Etude de cas pour une société dans le domaine: Import/Export

Réalisé Par: ***ZINABI Yassine***
IDOUHAMMOU Abderrahman
TIBA Mohamed

Encadré par ***Mme KEDAD Zoubida***

Année 2017/2018
Master 2 DataScales

Table de Matières:

Introduction

L'intégration de données est le processus qui consiste à prendre en entrée un ensemble de bases de données (schémas et populations), et à produire en sortie une description unifiée des schémas initiaux (le schéma intégré) et les règles de traduction (mapping) qui vont permettre l'accès aux données existantes à partir du schéma intégré.

- Un projet d'intégration de données implique généralement les étapes suivantes :
- Pré-intégration, une étape dans laquelle les schémas en entrée sont transformés de différentes manières pour les rendre plus homogènes (sur les plans sémantique et syntaxique).
- Recherche des correspondances, une étape consacrée à l'identification des éléments semblables dans les schémas initiaux et à la description précise de ces liens inter-schémas.
- Intégration, l'étape finale qui unifie les types en correspondance en un schéma intégré et produit les règles de traduction associées entre le schéma intégré et les schémas initiaux.

Parmi les problèmes de qualité qu'on puisse rencontrer lors de la mise en œuvre d'un projet d'intégration de données sont :

- Conformité à un format : codification .
- Hétérogénéité des échelles : granularité .
- Complétude des données .
- Présence de doublon .

Présentation du Scénario

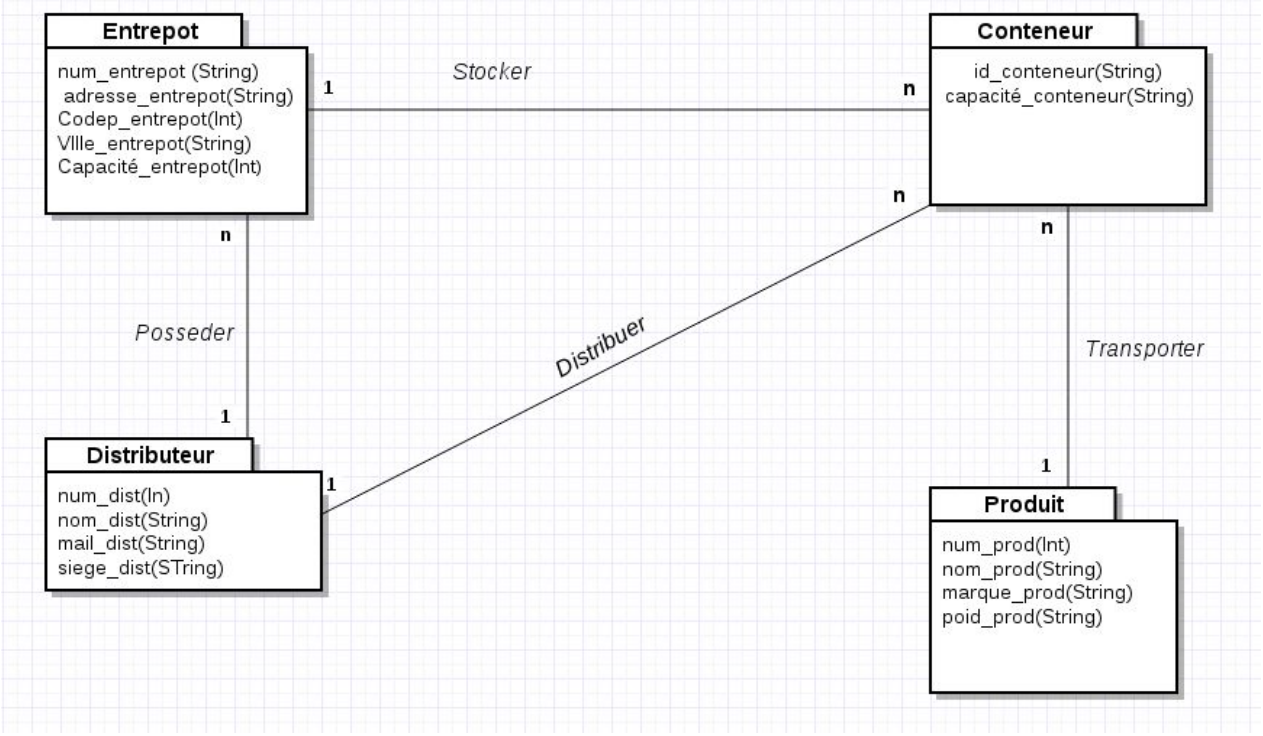
Notre travail consiste à créer un entrepôt de données d'une société dans le domaine:
Import/Export. L'entrepôt de cette société regroupe trois bases de données de types différents.

Les trois bases de données possèdes les mêmes structures mais de formats différents:

- **Entrepôts**: La liste des entrepôts avec leur id, adresse et capacités.
- **Conteneurs**: La liste des conteneurs avec leur id, capacité, numéros de leurs distributeurs et leurs produits
- **Distributeurs**: La liste des distributeurs avec leur id, noms, emails et sites.
- **Produits**: La liste des produits avec leur id, noms, marques et poids.

Notre travail consiste à intégrer ces différentes sources dans une base de données unique tout en assurant la qualité des données.

Diagramme Uml de la structure des bases de données:



Hypothèses :

⇒ Un entrepôt n'appartient qu'à un seul distributeur.

⇒ Un conteneur est distribué par un seul distributeur.

⇒ Un conteneur Transporte qu'un seul type de produit.

⇒ Un type de Produit peut être transporté dans plusieurs conteneurs.

⇒ La capacité du conteneur est mesuré en unité de poid comme le poid du produit.

⇒ Les valeurs de la capacité des conteneurs sont comprises entre 100 et 400 (kg)

⇒ Nous supposerons aussi que les unité possible pour les capacité sont :g, kg, t

⇒ 3 types de stockage de base de données seront présentés dans notre étude de cas: *XML, CSV ,SQL .*

⇒ 4 types de problèmes seront explicités dans cette étude de cas:

⇒ Problème de doublons.

⇒ Problème de granularité.

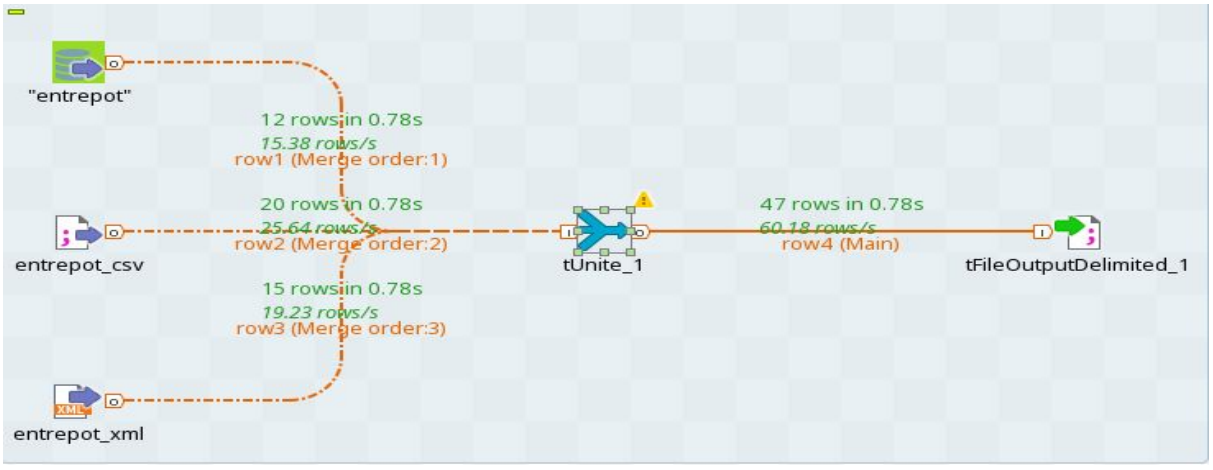
⇒ Problème de complétude.

⇒ Problème de Conformité à un format.

Réalisation de l'étude

Groupement des données

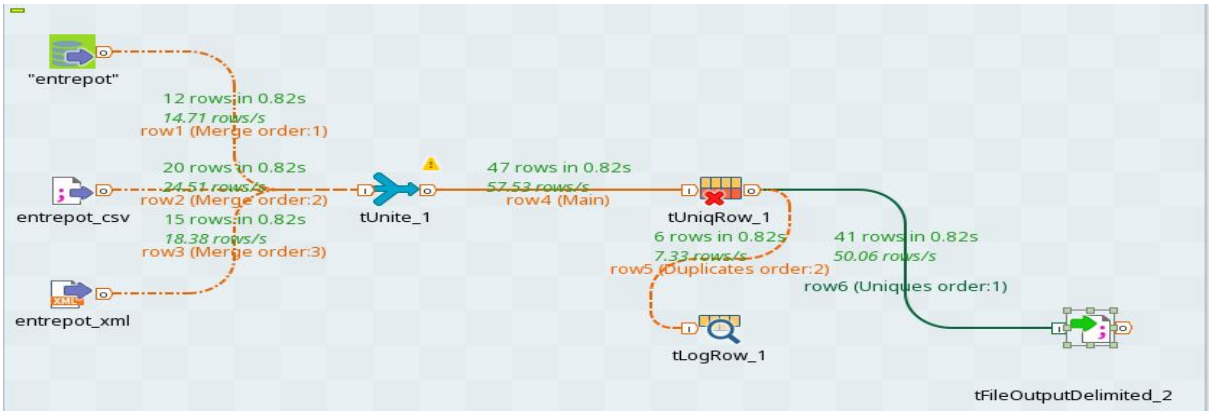
La première étape vise à regrouper les 3 différentes sources afin de détecter les problèmes en utilisant le composant Talend ***tUnit*** qui centralise des données provenant de sources diverses et hétérogènes.



Détection et amélioration des doublons

Après l'intégration des sources,le composant TUnite de Talend s'occupe des tuples égaux en faisant le produit cartésien entre les trois sources .à l'issue de ce processus le problème des doublons n'a pas été achevé à 100 %, il peut y avoir encore des tuples qui représentent le même objets .vu notre schéma de départ ,pour source un objet peut être présente avec un identifiant différent .

La détection des doublon reste un problème majeur , Talend nous a proposé un outils de détection en choisissant les critères de comparaison (Ex. *adr_ent, codep_ent, ville_ent, capacite_ent)* et supprime les tuples correspondent sans définir la source .



Le composant ***tUniqRow*** n'est pas une solution adéquate pour notre cas (BD distribués) , du coup on a proposé un amélioration qui pourrait faire en sorte que les sources auront un degré de fiabilité autrement dit , on va va définir une source de confiance sur laquelle le composant Talend sera basé pour éliminer les doublons .

Avant chaque suppression , on va comparer le degré de fiabilité des deux source ,donc le(s) moins fiable seront éliminé et stocké dans un fichier qui servira au profiling par la suite .

#5. tLogRow_1

key	value
num_ent	816136133-9
adr_ent	10 Bd Ben Tachafine
codep_ent	20300
ville_ent	Casablanca
capacite_ent	30
numdist	4

#6. tLogRow_1

key	value
num_ent	069884865-9
adr_ent	2 Kensington Park
codep_ent	92563
ville_ent	Hekou
capacite_ent	11
numdist	4

#1. tLogRow_1

key	value
num_ent	360138000-5
adr_ent	2 John Wall Drive
codep_ent	56743
ville_ent	Mpika
capacite_ent	35
numdist	5

#2. tLogRow_1

key	value
num_ent	011712132-0
adr_ent	16 Butternut Court
codep_ent	32451
ville_ent	Obonoma
capacite_ent	33
numdist	4

#3. tLogRow_1

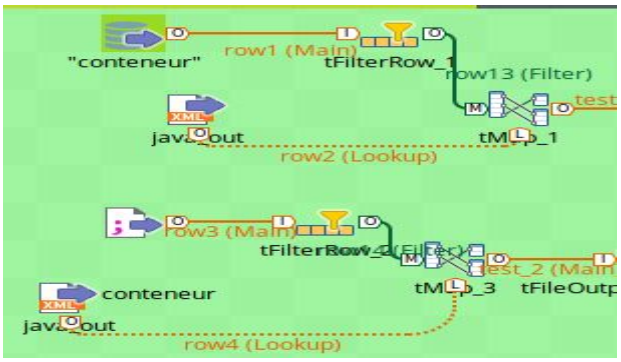
key	value
num_ent	313311252-X
adr_ent	16 Butternut Court
codep_ent	32451
ville_ent	Obonoma
capacite_ent	33
numdist	1

#4. tLogRow_1

key	value
num_ent	802097491-1
adr_ent	457 Becker Road
codep_ent	53505
ville_ent	Aga
capacite_ent	24
numdist	3

Détection et amélioration de la complétude

Les champs nulles peuvent provoquer des problèmes au niveau de la qualité d'une source. Un contact non complet coûte trop cher à l'entreprise, pour avoir des cibles

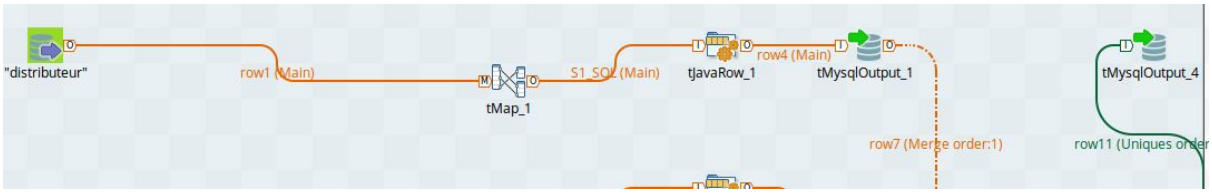


fiables , une études de complétude sur source nous donne une idée sur la fiabilité de nos données. ce problème peut provoquer d'autres comme les doublons.

L'amélioration n'est pas toujours facile. Prendre une décision sur la suppression ou non d'une cible reste à définir, c'est à dire , il faut définir les critères sur lesquels ma cible est fiable(nom, adresse,mail,..) . Dans notre cas , une donnée incomplète est automatiquement erroné .
l'étude de complétude nous montre à quel point nos sources sont fiable, l'amélioration sera faite à l'aide d'un composant Talent TfilterRow en choisissant les critères qui définissent si un tuple est fiable ou non. Comme son nom l'indique, le composant s'occupe de suppression.

Conformité à un format :

Une cible avec des données incomplet ou erroné est non fiable .un champ qui ne respecte pas une format prédéfini demande une amélioration qui peut être possible ou non comme dans la majorité des cas .
Dans notre cas , une étude sur la conformité syntaxique et sémantique a été faite sur les emails . on est parti sur l'hypothèque que distributeur possède un email (clé candidate) sous format suivante "nom_dist@gmail.com" .

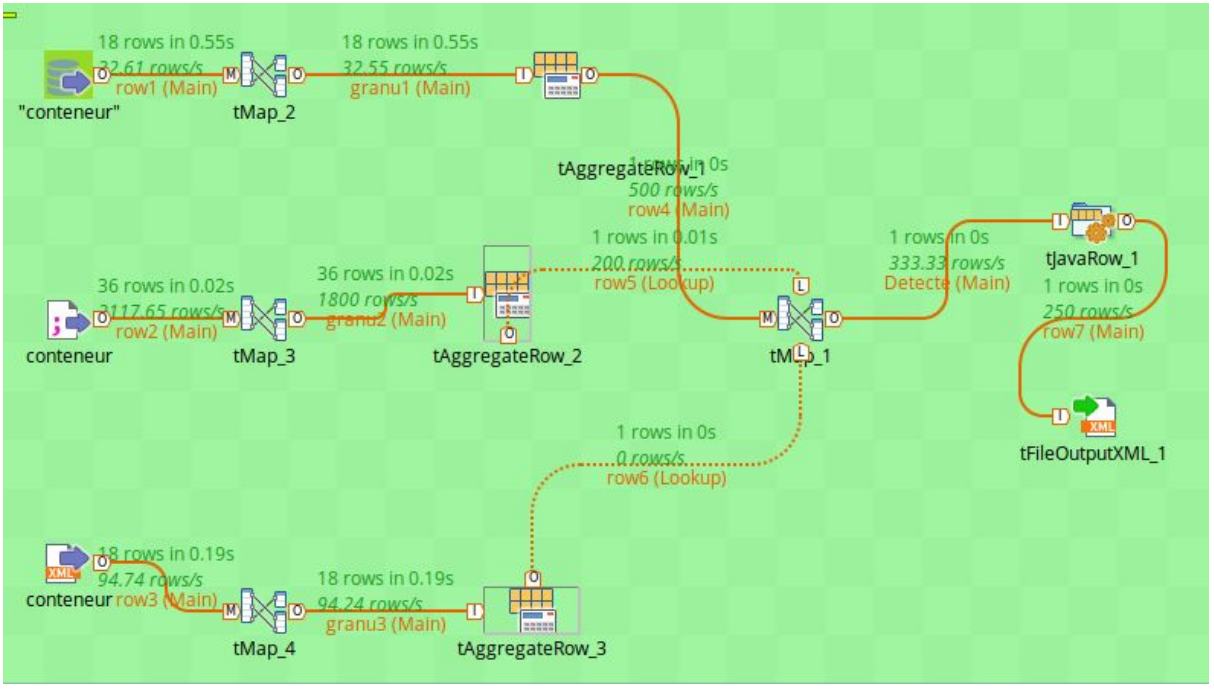


Dans un premier temps , on vérifie la syntaxe de l'email suivant une expression régulière ,après une étude sémantique et exactitude sera declencher

```
boolean b = Pattern.matches("^[\\w_.-]+@gmail.com", email );
```

l'amélioration n'est pas trop difficile sauf dans le cas où le nom de distributeur est nulle

Détection et amélioration de la Granularité



out_java.xml

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<root>
<row>
<capacite_conteneur_max_S1>400</capacite_conteneur_max_S1>
<capacite_conteneur_min_S1>100</capacite_conteneur_min_S1>
<capacite_conteneur_avg_S1>230</capacite_conteneur_avg_S1>
<capacite_conteneur_max_S2>376</capacite_conteneur_max_S2>
<capacite_conteneur_min_S2>202</capacite_conteneur_min_S2>
<capacite_conteneur_avg_S2>291</capacite_conteneur_avg_S2>
<capacite_conteneur_max_S3>376000</capacite_conteneur_max_S3>
<capacite_conteneur_min_S3>110000</capacite_conteneur_min_S3>
<capacite_conteneur_avg_S3>252888</capacite_conteneur_avg_S3>
<limite>252888</limite>
<coefession>1</coefession>
<coefessionS1>1101</coefessionS1>
<coefessionS2>870</coefessionS2>
<coefessionS3>1</coefessionS3>
<problemeS1> S3</problemeS1>
<problemeS2> S3</problemeS2>
<problemeS3> S1 S2</problemeS3>
</row>
</root>
```

Pour la détection de granularité nous utilisons la méthode citée dessous qui prend en entrée les valeurs du minimum, maximum et moyenne de la capacité pour chaque source et qui en sortit nous produira :

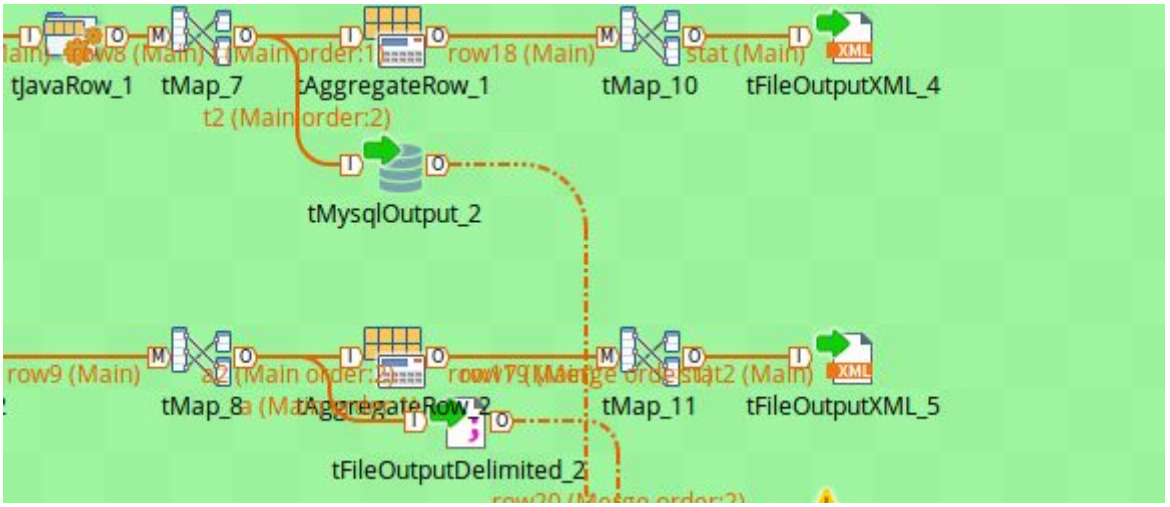
⇒ les même champs en entrée.

⇒ 3 champs coefficients qui correspondent aux 3 sources.
⇒ 1 champ limite qui sera le maximum des moyennes des sources.
⇒ 3 champs probleme dont chacun affichera pour chaque source les sources dont la granularité diffère avec celle ci.

les champs coefficients sont obtenus de la manière suivante:
nous calculons pour chaque source le quotient suivant: la somme des moyennes de toutes les sources divisé par la source actuelle .

le champ limite représente le maximum entre les moyennes de toutes les sources.

les champs problèmes sont obtenus grâce a une règle de comparaison (deux a deux) entre les sources qui stipule :
Si le résultat de la division de la moyenne (même chose pour max et min)d'une première source par la moyenne (max et min) d'une deuxième source est compris entre ¼ et 4 alors nos 2 sources ont la même granularité.
l'obtention des valeurs ¼ et 4 se fait de la manière suivante:
sachant que notre hypothèse met en évidence que la capacité d'un conteneur est comprise entre 100 et 400 alors nous étudions les cas extrêmes :
⇒ si une source 1 a comme moyenne de la capacité la valeur 100 et une autre (S2) 400 alors :
nous obtenons le quotient suivant :
100/400= ¼ .
⇒si l'ordre change : (moyenne de capacité s1=400, et moyenne de capacité s2=100)
alors:
nous obtenons le quotient suivant:
400/100=4 .
Donc 2 sources partagent la même granularité si le calcul de la division l des agrégats min max moy entre les 2 sources est compris entre ¼ et 4.



Pour la partie amélioration nous associons a chaque source un fichier contenant l'ensemble des des champs créés à partir de la détection (étape précédente) .
l'hypothèse est la suivante : nous savons que nous avons 3 unités: g,Kg,T
nous avons choisis d'unifier toutes les capacité en une seul unité qui sera le gramme
Pour chaque source nous utiliserons le coefficient correspondant .
L'utilisation du coefficient sera effectué de la manière suivante:
-si le coefficient est inférieur à 400 alors la correction de la capacité n'est pas nécessaire .
-si le coefficient est compris entre 500 et 3000 alors la correction sera de multiplier la capacité par la valeur 1000 .(correction d'une capacité en Kg)
-si le coefficient est supérieur à 1500 alors la correction sera de multiplier la capacité par 1 000 000.(correction d'une capacité en T).

Profilage:

Après la détection et la correction des problèmes, on stock dans des fichiers XML les statistiques et les informations sur les traitements fait:

Statistique_Granularite_1.xml

```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<root>
<row>
<Id_Statistiques>0</Id_Statistiques>
<Message>Nombre de Conteneur filtré selon la granularité dans la source 1</Message>
<Nombre_de_tuples>15</Nombre_de_tuples>
<Date>02-11-2017</Date>
</row>
</root>
```

Statistiques_des_doublons.xml

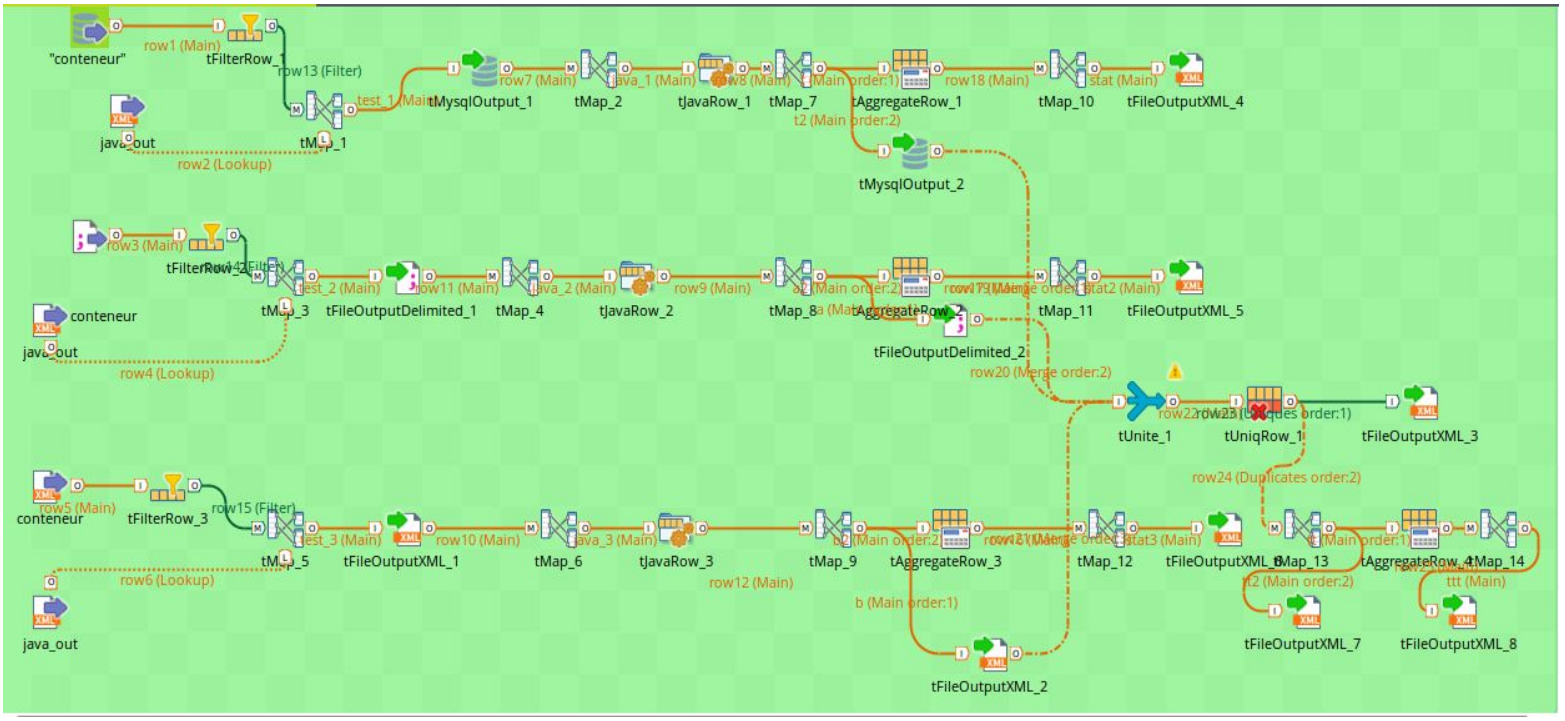
```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<root>
<row>
<Id_Statistiques>0</Id_Statistiques>
<Message>Nombre de Conteneur filtré selon les doublons</Message>
<Nombres_de_tuples>16</Nombres_de_tuples>
<Date>02-11-2017</Date>
</row>
</root>
```

Pourcentage1.xml

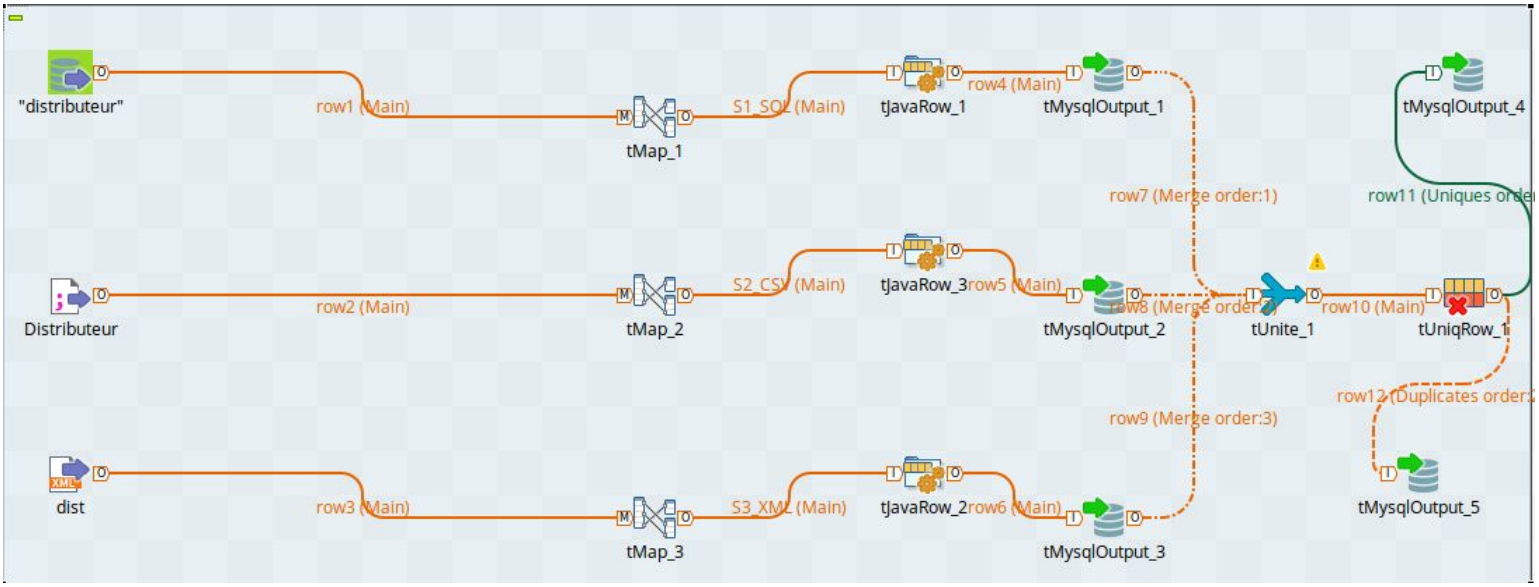
```xml
<?xml version="1.0" encoding="ISO-8859-15"?>
<root>
<row>
<Pourcentage>23.188406%</Pourcentage>
</row>
</root>
```

WorkFlow Final:

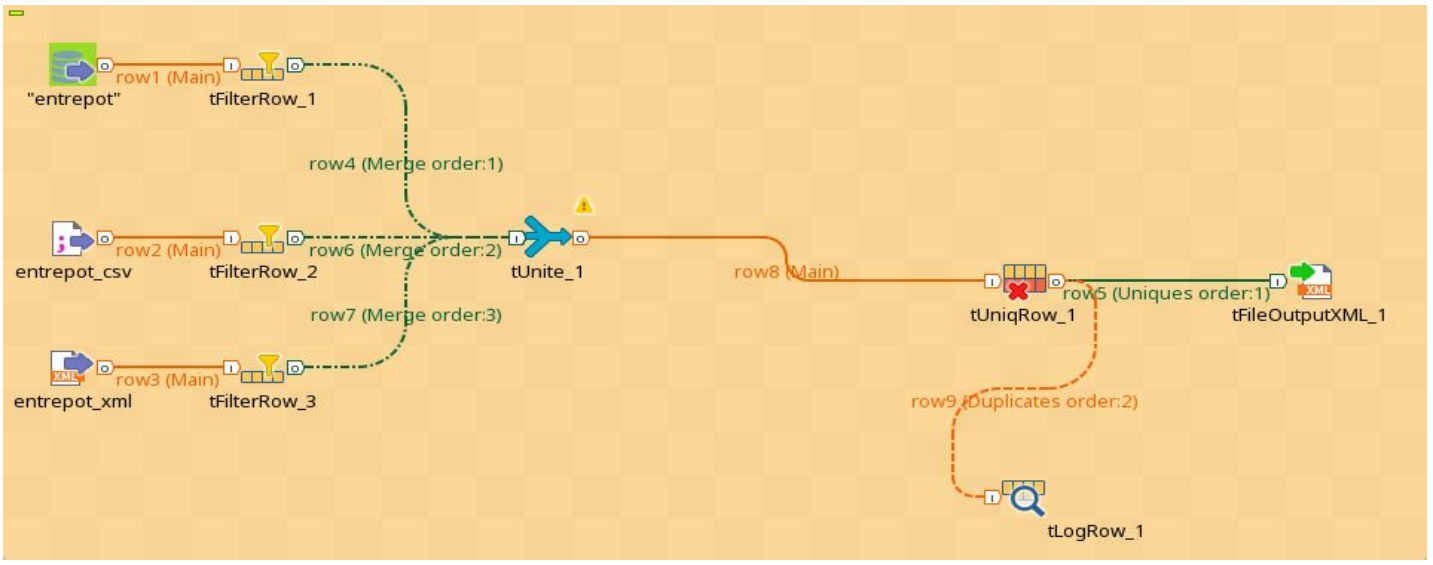
➔ La table Conteneur:



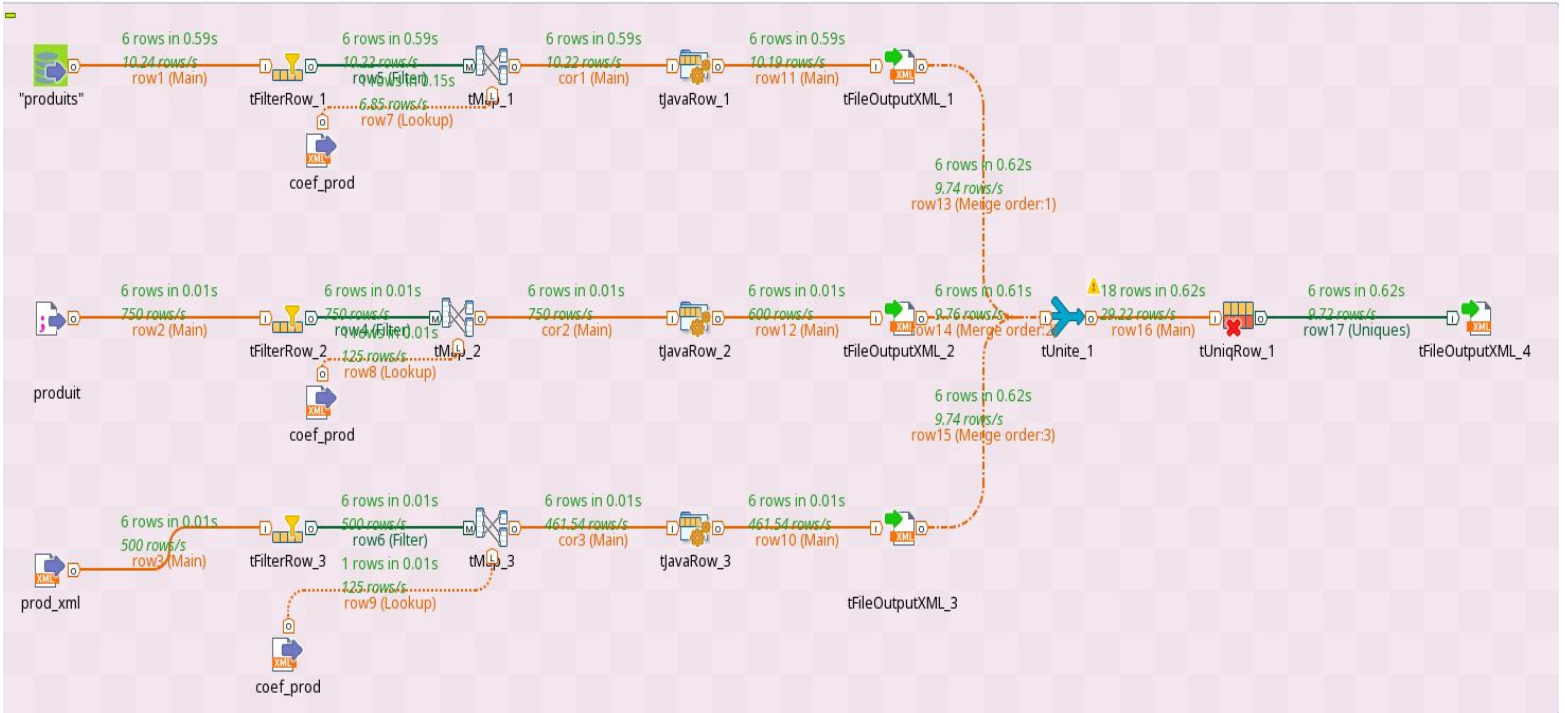
➔ La table Distributeur:



➔ La table Entrepôt:



➔ La table Produit:



Conclusion

Notre cas d'étude avait comme objectif d'intégrer des différentes bases de données dans un seule entrepôt de données cible.
Ce travail a été divisé en deux partie, une dédié à la détection des problèmes de qualité de données, l'autre l'amélioration de cette qualité de données.
Durant la réalisation du travail nous avons pris connaissance de l'ETL Talend qui nous a permet d'évaluer et améliorer la qualité des données sur les métrique que nous avons posé au début de cas d'étude.