

- *Système d'Aide à la Décision* -

TP 1 : Pentaho Data Integration (PDI)

Transformation vs. Tache

Exercice 1 :

1. Créer une nouvelle transformation.
2. Générer 10 lignes avec une colonne nommée « *val_1* » de valeur « 28 » (Génération lignes).
3. Ajouter une colonne nommée « *val_2* » de valeur « 30 » (*Ajout constantes*).
4. Modifier la première étape pour ajouter une colonne « *val_3* » de valeur « *bonjour* ».
5. Ajouter deux colonnes (*Création d'opérations de calcul*) :
 - Colonne « *hexa_val3* » va recevoir la valeur hexadécimale du champ « *val_3* »,
 - Colonne « *somme* » va recevoir la somme des deux champs « *val_1* » et « *val_2* ».
6. Numéroter les lignes (*Ajout séquence*).
7. Exporter le résultat dans un fichier Excel (*Alimentation*).

Exercice 2 :

1. Créer une nouvelle transformation.
2. Lire le fichier « *personnes.csv* » (*Extraction depuis fichier CSV*).
3. Ecrire les noms en majuscule, les prénoms en minuscule (*Manipulation de chaînes de caractères*).
4. Ajouter une colonne « *msg* » qui contient la chaîne de caractère « *Bonjour* » plus le nom de chaque personne (*Appel script interprété Rhino*).

5. Créer une nouvelle colonne qui contient le nom avec la première lettre en majuscule et le reste en minuscule (*Appel script interprété Rhino*).
6. Créer une autre colonne qui contient les 3 premières lettres du nom (*Extraction depuis chaînes de caractères*).
7. Calculer la moyenne d'âge par pays (*Agrégation sans tri*) et sauvegarder le résultat dans un fichier Excel (*Alimentation fichier MS Excel*).
8. Trier les lignes par ordre alphabétique du nom (tri lignes).
9. Sauvegarder la sortie du résultat précédent en deux formats :
 - Un fichier CSV.
 - Une table MySQL : vous devez tout d'abord télécharger le jar *mysql-connector-java-5.1.48* et le placer dans le répertoire {Pentaho_home}\data-integration\lib.
10. A partir du fichier csv de base :
 - Décomposer la colonne « *tel* » en deux colonnes : « *indicatif* » et « *telNum* » (*Décomposition Champs*).
 - Exporter les deux colonnes en fichiers csv.

Exercice 3 :

Le rôle d'un commissionnaire de transport est de recevoir les commandes des clients et de les attribuer à des transporteurs. Un nouveau logiciel a été implanté dans la société pour suivre les commandes en temps réel et on souhaite donc pouvoir comparer les horaires de chargement et de livraison demandés par le client et les horaires réels obtenus grâce au nouveau logiciel. Ceci permettra d'évaluer les différents retards au niveau des usines (*retard pour la préparation de la commande par exemple*) mais également ceux qui concernent le transporteur (*retard à la livraison par exemple*).

Vous avez deux fichiers à disposition :

- « **orders.csv** » : contient l'ensemble des commandes envoyées par le client ainsi que le transporteur attribué à chaque commande.

- « **realtime.csv** » : contient les commandes exploitées dans le logiciel de suivi. On y retrouve donc certaines commandes avec des horaires réels (*pas toujours complets*).

On veut procéder à la jointure des deux fichiers. Pour cela, nous devons procéder en deux étapes car **une jointure nécessite d'avoir trié les données au préalable**.

1. Créer une nouvelle transformation et ajouter les deux fichiers sources.
2. Faire un tri sur les deux fichiers par ordre croissant des champs **OrderNumber** (*Tri lignes*).
3. Ajouter une étape de jointure (*jointure comparaison lignes*) qui permet de garder toutes les lignes d'orders.csv même s'il n'y pas de correspondance avec le fichier realtimes.csv

Pour l'analyse future de ses données, l'entreprise a souhaité que les horaires manquants soient considérés comme une "vraie" information. Cela veut dire qu'il ne veut pas de valeurs nulles dans notre table mais plutôt une information du type "données inconnues" lorsqu'il manque une date ou un horaire.

4. Remplacez les types des dates et des horaires réels avec une information de type String et insérer par la suite une étape (*Remplacer valeur nulle*) qui va mettre en évidence le nouveau besoin de l'entreprise.
5. Supprimez la colonne **OrderNumber_1** de votre flux de données (*Altération structure de flux*).
6. Exportez le résultat en deux formats: un fichier Excel et une table MySQL.

Exercice 4 :

La tâche qui sera effectuée dans cet exercice aura pour but d'exécuter la transformation de l'exercice précédent, de renvoyer un message en cas d'erreur et de vérifier que le fichier de sortie est bien créé en fin de transformation.

1. Créer une nouvelle tâche.
2. Insérer un élément « *Start* », un élément « *Exécution Transformation* » et un élément « *Vérification existence fichier* ».

3. Insérer deux éléments (*Mise en échec tâche*). Relier le premier avec l'étape de transformation et le second avec la vérification.

Les liens n'ont pas tous la même couleur :

- **Un lien bleu** avec un cadenas indique que l'élément suivant sera toujours exécuté.
- **Un lien rouge** indique que l'élément suivant ne sera exécuté que s'il y a eu une erreur dans l'exécution de l'élément précédent.
- **Un lien vert** indique que l'élément suivant ne sera exécuté que si l'élément précédent s'est terminé avec succès.

On veut mettre la tâche en échec dans le cas où la transformation échoue d'une part mais également lorsque le fichier généré par la transformation n'existe pas.

4. Établir les types de liens créés précédemment afin de répondre à ce besoin.
5. Configurer les éléments : étape de transformation ; fichier dont on veut vérifier l'existence ; messages d'erreur.
6. Exécuter la tâche.
7. Afin de créer des tâches qui lancent successivement différentes transformations, ajouter dans la même tâche l'une des transformations créées précédemment avec un délai d'attente de 10 secondes entre les deux transformations (*Mise en place Temporisation*).
8. Ajouter une étape d'envoi d'un courriel si la 2^{ème} transformation a été effectuée avec succès (*Envoi Courriel*). Poursuivre cette étape avec un test de succès.
 - *Serveur SMTP : smtp.gmail.com*
 - *Port : 465*