TP1

Intégration de données en mode ETL avec TOS DI

© By Mourad Ouziri
Mourad.Ouziri@ParisDescartes.fr

Programme-objectifs :

- Ingestion de données de différents formats (CSV, Bases de données, XML, JSON).
- Enrichissement de données par croisement.
- Programmation en Java de traitements de transformation et de croisement personnalisés.

Documentation :

- https://help.talend.com/display/HOME/Talend+Open+Studio+for+Data+Integration

Partie 1 : Chargement de données

1. Chargement de fichiers de données en CSV et Excel

Manipulation 1:	Ajouter un fichier csv dans le Repository de Talend.

Manipulation 2:	Charger les données du fichier à l'aide du composant tFileInputDelimited.

Manipulation 3:	Afficher les données avec tLogRow.

Manipulation 4:	Enregistrer les données en résultat dans un fichier csv puis excel.

Manipulation 5:	Refaire le même travail à partir d'un ficher Excel à l'aide de tFileInputExcel.

2. Chargement dans une base de données

Manipulation 6:	Ajouter une connexion à une base de données MySQL au projet.

Manipulation 7:	Charger les données du fichier à l'aide du composant tMySQLInput.

Manipulation 8:	Afficher les données à l'aide du composant tLogRow.
Manipulation 9:	Filtrer les données et afficher le résultat.

Manipulation 10:	Enregistrer les données dans un fichier (csv, excel, etc.).

3. Sélection de données

Manipulation 11:	Sélectionner les données (par ville d'habitation, par âge ou par sexe des

clients) chargées de fichiers CSV avec tFilterRow (utiliser le mode avancé).
Manipulation 12: Stocker les résultats dans deux fichiers différents (parisiens/non parisiens, adultes/enfants, hommes/femmes). Utiliser tMap.
Manipulation 13: Rendre le job dynamique à l'aide tMsgBox (de type Question) permettant de saisir le critère de sélection de manière dynamique (la valeur saisie est récupérée dans la *globalMap* par la clé NomDuCompossnt_RESULT).

4. Sélection de données dynamique

Manipulation 14: Reprendre les travaux précédents et rendre les jobs dynamiques à l'aide tMsgBox (de type Question) afin de permettre la saisie du critère dans une boite de dialogue (la valeur saisie est récupérée dans la *globalMap* par la clé NomDuCompossnt_RESULT).

5. Chargement de données CSV dénormalisées

Manipulation 15: Utiliser le composant tNormalize pour normaliser (selon la première forme normale du modèle relationnel) les données d'un fichier csv avant de les insérer dans la base de données.

Manipulation 16: Utiliser le composant tDenormalize pour dénormaliser les données (issues d'une base relatonnelle) dans un fichier csv.

6. Chargement de données CSV en ligne et transformation

Manipulation 17: Utiliser le composant tPivotToColumnsDelimited pour remettre les données d'une même entité éclatées sur plusieurs lignes en colonne d'attributs (Pivot column : attribut, Aggregatition column : valeur, Aggregation function : first).

Manipulation 18: Afficher le résultat avec tLogRow puis les stocker dans un nouveau fichier.

7. Chargement de données XML et JSON

Manipulation 19: Ajouter un fichier XML dans le Repository de Talend.

Manipulation 20: Charger les données XML à l'aide du composant tFileInputXML.

Manipulation 21: Afficher les données sous forme de table avec tLogRow puis les enregistrer dans une base de données.
Manipulation 22: Filtrer les données (par année de naissance par exemple, utiliser la fonction de Talend *TalendDate.getPartOfDate* pour extraire l'année d'une date complète) et afficher le résultat.

Manipulation 23: Faire le même travail avec des données JSON avec tFileInputJSON.

8. Utilisation de variables de contexte

Manipulation 24: Créer deux contextes différents : apprentissage et production qui vont variabiliser les valeurs absolues définies dans les composants.

Manipulation 25: Pour les composants tFileInput, créer plusieurs variables stockant les répertoires des fichiers de données utilisés en entrée, une variable pour chaque nom de fichier.
Manipulation 26: Remplacer les valeurs absolues saisies dans les composants tFileInput par la variable de contexte correspondante.
Manipulation 27: Réexécuter les jobs dans les contextes créés.

Partie 2 : Croisement de données

9. Croisement de données CSV

Manipulation 28: Faire le croisement des deux fichiers CSV avec tMap.
Manipulation 29: Afficher le résultat du croisement avec tLogRow puis l'enregistrer dans un fichier sur disque avec tFileOutpuDelimited.

10. Croisement de bases de données

Manipulation 30: Injecter deux bases de données (de clients) avec tMySQLInput.
Manipulation 31: Faire le croisement des deux bases avec tMap.
Manipulation 32: Afficher le résultat du croisement avec tLogRow puis l'enregistrer dans un nouveau fichier.

11. Personnaliser le croisement avec les propriétés de tMap

Manipulation 33: Utiliser les fichiers CSV et les bases de données traités ci-dessus. Les modifier si nécessaire pour adapter leur contenu à ces tâches.
Manipulation 34: Appliquer un filtre sur la table de jointure et afficher les résultats.
Manipulation 35: Afficher les lignes (clients) n'ayant pas satisfait la condition du filtre (nouvelle sortie avec l'option « catch reject » à true).
Manipulation 36: Changer le type de jointure à « Inner join » et afficher les résultats.
Manipulation 37: Afficher les lignes (clients) qui ne répondent pas à ce type de jointure/croisement (nouvelle sortie avec « catch inner join » à true).
Manipulation 38: Changer le nombre de correspondances et ré-exécuter les jobs (CSV et bases de données). Afficher les résultats avec tLogRow.
Manipulation 39: Pour une jointure interne, ajouter une sortie des lignes non satisfaites et

l'afficher avec tLogRow.

12. Croisement de données CSV, XML et JSON

Manipulation 40: Insérer un fichier XML dans le Repository.
Manipulation 41: Lire les données du fichier XML à l'aide de tFileInputXML.
Manipulation 42: Faire le croisement de ces données XML avec les données CSV puis avec une base de données avec tMap.
Manipulation 43: Afficher le résultat du croisement avec tLogRow puis l'enregistrer dans un nouveau fichier.

Manipulation 44: Croiser des données JSON avec XML et afficher/enregistrer le résultat en XML puis en JSON.

Partie 3 : Croisement de données internes avec de données externes

13. Ingestion de données de services Web REST et SOAP

Manipulation 45: Collecter les données (JSON et XML) de services Web REST (Velib, Google Maps, Paris connect et d'autres) à l'aide de tREST et les afficher avec tLogRow.

Manipulation 46: Utiliser tExtractJSONFields ou tExtractXMLField selon le format des réponses pour extraire les données JSON ou XML et les enregistrer dans un fichier texte (au format CSV) puis dans une base de données MySQL.

Manipulation 47: Collecter les données de service Web SOAP (conversion de devises, informations météo, informations pays) à l'aide de tSOAP et les afficher avec tLogRow.

Manipulation 48: Utiliser tExtractXMLField pour extraire les données SOAP/XML et les enregistrer dans un fichier texte (au format CSV) puis dans une base de données MySQL.

Manipulation 49: On voudrait récupérer les stations Velib de plusieurs villes fournies dans un fichier csv. Utiliser tFlowToIterate pour réexécuter le service Web tREST pour chaque ligne récupérée du fichier csv (penser à reparamétrer tREST).

14. Croisement de données de services Web

Manipulation 50: Utiliser les données issues des services Web pour enrichir les bases internes avec tMap.