Chapitre 4
Présentation de l'ETL PDI

1. Principes de fonctionnement des ETL (Extract-Transform-Load)

1.1 Définition générale

Dans un système d'information d'entreprise, les données sont stockées dans de multiples formats :

- systèmes de gestion de base de données relationnels (**SGBD-R**),
- fichiers plats (**CSV**...),
- flux **XML**,
- tableurs,
- ...

et ne sont pas directement utilisables pour une exploitation en Informatique Décisionnelle.

Les raisons sont multiples :

- Non-disponibilité de l'information : les données hébergées par les SGBD-R sont gérées de manière transactionnelle. Cela signifie que des ajouts, modifications, consultations, suppressions, mises à jour des enregistrements ont lieu de manière continue et également concurrente (accès aux données depuis des postes de saisie multiples). Il n'est donc pas satisfaisant d'intervenir directement sur ces données avec des outils relevant de l'Informatique Décisionnelle.

- Présentation inadéquate des données : dans un système dit de production, les données sont conservées et actualisées sur une période de référence, éventuellement entreposées ensuite sous forme d'historique mais rarement agrégées. Dans un stockage orienté décisionnel, les contraintes sont différentes. À titre d'exemple pour une analyse de chiffre d'affaires sur des familles de produits, il n'est pas nécessaire de conserver toutes les propriétés caractérisant les produits eux-mêmes (le code produit, la désignation et le code d'appartenance à une famille pourraient suffire). Par contre conserver les informations sur les ventes sur plusieurs années peut avoir du sens si une analyse comparative dans la durée est envisagée.
- Lisibilité des données : les données de base stockées dans des SGBD-R ne sont pas toujours facilement interprétables par les analystes décisionnels. À titre d'exemple, les noms des tables, des champs sont très souvent peu significatifs. Il ne faut pas perdre à l'esprit que les grandes fonctions de l'entreprise ou de l'organisation au sens large sont de plus en plus gérées par des logiciels de type **ERP** (traduction approximative en **Progiciel de Gestion Intégré** en français). Les appellations des tables et des champs doivent parfois être modifiées pour faciliter l'interprétation des documents générés en Informatique Décisionnelle (listes, graphes, analyses, tableaux de bord...). L'ETL joue souvent ce rôle. On peut aussi utiliser un référentiel métier pour stocker ces correspondances.

Historiquement les retraitements, désormais assurés par les logiciels ETL, étaient réalisés par l'intermédiaire de programmes informatiques classiques (développés dans des langages de programmation divers) capables d'accéder en lecture/écriture aux SGBG-R. Les inconvénients de cette démarche de développement étaient multiples :

- Nécessité de disposer de compétences en développement informatique (compétences techniques pas toujours acquises par les intervenants dans le monde du décisionnel).
- Maintenance difficile de ces mini-programmes, parfois développés dans des langages de programmation hétérogènes.
- Contrôle du séquençage des retraitements pas toujours aisé à maîtriser.
- ...

1.2 Les phases Extract-Transform-Load

ETL sont les initiales de Extract-Transform-Load. Concrètement ce type d'outil assure trois grandes fonctions :

- **Extraction** : sélection des données utiles dans les tables gérées par des SGBD-R, intégration de flux XML, de fichiers plats...
- **Transform** : retraitement des données extraites pour qu'elles soient utilisables dans un processus décisionnel.
- **Load** : les outils de restitution en Informatique Décisionnelle (générateurs de rapports, grapheurs, tableaux de bord...) ne sont pas directement alimentés en données par les ETL. Les ETL entreposent les données transformées dans des entrepôts de données (datawarehouses ou datamarts) eux-mêmes souvent hébergés par des SGBD-R.

Il est rare que les données de base soient reprises en l'état. Les transformations jouent donc un rôle essentiel dans la séquence ETL.

Les transformations (Transform), modélisées dans les ETL par l'intermédiaire d'une interface graphique très conviviale, constituent la majorité des traitements.

Pour la phase Extract, les composants utilisés permettent essentiellement de :

- se connecter aux sources de données via des **connecteurs** (pilotes, librairies d'accès aux données...),
- constituer les jeux de données à prendre en compte (**DataSet**) par l'intermédiaire de requêtes SQL ou d'un requêteur graphique (**Query Builder**).

En ce qui concerne le Transform, les composants de transformation sont extrêmement nombreux. Sans vouloir être ici exhaustif, ils permettent en particulier les transformations suivantes :

- Sélection de certaines colonnes uniquement pour retraitement (Transform) et report en sortie (Load) alors qu'en entrée (Extract) le jeu de données pouvait être plus conséquent.
- Sélection de données provenant de multiples sources (SGBD-R...).
- Non-prise en compte de valeurs nulles ou erratiques (ne pas prendre en compte les valeurs de chiffres d'affaires négatives par exemple ou encore les ventes pour lesquelles le client n'est pas correctement renseigné).

- Transformation ou harmonisation de certaines codifications (la codification des produits en entrée pourrait par exemple être différente dans l'univers des ventes et celui de la production).
- Mise place de champs calculés (Montant ligne de facture=Prix de Vente Unitaire du produit * Quantité facturée) qui évite ces mêmes calculs dans les outils de restitution.
- Filtrage des données pour réduire le périmètre d'analyse (prise en compte uniquement des ventes des trois dernières années).
- Tri des données, ce qui évite aussi de refaire cette opération dans les outils de restitution.
- Agrégation de données pour réduire la taille du datawarehouse (cumul du chiffre d'affaires sur une période comme le mois si une analyse plus fine n'est pas envisagée).
- Recherche d'un libellé manquant dans une table satellite pour éviter de multiplier les tables dans le datawarehouse.
- Concaténation ou au contraire éclatement de champs.
- ...

Pour la phase Load, les composants utilisés sont assez semblables à ceux vus au niveau de la phase Extract. Ils gèrent le report dans le datawarehouse ou les datamarts des données retraitées.

2. Installation de Pentaho Data Integration (PDI)

2.1 Téléchargement

Dans le chapitre "Prise en main rapide de Pentaho", les composants logiciels suivants ont été installés :
- le serveur Pentaho Business Intelligence Server,
- le composant Pentaho User Console qui est l'interface client du serveur Web de publication des résultats,
- le composant Pentaho Administration Console qui est l'interface d'administration de Pentaho.

L'ETL **Pentaho Data Integration** (**PDI**) fait aussi partie de la suite Pentaho mais doit être installé séparément.

Pour des raisons de disponibilité de connecteurs pour certains types de bases de données (en particulier **JDBC-ODBC**), le choix a été fait ici d'utiliser l'avant-dernière version de Pentaho Data Integration, c'est-à-dire la version 3.2.0.

La transposition des transformations développées dans le cadre du chapitre "Mise en œuvre de PDI" ne devrait poser aucune difficulté (compatibilité ascendante) dans la récente version 4.0.0. D'autre part dans cette nouvelle version, l'interface utilisateur n'a été modifiée que très peu.

PDI est disponible en téléchargement à l'adresse
http://sourceforge.net/projects/pentaho/files/Data%20Integration/.

2.2 Installation

Il est conseillé de sélectionner la version **3.2.0-stable** dans le format correspondant à votre système d'exploitation (Windows ou Linux) et de l'entreposer dans votre répertoire habituel pour ce type de téléchargements (C:\Pentaho_CE_download par exemple pour un utilisateur Windows).

Le logiciel téléchargé est ensuite décompressé dans un répertoire comme C:\Pentaho_CE\PDI_CE-3.2.0 par exemple.

2.3 Lancement de Pentaho Data Integration

Le démarrage de Pentaho Data Integration s'obtient sous Windows par un double clic sur le fichier exécutable **Kettle.exe** se trouvant dans le répertoire C:\Pentaho_CE\PDI_CE-3.2.0.

Pour Linux un fichier de commandes **spoon.sh** est aussi prévu.

3. Présentation générale de l'ETL PDI

3.1 Fonctionnalités principales de PDI

Pentaho Data Integration, antérieurement connu sous l'appellation **Kettle**, est un ETL Open Source qui permet de concevoir et exécuter des opérations de manipulation et de transformation de données très complexes.

PDI, comme la majorité des ETL d'ailleurs, permet une modélisation graphique des opérations sur les données que ce soit en matière de récupération (**Extract**), de retraitement (**Transform**) ou de stockage en sortie (**Load**) essentiellement dans des datewarehouses ou des datamarts.

La spécificité de PDI est que l'utilisateur n'est pas contraint à utiliser des séquences de programmation dans la définition des étapes de son traitement **ETL**. Toutes les étapes du traitement, qui sont positionnées et séquencées sur un flux, sont élaborées au travers d'un assistant qui évite à l'utilisateur d'avoir recours à du code programmé. Seules des connaissances en SQL peuvent être nécessaires. Pour des traitements avancés, on peut avoir recours à des scripts rédigés en JavaScript.

Kettle est devenu Open Source à partir de la version 2.2 et depuis son entrée dans le projet de plate-forme décisionnelle Pentaho il bénéficie d'une parfaite intégration.

Pentaho Data Integration offre la possibilité de créer deux types de processus :

- Les **transformations** : ces traitements (ensemble de tâches intégrées à un flux) permettent d'extraire les données d'une ou plusieurs sources de données (SGBD-R, fichiers plats, flux XML...), de les traiter et ensuite de les stocker dans des systèmes de stockage cible identiques à ceux en entrée mais aussi dans des datawarehouses, des datamarts ou encore des cubes multidimensionnels.
- Les **tâches** : il s'agit tout simplement de transformations auxquelles sont rajoutées des actions complémentaires plus sophistiquées comme l'envoi automatique de mail, la publication de données sur un serveur FTP, le traitement conditionnel...