

PDI

PENTAHO Data Integration

Planification ETL

Identification des Sources et Destinations de Données

- Les sources de données
- Périodes d'extraction et de chargement
 - Extraction
Définir les fenêtres d'extraction pour chaque source de données, les données peuvent éventuellement être stockées dans des bases de données temporaires
 - Chargement
- Estimer

Evaluation des Données Sources

- Structure et format des données
 - BDR, Tableurs, csv, xml, services web, fichiers plats
- Format des données
- Evaluer le volume des données
- Identifier les données à extraire
 - Il faut sélectionner uniquement les lignes nouvellement créées ou modifiées depuis la dernière extraction.
- Identifier les anomalies

PDI

- Pentaho Data Integration (anciennement K.E.T.T.L.E – Kettle ETLT Environment) est un E.T.T.L,
 - Extraction
 - Transport
 - Transformation
 - Loading.

Concepts PDI

- Transformations
- Jobs

Composants PDI

- SPOON: un EDI pour créer les transformations et les jobs.(ou tâches)
- Kitchen: outil en ligne de commande pour exécuter les jobs.
- Pan: outil en ligne de commande pour exécuter les transformations.
- Carte: un serveur léger pour exécuter les jobs et les transformations sur un serveur distant.

SPOON

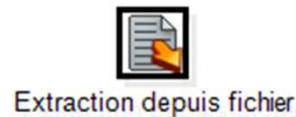
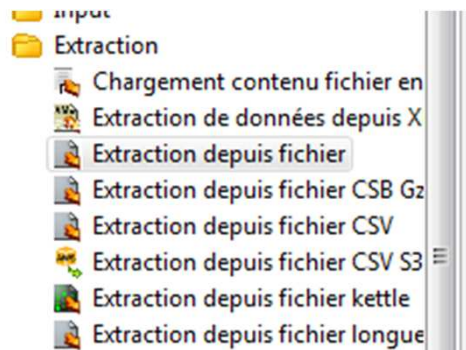
Créer une transformation à l'aide de SPOON

- Créez un fichier texte dans un dossier nommé D:/BI/atelier1/ nommé source1.txt Contenant une liste de noms(tous les fichiers de cet atelier doivent être enregistrés dans le même dossier atelier1) exemple:
 - Lancez SPOON à partir du dossier Pentaho\data-integration
 - Fichier /Nouveau /Transformation, enregistrer la transformation nom de la transformation: Atelier1.ktr

```
Nom  
Bill Gates  
Steve Jobs  
Steve Balmer  
Daniel Ritchie  
Noam CHomsky
```

Ajout de l'étape « Extraction depuis fichier »

- Dans le volet « Palette de création », glissez l'étape « Extraction depuis fichier », vers la transformation atelier1



- Double cliquez sur l'icone « Extraction depuis fichier » pour définir ses propriétés
 - Fichier ou répertoire: Fichier source1.txt, sélectionnez le fichier et cliquez sur « ajouter »
 - Effacer le contenu des champs « Delimited » et « Entouré par », dans l'onglet contenu.
 - Type fichier: Fixed.
 - Dans l'onglet « Champs », cliquez sur le bouton « Obtenir les champs »

Etape « Ajout Constantes »

- Dans la catégorie « Transformation » glissez l'étape « Ajout constantes »
- Définir les deux constantes suivantes:
 - Nom: message; Type: String; Value: Bonjour
 - Nom: exclamation; Type: String; Value:!



Extraction depuis fichier



Ajout constantes

Ajouter des constantes

Nom étape: Ajout constantes

Champs :

#	Nom	Type	Format	Longueur	Precision	Devise	Décimal	Groupe	Valeur
1	message	String							Bonjour
2	!	String							!

Ajout d'un lien de « Extraction depuis fichier » vers « Ajout constantes »

- Gardez « MAJ » (ou la roulette de la souris) Appuyée et glissez « Extraction depuis fichier » vers « Ajout constantes »
- Une autre possibilité est d'ajouter un lien depuis le nœud « liens » dans le volet « navigation »



Ajout d'une étape « Alimentation fichier »

- Ajoutez une étape « Alimentation fichier » à partir de la catégorie « Alimentation ».
- Ajoutez un lien « Ajout constantes » vers « Alimentation fichier ».
- Définir les propriétés de l'étape « Alimentation fichier »:
 - Nom: sortie
 - Extension : txt
- Dans l'onglet « Contenu » effacer le « Entouré par ».
 - champs « Délimiteur »: (espace)
- Dans le volet « Champs », cliquez sur « Récupérer champs ».
- Cliquez sur « Largeur minimale » pour effacer les espaces.
- Changez l'ordre des champs comme suit:

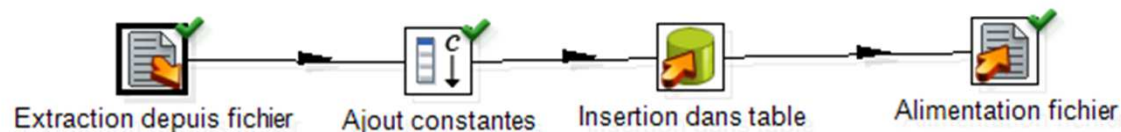
#	Nom	Type
1	message	String
2	Nom	String
3	!	String

Créer des connexions

- .Dans le vote « Connexions », le nœud permet de créer des connexions à des bases de données.

Exemple: Création d'une connexion SqlLite

- enregistrez la transformation précédente sous « atelier2.ktr ».
- Créez une connexion avec les paramètres suivants:
 - Nom de la connexion: SQLite
 - Type de base de données: SQLite.
 - Type d'accès base de données: JDBC.
 - Nom du serveur: atelier2/Sqlite.sqlite
 - Port : -1
- Glissez l'étape « Alimentation dans table » sur le lien entre « Ajout constantes » et « Alimentation fichier »



- Propriétés de l'étape « Insertion dans table »:
 - Table cible: table1
 - Cliquez sur le bouton SQL, puis sur le bouton « Exécuter » dans la nouvelle fenêtre qui s'ouvre pour créer la table table1.

Exemple 2

- Objectifs: Chargement des données sur les ventes dans une table à partir d'un fichier texte csv avec traitement des données manquantes:
- Fichier source: C:\pentaho\data-integration\samples\transformations\files\sales_data.csv.

Nom étape

[d'erreur](#) [Filtres](#) [Champs](#) [Champs additionnels](#)

Type fichier

Délimiteur

Entouré par

Autoriser arrêt entre champs avec quote

Échappement

En-tête Nombre de lignes d'en-tête

Fin de fichier Nombre de lignes de fin de fichier

Lignes césurées Nombre de lignes césurées

Document paginé Nombre de lignes par page

Nombre de lignes d'en-tête

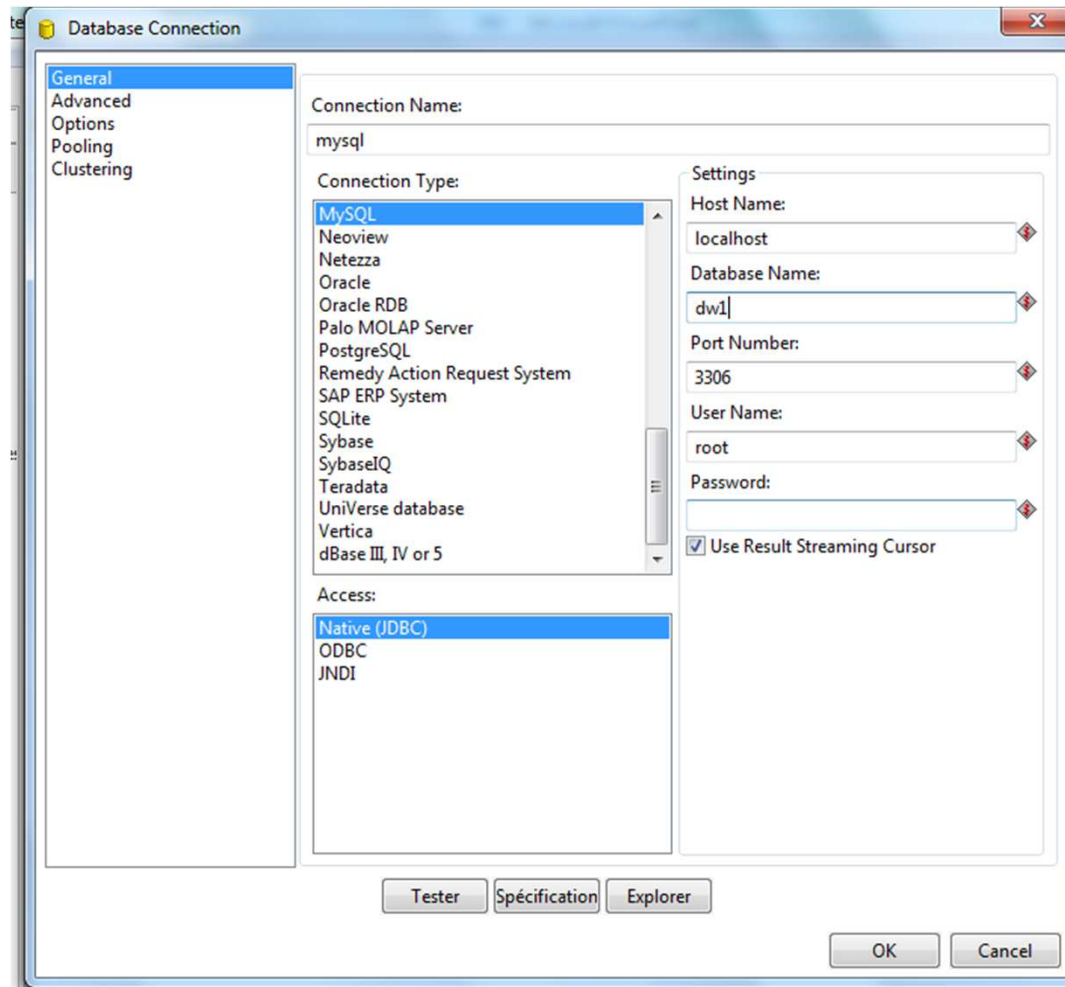
Compression

Connexion HSQLDB (Hypersonic SQLDB)

The image shows a configuration window for a database connection. It is divided into several sections:

- Connection Name:** A text field containing "Sampledata".
- Connection Type:** A list box with "Hypersonic" selected. Other options include IBM DB2, Infobright, Informix, Ingres, Ingres VectorWise, Intersystems Cache, KingbaseES, LucidDB, MS Access, MS SQL Server, MS SQL Server (Native), MaxDB (SAP DB), MonetDB, MySQL, and Neoview.
- Access:** A list box with "Native (JDBC)" selected. Other options are ODBC and JNDI.
- Settings:** A group of text fields for configuration:
 - Host Name:** localhost
 - Database Name:** sampledata
 - Port Number:** 9001
 - User Name:** sa
 - Password:** (empty field)

Connexion Mysql



- Le pilote Mysql doit être copié dans le dossier pentaho\data-integration\libext\JDBC

Étape Recherche de code postal

Nom étape Code postal manquant

Envoyer les données 'VRAI' à l'étape

Envoyer les données 'FAUX' à l'étape

Condition (VRAI)

= <field>

<value>

Champs

Sélectionnez un champ

- ORDERNUMBER (Integer)
- QUANTITYORDERED (Integer)
- PRICEEACH (Number)
- ORDERLINENUMBER (Integer)
- SALES (Number)
- ORDERDATE (Integer)
- STATUS (String)
- QTR_ID (Integer)
- MONTH_ID (Integer)
- YEAR_ID (Integer)
- PRODUCTLINE (String)
- MSRP (Integer)
- PRODUCTCODE (String)
- CUSTOMERNAME (String)
- PHONE (String)
- ADDRESSLINE1 (String)
- ADDRESSLINE2 (String)
- CITY (String)
- STATE (String)
- POSTALCODE**
- COUNTRY
- TERRITORY
- CONTACTLASTNAME
- CONTACTFIRSTNAME

Condition (VRAI)

POSTALCODE IS NOT NULL

Etape: Recherche dans flux

Recherche valeurs dans flux

Nom étape Recherche code postaux manquants

Etape source données Codes Postaux

Champs de recherche (clé)

#	Champ (flux principal)	Champ (étape source)
1	CITY	CITY
2	STATE	STATE
3		

Indiquer les champs à retourner depuis l'étape source

#	Champ (étape source)	Nouveau nom	Défaut	Type
1	POSTALCODE	CP		String

Préserver mémoire

Clé et valeur sont des entiers

Utiliser liste triée (table hash)

OK Annuler Récupérer champs (clé) Récupérer champs (étape source)