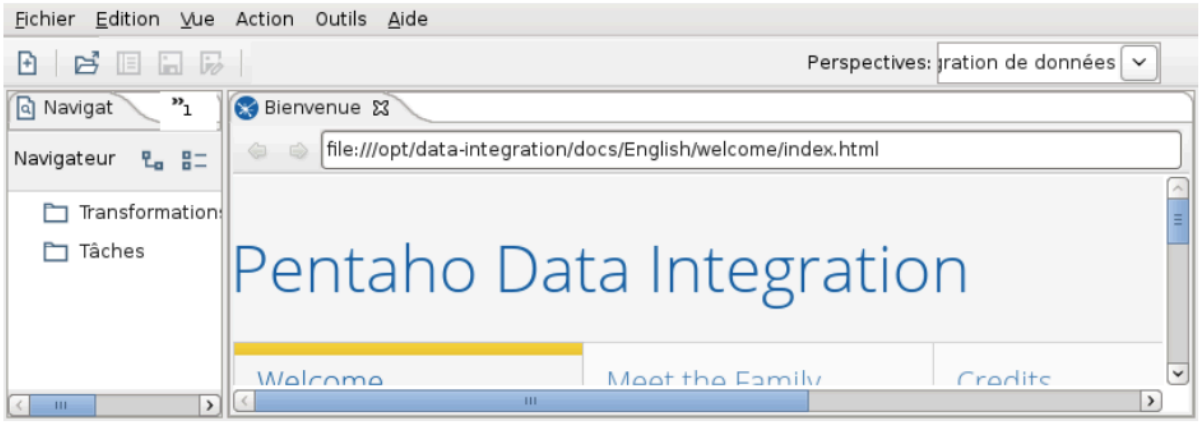
BI M2DAC – TP1 ETL

Ludovic Denoyer - Laure Soulier

Configuration Pentaho PDI :

- Installer Pentaho PDI (7.0) dans le répertoire /tmp de l'ordinateur http://community.pentaho.com/projects/data-integration/
- Téléchargez le fichier `http://ftp.mozilla.org/pub/mozilla.org/xulrunner/nightly/2012/03/2012-03-02-03-32-11-mozilla-1.9.2/xulrunner-1.9.2.28pre.en-US.linux-x86_64.tar.bz2`
- Décompressez ce fichier dans /tmp – le répertoire /tmp/xulrunner devrait être créé
- Dans le fichier spoon.sh, à la ligne « OPT="$OPT $PENTAHO_DI_JAVA_OPTIONS» , rajoutez l'option suivante `-Dorg.eclipse.swt.browser.XULRunnerPath=/tmp/xulrunner`
- Lancer Pentaho PDI : sh spoon.sh



Créer des ETL avec Pentaho PDI :

- Fichier → Nouveau → Transformation
- Vous avez un catalogue de chargement/transformation/exportation de données dans l'onglet « Palette de création »

Exercice 1 :

- Générer 10 lignes avec un champs *test* de valeur = 1 – (**générer lignes**)
- Rajouter une colonne *gender* de valeur *male* (**Ajout constantes**)
- Prévisualiser le résultat -> (Bouton droit -> prévisualiser puis fenêtre du bas : prévisualiser)
- Rajouter une colonne *cola* dans la génération de lignes de valeur *test*
- Rajouter un checksum après la génération de ligne (**Ajout checksum**)
  - De type CRC32
  - Récupérer les champs en entrée
  - Le champ de sortie sera *checksum*
- Numérotez les lignes en utilisant **Ajout séquence**
- Exporter le résultat dans un fichier XML **(XML Output)**

Exercice 2 :

- Créez un fichier CSV contenant une colonne *cola* dont les 10 premières valeurs valent 1 et les 10 suivantes valent 2
- Numérotez les lignes dans un champs *index*
- Numérotez les lignes dont cola vaut 1 et les lignes dont cola vaut 2 indépendamment à l'aide de **Ajout séquence ré-initialisable** dans un compteur *colb*
- Rajouter une colonne à l'entrée de valeur *cola/colb* grâce à un **Calculateur** en utilisant la formule *[cola]/[colb]*

Exercice 3 :

- Importer le fichier noms-prenom.csv
- A l'aide de **Manipulation de chaînes de caractères,** écrire les noms en majuscule, les prénoms en minuscule et les pays avec la première lettre en majuscule
- Créer une troisième colonne qui contient les 3 premières lettres du nom de famille à l'aide de **Calculateur** ainsi que de **Extraction depuis chaînes de caractères**
- Trier par ordre lexicographique sur le nom
- Sauvegarder la sortie dans un fichier CSV
- Sauvegarder la sortie dans un fichier Excel et ouvrez le fichier

Exercice 4 :

- Téléchargez le fichier **Mobiliers de Stationnement** depuis le site *http://opendata.paris.fr* au format CSV
- Créer un fichier qui ne contient que les coordonnées x et y des stations Velib
- Sauvegarder la sortie au format JSON

Exercice 5 :

Soit le fichier titanic.csv (le récupérer sur kaggle)

- Compter le nombre de survivants pour toutes les combinaisons des champs de classe et sexe
- Calculer la fréquence des survivants pour ces mêmes combinaisons

Exercice 6 :

Récupérer des données depuis http://opendata.paris.fr/api/records/1.0/search?dataset=stations-velib-disponibilites-en-temps-reel

- Pour cela, il faut créer une colonne url contenant l'URL
- Utiliser l'icône **Client REST**
- Sauvegardez la sortie dans un fichier

Exercice 7 :

- Importer le fichier noms-prenom.csv
- Calculer la moyenne d'âge par pays - Exportez le fichier afin de pouvoir l'utiliser dans Google Map : https ://developers.google.com/chart/interactive/docs/gallery/geochart