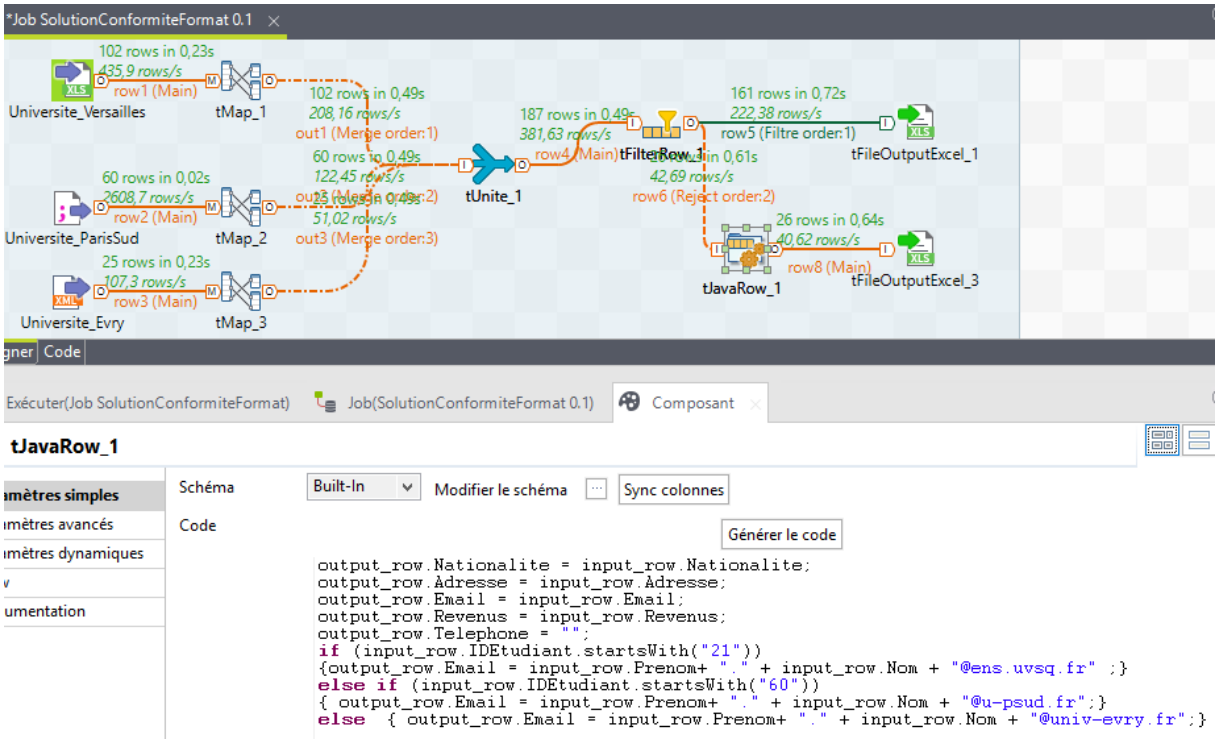

Figure : Workflow de résolution du problème de conformité à un format

⍰ Complétude des données :

Objet : attributs IDEtudient et Revenus

Problème : Un tuple qui n'a pas de valeurs dans les atttributs « IDEtudiant » ou « Revenus »

Méthode : Le composant tFilterRow qui permet de diviser un fichier selon un critère bien défini.

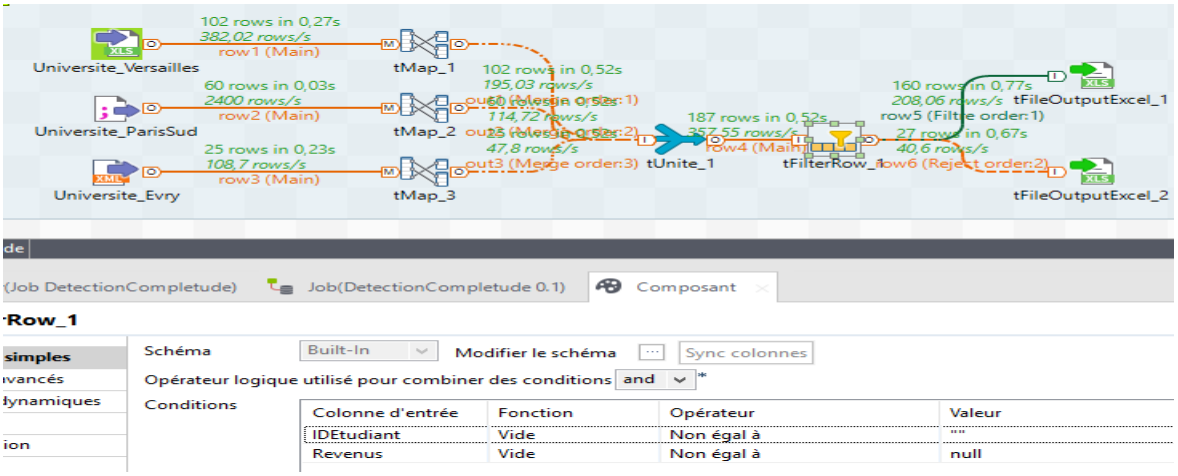

Figure : Workflow de detection du problème de complétude de données

Résultat : 27 tuples rejetés (Informations incomplètes).

Amélioration : Les étudiants avec des informations incomplètes seront séparés dans un fichier à part :

- Supprimer les tuples qui n'ont pas d'IDEtudiant

- Les valeurs NULL dans la colonne Revenus seront remplacées par une valeur par défaut '0'.

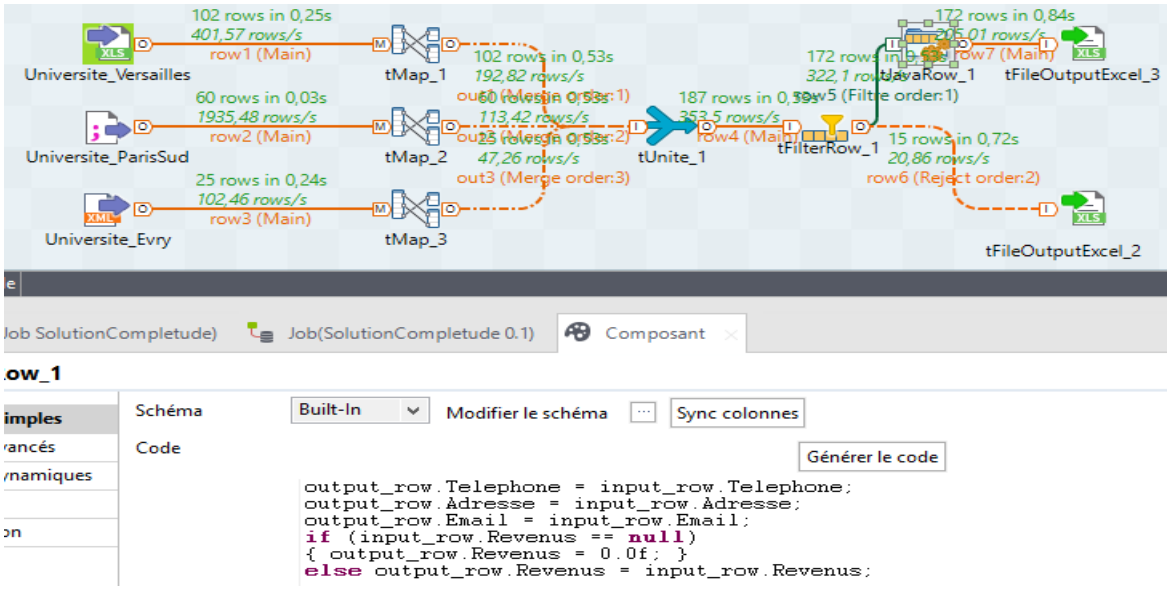

Figure : Workflow de résolution du problème de complétude de données

⍰ Hétérogénéité des échelles, de la granularité :

Objet : L'attribut Revenus

Problème : les revenus de l'étudiant ne sont pas exprimés sur la meme echelle dans toutes les sources. Ils sont données en K.Euro dans une base et en Euro dans les autres.

Evaluation :

- calculer le min, le max et la moyenne des revenus de tous les étudiants dans chaque base source, soient mean_1, mean_2, mean_3

- calculer la moyenne des moyennes mean = (mean_1+mean_2+mean_3)/3

- calculer les ratios R_i = mean_i/mean pour chaque source

- si R_i < seuil = 1/√1000 alors cette source présente un problème de granularité.

Methode : le composant tAggregateRow permet de calculer le min, le max et la moyenne pour chaque source et le compsant TjavaRow permet de faire la comparaison.

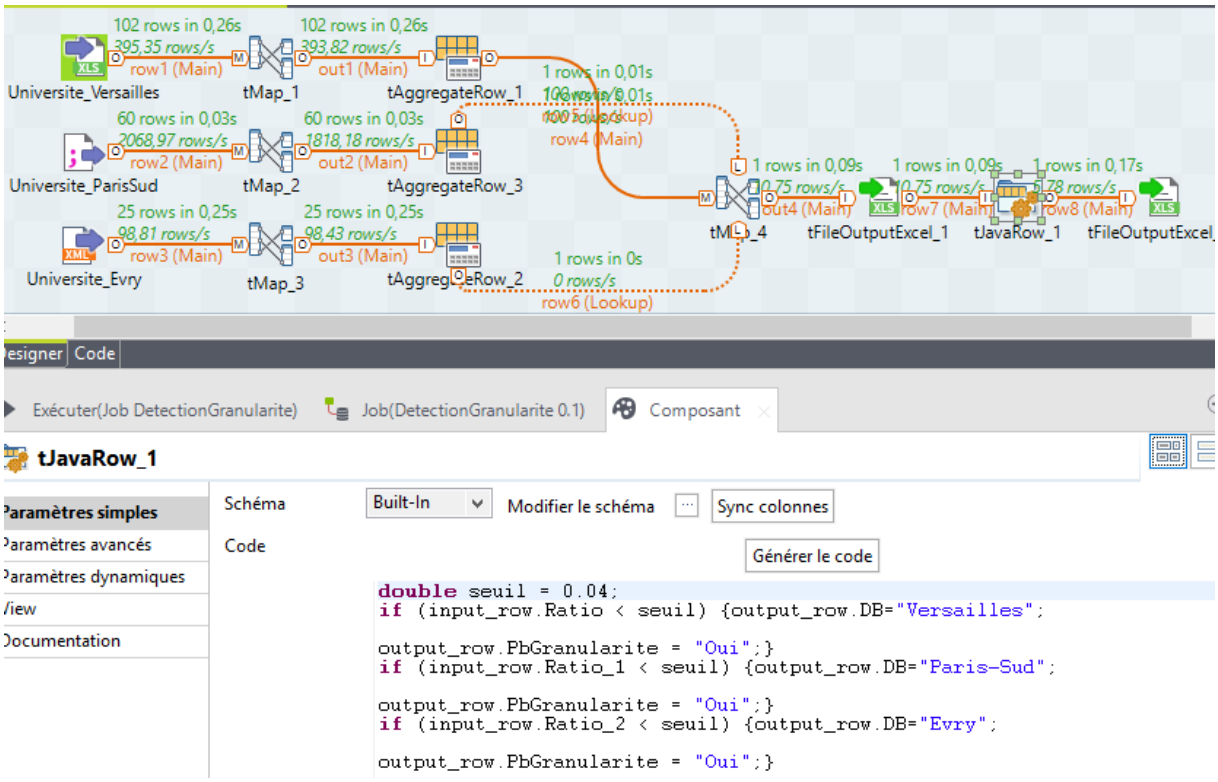

Figure : Workflow de détection du problème de différence de granularité

Amélioration: dans la base où un problème de granularité est détecté, les revenus de tous les étudiants sont multipliés par 1000.

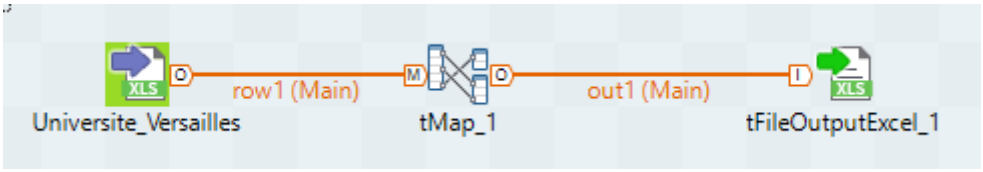

Figure : Workflow de résolution du problème de différence de granularité

WorkFlow final :

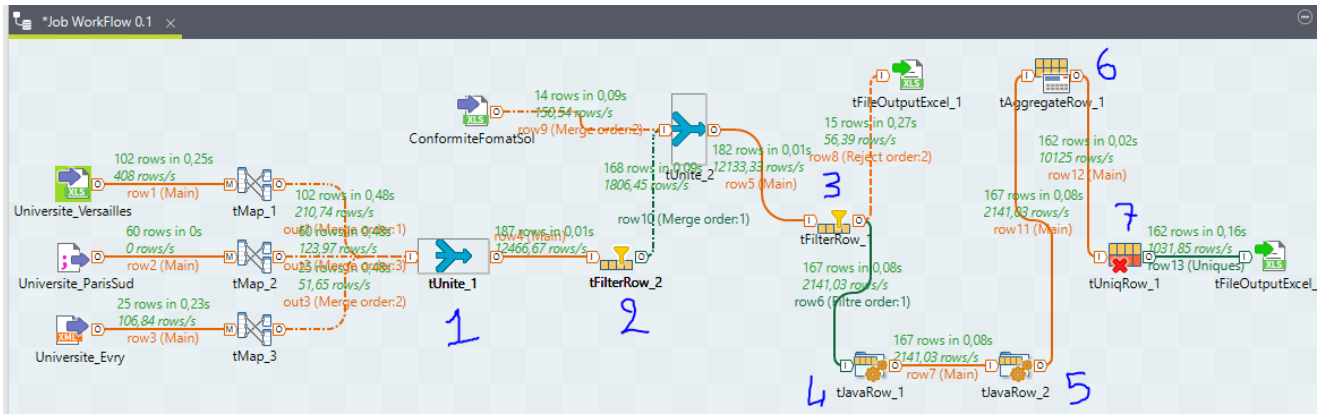

Figure : Workflow Final

1- Fusionner les sources

2- tFilterRow$_2$: Filtrage des étudiants avec des adresses mail et numéros de téléphone incorrect qui ont été corrigés dans le fichier « ConformiteformatSol »

3- tFilterRow$_3$: Filtrer les tuples d'ID étudiants NULL

4- Traiter les Revenus NULL en leur affectant la valeur 0 par défaut comme expliqué

5- détecter les valeurs de granule différent et les rendre conforme

6-7: détection et élimination des doublons

Les informations collecter lors de la détection et la résolution des problèmes liés aux différents facteurs de qualité sont stocker dans des fichiers et retournés à l'administration