Entrepôt de données : de l'alimentation des données au décisionnel de l'établissement

Matthieu Leblanc

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

PARIS

Mémoire présenté en vue d'obtenir

Le diplôme d'ingénieur

Spécialité : INFORMATIQUE

Par

LEBLANC Matthieu

Entrepôt de données : De l'alimentation des données au décisionnel de l'établissement

Soutenu le 13/03/2012

JURY

PRESIDENT : M. Yann Pollet

MEMBRES : M. Marc-Antoine Carlet, M. Jean-Michel Douin, M. Emmanuel Gallis et M. Pascal Graffion

Remerciements

Je tiens à remercier toutes les personnes qui ont donné de leur temps, talent et expérience tout au long de ce projet et durant mes cinq années au sein de la DSI du C.N.A.M de Paris.

Je souhaite tout particulièrement remercier Monsieur Jean-Michel Lery, chef de projet technique du projet SISCOL, pour ses remarques pertinentes et ses précieux conseils durant toutes les phases du projet ainsi que lors de la rédaction de ce mémoire. Je remercie aussi messieurs Denis Corée, coordinateur du projet et responsable du service informatique, et Christophe Dumoulin qui m'ont apporté leur soutien ainsi que leur expertise tout au long du projet.

Je tiens également à remercier l'établissement du CNAM qui a su me faire confiance et me laisser mener une tâche aussi importante que sensible. Je remercie tout particulièrement Messieurs Jean-Michel Douin et Pascal Graffion pour leur conseil lors du choix de mon sujet de mémoire.

Bien entendu, je n'aurais probablement pas réalisé tout cela sans le soutien de mes proches, parents et amis.

Glossaire

Agréger : action de réunir des éléments distincts en un tout.

Base de données relationnelle : base de données structurée suivant les principes de l'algèbre relationnelle.

Correspondance : rapport logique défini par l'utilisateur entre une ou plusieurs colonnes des bases sources et une colonne de la base cible d'un processus ETL.

Cube : structure matricielle à trois dimensions.

Datawarehouse : entrepôt de données. Concept de stockage de données.

Datamart : magasin de données : C'est un sous ensemble de l'entrepôt de données.

Métadonnées (d'une base de données) : propriétés d'une base de données (liste des tables, des index, des clés, …).

Schéma global : schéma de la base cible dans une approche matérialisée ; schéma du médiateur dans une approche virtualisée.

Schéma local : schéma d'une base source.

Snippet : petite région réutilisable de code source.

Transformation : opération résultant du traitement d'une correspondance dans Talend.

Infotype : objet porteur d'information dans SAP.

Abréviations

ABAP : Advanced Business Application Programming. Langage de programmation propriétaire SAP.

BI : « Business intelligence ». Ensemble de données consolidées qui permet la prise de décision.

BO : « Business Objects ». Solution de la société SAP permettant la construction de requêtes et de rapports d'analyse ou tableaux de bord.

CEP : Centre d'Enseignement de Paris.

CSV : « Comma Separated Values ». Valeurs séparées par des virgules.

DSI : Direction des systèmes d'information. Elle régit l'intégralité du parc informatique, du réseau et de l'information.

ERP : « Enterprise Resource Planning ». Progiciel de gestion intégré (PGI).

ETL : « Extract Transfort Load ». Processus ayant pour but de récupérer les données des bases de production pour les injecter dans le datawarehouse après avoir effectué des transformations.

ODS : « Operational Data Store ». Zone de préparation des données.

OLAP : « Online Analytical Processing ». Traitement analytique en ligne.

SGBD : Système de Gestion de Base de Données.

SID : Systèmes d'information décisionnels

SIO : Systèmes d'information opérationnels

UML : « Unified Modeling Language ». Langage de modélisation unifié.

Plan du mémoire

Ce mémoire est divisé en cinq chapitres.

La première partie est une présentation de l'établissement du Conservatoire National des Arts et Métiers ainsi que du projet de mise en production d'un nouveau progiciel de scolarité, SAP, auquel j'ai participé. Cette première section est essentielle car elle identifie le cadre fonctionnel, parfois spécifique au CNAM, dans lequel doit s'insérer ce projet de gestion unifiée de la scolarité parisienne. Elle présente également le rôle spécifique que joue la direction des systèmes d'information (DSI) dans la vie de l'établissement. Elle présente également certains aspects et enjeux liés à la mise en place d'un entrepôt de données dans la réussite du projet. Enfin elle introduit l'importance de la reprise et le traitement des données dans la réussite du projet.

La seconde partie présente certains des concepts sur lesquels j'ai pu m'appuyer lors des phases d'étude et de réalisation. En effet n'ayant qu'une expérience théorique dans le domaine de l'informatique décisionnelle et le traitement des flux de données, je me suis beaucoup documenté afin d'approfondir mes connaissances des concepts généraux aux entrepôts de données ainsi que sur les outils d'extraction de données. Vous y trouverez également une introduction au fonctionnement de l'outil TALEND avec lequel j'ai réalisé l'ensemble des tâches liées à la gestion des flux de données.

La troisième partie décrit les différentes études réalisées lors de l'analyse des besoins ainsi que certaines des tâches dont j'étais en charge durant le projet. Par ailleurs elle synthétise certaines des problématiques récurrentes rencontrées lors de la mise en qualité des données ainsi que les contraintes spécifiques liées à leur intégration dans l'entrepôt de données.

La quatrième partie présente les différentes réalisations liées à la reprise de données de production qui fut ma principale tâche durant le projet. Le périmètre des données concernées étant important, je me suis efforcé de choisir des exemples représentatifs des tâches effectuées. J'espère donner une vision d'ensemble de mon travail tout en cherchant à être le plus exhaustif possible.

Puis j'ai consacré un chapitre à la présentation des tâches d'interfaçage avec le nouveau progiciel de scolarité SAP.

Enfin le dernier chapitre propose un bilan du travail effectué autour de l'entrepôt de données notamment la valeur ajouté lors de l'alimentation de la nouvelle application de scolarité ainsi que les perspectives envisageables en termes de pilotage de l'établissement. Je finirai par une synthèse sur les enrichissements personnels que m'a apportés ma participation au projet.

Table des matières

Table des matières 7

I. Contexte du projet 11

1. Présentation générale du CNAM 11

1) Un enseignement supérieur ouvert à tous 11
2) Missions et activités 11
3) La formation tout au long de la vie 11
4) La recherche technologique et l'innovation 11
5) La promotion de la culture scientifique et technique 12
6) Organisation actuelle 12

2. Présentation de la DSI du CNAM 13

1) Les missions de la DSI 14
2) L'équipe application 14
3) Mon profil 15

3. Introduction au projet 16

4. Description Fonctionnel du projet de scolarité SISCOL 17

1) Environnement fonctionnel du projet de scolarité 17
2) Etat actuel du système de gestion de la scolarité parisienne 18
3) Le nouveau progiciel de scolarité 21

5. Les objectifs de l'entrepôt de données 22

1) Enjeux de l'entrepôt de données pour le CNAM 22
2) Objectifs principaux 22

6. Objectifs de la reprise de données 24

1) Enjeux de la reprise de données 24
2) Objectifs principaux 25

7. Synthèse 26

II. Introduction aux entrepôts de données 27

1. L'entrepôt de données....28
1) Définition....28
2) Définition générales....29
3) Caractéristiques....29
2. Les datamarts....32
1) Modélisation d'un magasin de données....33
2) Les faits....33
3) Les dimensions....33
3. Modélisation logique des données....34
1) Modélisation en étoile....35
2) Modélisation en flocon de neige....36
3) Modélisation en constellation....36
4) Comparaison des modèles en étoile et en flocon....37
4. Alimentation de l'entrepôt et ETL....38
1) Introduction....38
2) L'Intégration ou alimentation de données....38
3) Extraction des données....39
4) Le chargement et le transfert des données....39
5) Comparatif des outils actuels....40
5. L'outil Talend....42
1) Fonctionnement....43
2) Le référentiel....44
3) La palette de composants....47
4) Le job designer....48
5) La création de composant....53
6) Génération de code....54
6. Synthèse....55
III. Etude de la reprise de données....56
1. Business model de la reprise de données des applications de scolarité....57
1) Alimentation des référentiels....58

2) Extraction des données 58
3) Intégration de ces données dans l'ODS 58
4) Mise en qualité des données dans l'ODS 58
5) Extraction des données mise en qualité dans l'ODS dans des fichiers destinés à être intégrés dans l'entrepôt de données 58
6) Extraction des données de rejets dans des fichiers 59
7) Intégration des données dans l'entrepôt 59
8) Extraction des données de l'entrepôt dans des fichiers textes en direction de SAP 59
2. Etapes de la reprise de données 60
1) Accéder aux données. 61
2) Interpréter les données 62
3) Standardiser des sources de données hétérogènes 62
4) Valider des données 64
5) Gérer les rejets 64
6) Lier les données 65
7) Consolider les données 65
8) Intégrer les données dans l'entrepôt de données 65
3. Problématiques liées à la réconciliation de données 66
1) Enjeux 66
2) Problématiques 66
3) Détection, correction et nettoyage des données 67
4) Résolution des problèmes spécifiques 70
4. Synthèse 73
IV. Reprise de données production 74
1. Reprise des dossiers administratifs du CEP et de l'INTEC 75
1) Extraction des sources du CEP 75
2) Mise en qualité des critères d'unicité 76
3) Reprise des dossiers administratifs de l'INTEC 79
2. Reprise des dossiers pédagogiques de l'INTEC 83
3. Reprise EICNAM 86

1) Modalités de reprise....86
2) Réalisation....88
4. Gestion des rejets....91
1) Graphique général....91
2) Description de la structure des tables....92
3) Alimentation de ces champs....92
4) Exemples....93
5) Déroulement....94
6. Intégration des données élèves dans l'entrepôt....95
7. Synthèse....97
V. Interfaçage avec SAP....98
1. La migration de données vers le système SAP....98
1) Motivations....98
2) Processus de migration de Données....98
3) Les Challenges de la Migration de Données....99
2. Interfaces entrantes SAP....100
1) Objectifs....100
2) Dossier administratif....101
3) Recette des fichiers générés....104
3. Interfaces sortantes de SAP....105
1) Objectif....105
2) Dictionnaires de données....106
3) Besoin fonctionnels....106
4) Protocole de chargement....109
5) Réalisation....111
4. Synthèse....113
VI. Conclusion....114

I. Contexte du projet

1. Présentation générale du CNAM

Le Conservatoire National des Arts et Métiers est un grand établissement public d'enseignement supérieur et de recherche sous tutelle du Ministère de l'enseignement supérieur et de la recherche.

1) Un enseignement supérieur ouvert à tous

Le CNAM est ouvert à tous, sans exigence préalable de diplômes. Il s'adresse à tous les adultes engagés dans la vie active, quel que soit leur secteur d'activité. Les enseignements du CNAM permettent à tous ceux qui le souhaitent d'améliorer leurs compétences professionnelles, d'acquérir une formation complémentaire, d'anticiper l'évolution de leur métier, d'obtenir un diplôme, etc.

2) Missions et activités

Les trois missions du CNAM sont : la formation tout au long de la vie, la recherche technologique et la promotion de la culture scientifique.

3) La formation tout au long de la vie

Le CNAM répond aux besoins de formation des adultes, en leur permettant d'enrichir, tout au long de leur parcours professionnel, leurs compétences, et ce, en tous lieux du territoire. L'enseignement est conçu pour tous ceux qui travaillent, sans discrimination. Il doit être accessible sur tout le territoire, le soir ou en journée, en présentiel, en alternance ou à distance.

4) La recherche technologique et l'innovation

Brevets, essais, innovation, transfert de technologies, incubation d'entreprises, pôles de compétitivité... la recherche au CNAM s'appuie sur une activité pluridisciplinaire et l'engagement des entreprises : une École doctorale arts et métiers, 8 écoles doctorales partenaires, 24 équipes de recherche reconnues, 33 doctorats habilités, 330 doctorants accueillis, 188 thèses soutenues, 923 mémoires d'ingénieur.

5) La promotion de la culture scientifique et technique

Le CNAM, avec son Musée des arts et métiers et son réseau documentaire et numérique, est un acteur majeur de diffusion de la culture scientifique et technique (350 événements et conférences ouverts à tous, 50 000 participants dans toute la France, 200 000 visiteurs du Musée des arts et métiers, 534 500 consultations de la bibliothèque en ligne et du Conservatoire numérique).

6) Organisation actuelle

L'organisation générale du CNAM est représentée par le schéma suivant :

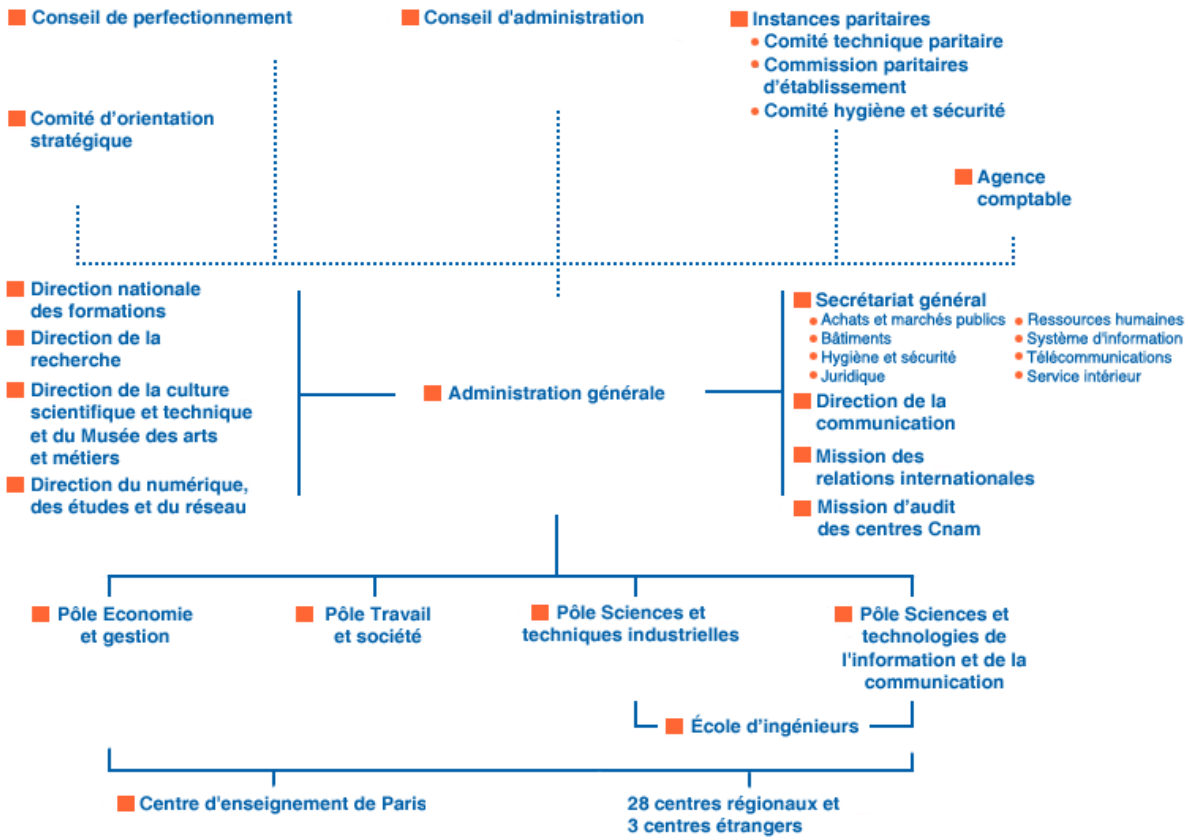


Figure 1 : Organisation générale du CNAM de Paris

Les principaux intervenants dans la mission d'enseignement sont : les instances, la direction nationale des formations (DNF), le centre d'enseignement parisien (CEP), les pôles, les instituts (INTEC, etc.), l'école d'ingénieurs, les centres régionaux associés (CRA).

2. Présentation de la DSI du CNAM

Je travaille à la direction des systèmes d'information du CNAM de Paris depuis le 1er mai 2005. La direction des systèmes d'information du CNAM emploie environ 40 personnes. Sous la responsabilité du chef de service M. Denis Corée, le service est en charge d'une part de l'installation, de la maintenance et de l'évolution de l'ensemble des composants matériels informatiques et d'autre part de l'achat, du développement et de la mise à jour de l'ensemble des logiciels. Toutes ces activités sont partagées entre plusieurs équipes aux métiers et aux profils différents (Figure 2).

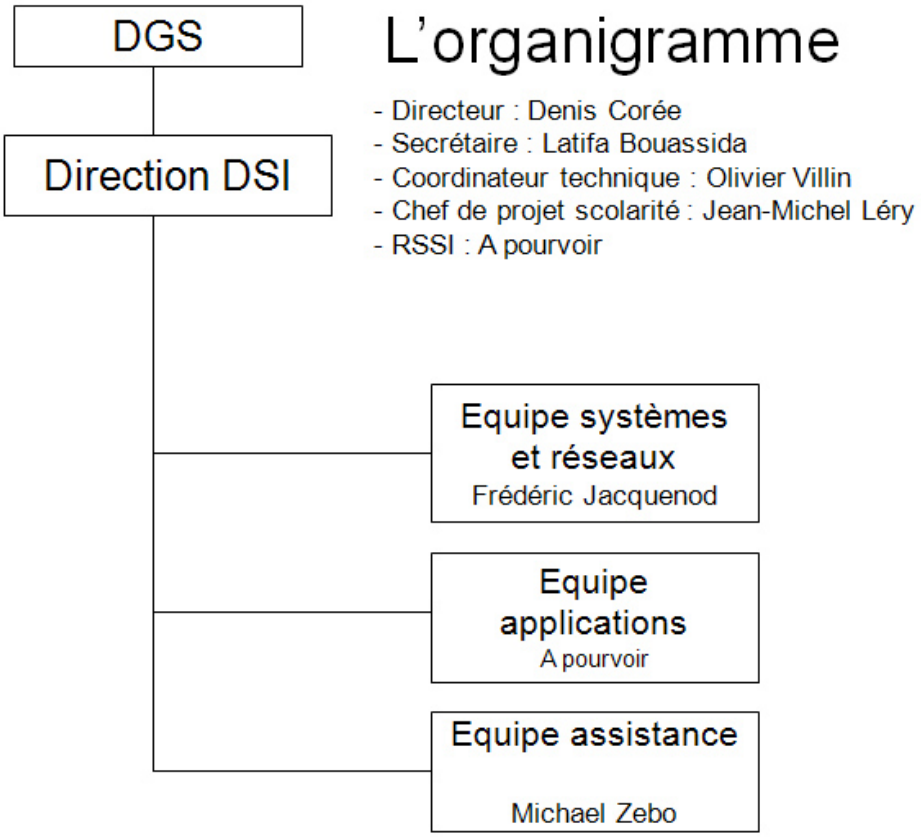


Figure 2 : Organigramme de la DSI du CNAM

1) Les missions de la DSI

Les principales missions du service sont :

- Mettre en œuvre les systèmes informatiques et les applications de l'Etablissement et en garantir les performances, la sécurité et la disponibilité ;
- Faire évoluer de manière cohérente le système d'information de l'Etablissement ;
- Assurer une assistance aux utilisateurs ;
- Produire des tableaux de bord sur l'activité de la DSI.

2) L'équipe application

L'équipe application, composée de 8 personnes, a pour mission de concevoir et le développer des solutions informatiques répondant aux besoins des utilisateurs. Afin d'y répondre au mieux, les membres du service doivent analyser les besoins, imaginer des solutions, les modéliser et les implémenter vers un système et une plateforme cible. Il s'agit de métiers alliant capacité d'analyse, esprit de synthèse, mettant en œuvre technique et créativité. Ces métiers sont surtout portés vers l'informatique de gestion, caractérisés par l'utilisation de l'outil informatique pour simplifier la gestion administrative de l'établissement (suivi des auditeurs, gestion du personnel et des salaires, comptabilité et facturation des fournisseurs). L'informatique de gestion est étroitement liée au système d'information de l'établissement et prend la forme, depuis l'implantation de SAP, d'un progiciel de gestion intégré également envisagé pour le prochain logiciel de scolarité.

Les principales missions de l'équipe étude et développement sont :

- Développer, maintenir et faire évoluer les applications de gestion et les applications dédiées à la pédagogie ;
- Assister la maîtrise d'ouvrage pour la définition de ses besoins et des solutions à mettre en œuvre ;
- Tenir à jour un inventaire des applications et de leurs interfaces ;
- Produire des tableaux de bord sur le fonctionnement des applications ;
- Produire la documentation des programmes.

3) Mon profil

En tant que développeur (également nommé analyste-programmeur) je conçois et développe des applications informatiques, c'est-à-dire je transcris un besoin en une solution informatique indépendante d'un langage cible puis l'implémente sur une plateforme spécifique. Ce processus est généralement découpé en plusieurs phases. Habituellement, le développement informatique est assuré par un analyste-programmeur alors que le recueil et l'expression des besoins est généralement assuré par le chef de projet. L'analyste est chargé de la modélisation de l'application et du choix de la plateforme cible. Il traduit fonctionnellement le besoin d'un client et propose une modélisation informatique. Le programmeur est chargé du codage, de la configuration des outils et du déploiement sur une plateforme spécifique. J'ai, la plupart du temps, assuré ces phases moi-même. Mon métier requiert des compétences techniques comme la programmation objet ou la modélisation avec UML (langage de modélisation unifié) mais aussi des capacités rédactionnelles ainsi que le sens de l'écoute et de la synthèse. Pour m'adapter aux différents types de projet, J'ai dû maîtriser au cours du temps plusieurs langages de programmations tels que le JAVA (langage compilé puis exécuter par une machine virtuelle) et le PHP (langage de script interprété) et donc plusieurs environnements de déploiement comme JBOSS (serveur d'applications implémentant les services J2EE) ou Apache (serveur WEB). Il est aussi indispensable de connaître le SQL pour pouvoir interroger une base de données et le XML pour pouvoir échanger et valider des données structurées. Par ailleurs, avec l'explosion des applications web J'ai appris à utiliser, le HTML, les CSS ainsi que le JavaScript. Enfin, la maîtrise de l'anglais est indispensable dans la mesure où le développeur est amené à se documenter sur des sujets pointus généralement rédigés en anglais et peut parfois être en relation avec des correspondants étrangers.

3. Introduction au projet

La Direction des Systèmes d'Information du Conservatoire national des arts et métiers a en charge la gestion de la scolarité pour le Centre Parisien ainsi que celle de certains de ses instituts. Cette gestion de la scolarité était répartie sur une dizaine d'applications et sur plusieurs bases de données hétérogènes. Le CNAM est actuellement en train de refondre son système d'information dédié à la scolarité sur son site parisien et a retenu pour ce faire le progiciel de scolarité SLM de SAP qui doit intégrer ou s'interfacer avec les diverses applications existantes. Afin de réduire l'effort porté sur le développement des multiples interfaces et d'améliorer la qualité des données reprises, le choix a été retenu d'interfacer le progiciel de scolarité avec un entrepôt de données.

Depuis novembre 2009, ma mission principale fut de mettre en place et surtout d'alimenter un entrepôt de données permettant d'une part d'effectuer la reprise des données fonctionnelles de scolarité de l'établissement et d'autre part de servir de référentiel afin d'être exploité dans la prise de décision et le pilotage de l'établissement. Le but étant d'offrir une vision homogène et transverse.

Aujourd'hui, cet entrepôt de données a permis d'alimenter le nouveau logiciel de scolarité mais aussi de collecter, d'ordonner, d'historiser et de stocker des informations provenant des différentes bases de données opérationnelles. L'alimentation et la gestion des données présentes dans l'entrepôt se font à l'aide de l'ETL (Extract Transform Load) Talend présenté en paragraphe 2.5. Par ailleurs, j'ai pu travailler sur des outils de restitutions des données permettant aux utilisateurs d'exploiter les données présentes dans l'entrepôt.

4. Description Fonctionnel du projet de scolarité SISCOL

1) Environnement fonctionnel du projet de scolarité

Le système d'information du CNAM se décompose en 4 axes principaux (Figure 3) :

- la gestion de la paye et des ressources humaines (10%)
- la gestion du patrimoine (10%)
- la gestion financière et comptable (30%)
- la gestion de la scolarité (60%).

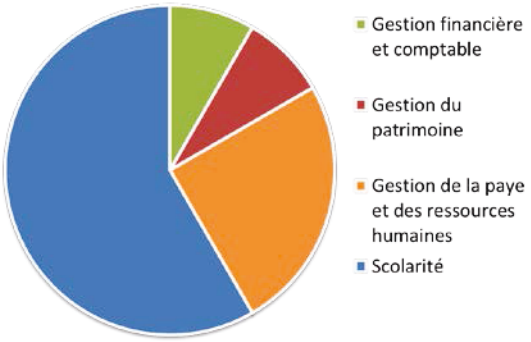


Figure 3 : Périmètres du S.I

Le système de gestion de la scolarité est un élément du système d'information global du CNAM. Il doit donc pouvoir s'interfacer (Figure 4) avec les autres logiciels du système d'information et en particulier avec :

- la gestion de l'offre de formation nationale à partir de la base BDO de la DNF ;
- le logiciel de délivrance des diplômes ;
- l'annuaire LDAP du CNAM;
- le logiciel financier et comptable SIFAC ;
- le logiciel de planification des salles Hyper Planning ;
- le logiciel de paiement des enseignants HR Access;
- le logiciel de gestion du personnel Virtualia.

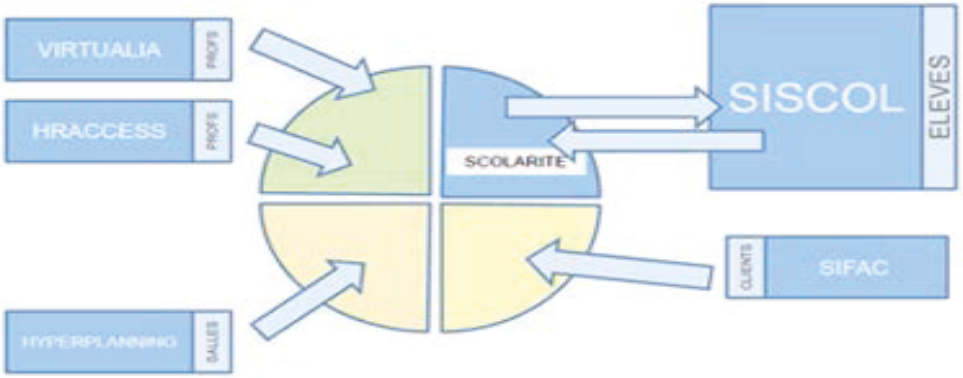


Figure 4 : Interactions entre les différents logiciels du SI

2) Etat actuel du système de gestion de la scolarité parisienne

La gestion de la scolarité du CEP, du CACEMI et de l'INTEC est basée sur plusieurs applications distinctes et complémentaires telles que : Grafic, BDCEP, UTINTEC, Hyperplanning, GAGE UE, GAGE Diplômes, ANGE, VES et Viatic2, EICNAM.

Grafic est une application Intranet sécurisée gérée par la DSI, développée par la société Synergie3r. Elle est utilisée pour la gestion du Centre d'Enseignement de Paris (CEP), de la Formation Continue (FC) et du Centre d'Actualisation des Connaissances et de l'Étude des Matériaux Industriels (CACEMI).

Les principales fonctionnalités sont les suivantes :

- Gérer les sessions de formation ;
- Gérer les inscriptions et les paiements ;
- Gérer les examens, les convocations et les relevés de notes ;
- Offrir une interface avec le logiciel comptable du CNAM.

Grafic utilise le protocole réseau TCP/IP, le SGBD Oracle version 10g, des serveurs UNIX et WINDOWS 2003, le navigateur Internet Explorer et des postes de travail de type PC sous WINDOWS.

BDCEP est une application intranet que nous avons développée au sein de la Direction des Systèmes d'Information en PHP/MySQL. Elle permet au Centre d'Enseignement Parisien de gérer en complète autonomie, l'ouverture, la fermeture, les modifications et la tarification de ses sessions de formation. *BDCEP* contient les référentiels de la scolarité du CEP sur lesquels s'appuient les autres modules aussi développés par la DSI dont les noms sont : *catalogue, présence, calendrier, fiche UE, tableau de bord et réservation*. Les informations gérées par *BDCEP* sont ensuite « versées » dans *Grafic* pour la gestion de la scolarité.

La scolarité de l'INTEC est gérée par une application développée en interne nommée *UTINTEC*. Cette application fonctionne sous Oracle 10g en architecture 3-tiers. Elle prend notamment en charge :

- la gestion des inscriptions ;
- le dossier pédagogique ;
- la gestion des examens ;
- un module « comptable » qui enregistre les paiements, gère les relances;
- la gestion du diplôme.

Plusieurs autres applications développées en interne complètent UTINTEC :

- inscriptions en ligne ;
- prise de rendez-vous ;
- gestion des heures d'enseignement pour le paiement des enseignants ;
- support pédagogique ;
- produit de régie ;
- affichage des résultats sur le Web.

Hyperplanning est une application développée par la société *Index Education* et gérée par la DSI. Son rôle est de gérer les emplois du temps, la planification des salles et des ressources associées. Il est principalement utilisé par le CEP. Il autorise la délégation de droits, et l'accès aux informations via une interface Web.

GAGE UE et *GAGE Diplômes* sont des applications développées au sein de la Direction Nationales des Formations (DNF), dont les rôles respectifs sont :

- de gérer les agréments des enseignements et des enseignants au niveau des UE.
- de gérer les agréments au niveau des diplômes et des certificats.

ANGE est une application développée au sein de la Direction Nationales des Formations pour la délivrance des *diplômes* et du *supplément au diplôme* (pièce attachée au diplôme qui fournit une description détaillée du cursus suivi).

VES et *Viatic2* sont des applications développées au sein de la Direction Nationales des Formations (DNF), dont les rôles respectifs sont :

- la gestion de la VES (Validation des Études Supérieures) ;
- la gestion de la VAE (Validation des Acquis d'Expérience) et de la VAP85 (Validation des Acquis Professionnels de 1985).

EICNAM est une application développée par la DSI pour la délivrance des *diplômes* (et du *supplément au diplôme*) d'Ingénieur. Elle est utilisée par l'école d'ingénieurs afin de permettre le suivi d'un auditeur durant tout le processus de diplomation.

3) Le nouveau progiciel de scolarité

L'objectif de ce projet est de doter le Conservatoire National des Arts et Métiers d'un système unifié et homogène de gestion de la scolarité Parisienne. Ce système devra intégrer les fonctionnalités nécessaires à la gestion de la scolarité du Centre d'Enseignement de Paris (**CEP**), du Centre d'Actualisation des Connaissances et de l'Étude des Matériaux Industriels (**CACEMI**), de l'Institut National des Techniques Économiques et Comptables (**INTEC**) et de l'école d'ingénieurs du CNAM (**EICNAM**).

Ce système doit également être modulaire et non monolithique, c'est-à-dire regrouper en modules internes autonomes, les données et les traitements portant sur le même domaine d'activité (par exemple la gestion des prospects, la gestion des élèves, la gestion de l'offre de formation). Ces différents modules internes doivent interagir entre eux et former ensemble le système de gestion de la scolarité.

Cette modularité doit permettre de faire évoluer le système pour le proposer, par la suite, aux autres instituts du CNAM, et selon les besoins exprimés aux autres CRA. La société de service Logica a accompagné de CNAM durant toutes les étapes du projet de la mise en production du progiciel à la formation des utilisateurs.

Afin de pouvoir démarrer correctement les inscriptions avec le nouveau logiciel, celui-ci doit impérativement être alimenté avec l'historique de chaque auditeur. Afin de répondre à ce besoin il a été décidé de créer un entrepôt de données permettant d'accueillir et de traiter les données provenant de toutes les applications remplacées. Cette opération est la base de la mise en production du projet elle est donc une étape particulièrement sensible.

5. Les objectifs de l'entrepôt de données

1) Enjeux de l'entrepôt de données pour le CNAM

Au départ deux objectifs ont motivé la création d'un entrepôt de données : la reprise des données de scolarité afin de les intégrer dans SISCOL et la création d'un accès aux données consolidées sur l'activité du CNAM permettant aux utilisateurs finaux d'exécuter directement des requêtes sur les sources de données.

En dehors de l'intégration et du nettoyage des données de l'établissement afin de permettre un meilleur pilotage, un autre bénéfice attendu est de compenser l'effort fourni dans la construction et l'alimentation de l'entrepôt de données par un gain de temps lors de l'alimentation de nouvelles sources comme le logiciel SISCOL puisant dans cet entrepôt. Les applications alimentées par l'entrepôt portent sur des domaines fonctionnels plus restreints on parle parfois de « datamart ». Le but étant d'arriver à une meilleure gestion des auditeurs à travers la mise à disposition de tableaux de bord et d'outils de pilotage.

2) Objectifs principaux

La mise en cohérence progressive du SI passe à travers l'usage de référentiels pour l'ensemble de l'établissement et à la mise en place d'un entrepôt de données. Le but étant de migrer vers un système d'information qui se met progressivement au service du pilotage de l'établissement. Les bénéfices attendus sont :

- l'usage de référentiels
- faciliter la mise en qualité des données
- permettre le regroupement et le croisement des données
- contribuer à la cohérence du SI

Dans un premier temps le périmètre du projet comprend les domaines fonctionnels suivants :

- l'offre de formation du CNAM
- les inscriptions administratives
- des inscriptions pédagogiques
- la planification des séances

- la planification des salles
- la gestion des examens
- la délivrance des UE
- la gestion des certificats et diplômes
- les inscriptions administratives des élèves.
- l'offre de formation de la DNF (unités d'enseignement, sessions, diplômes, cursus)
- les personnels du CNAM (enseignants)
- les matériels
- les salles d'enseignement

Plus précisément, la reprise des inscriptions pédagogiques englobe les domaines suivants :

- les diplômes obtenus
- les diplômes en cours d'obtention
- l'historique des inscriptions aux unités d'enseignement
- les notes obtenues

Une généralisation progressive aux autres applications ainsi qu'aux autres établissements est programmée.

6. Objectifs de la reprise de données

1) Enjeux de la reprise de données

La reprise des données est un chantier important qu'il est difficile d'estimer précisément. Le CNAM a pour vocation de former ses élèves tout au long de la vie. Un élève peut faire valider un diplôme par consolidation des UE qu'il a obtenues durant plusieurs années. Il est donc impératif de pouvoir suivre le dossier élève, a priori, sans limite dans le temps. Par conséquent, la reprise des données concerne principalement les données actuellement gérées par les différents systèmes de scolarité (Grafic et UTINTEC) du CNAM, mais aussi par des produits divers (Access, Excel, bases diverses, …) utilisés dans les instituts ou les CRA.

De plus, en 2005 le CNAM est passé au système LMD (Licence, Master, Doctorat), ce qui a imposé le changement de codification des UE et des diplômes. Enfin en 2002, les pôles d'enseignement ont été créés, et les diplômes ont été répartis en quatre pôles.

La reprise de données avant ces deux périodes, suppose la transformation des anciennes références, dans la nouvelle nomenclature avant intégration dans le nouveau système de scolarité. Il faut donc valider les données avant leur intégration dans l'entrepôt puis dans le nouveau logiciel de scolarité. Dans tous les cas, les données éventuellement non reprises doivent être conservées et accessibles sous un format simple, et indépendant des systèmes actuels de gestion de la scolarité.

D'autre part, dans l'optique d'une consolidation de l'offre de formation nationale du CNAM incluant l'ensemble des formations, il doit être envisagé d'effectuer une reprise de données en une seule fois, de l'offre de formation actuelle, complétée avec les offres de formation locales, pour proposer par la suite une gestion homogène et unifiée à travers le système d'information de la scolarité.

2) Objectifs principaux

Afin de garantir une continuité lors du passage au nouveau logiciel ainsi qu'un niveau de qualité acceptable des données intégrées dans l'entrepôt, la reprise des données est un des aspects du projet sur lequel un effort tout particulier a été porté et sur lequel j'ai été particulièrement sollicité. L'objectif fut de reprendre les dossiers élèves des applications remplacées par Siscol (Figure 5).

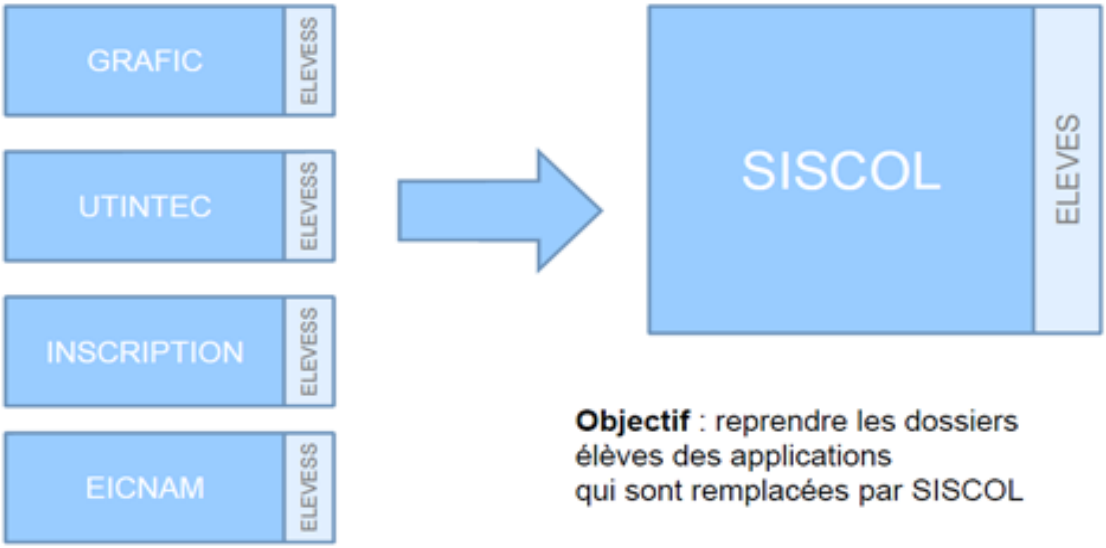


Figure 5 : Sources de données à intégrer dans SISCOL

Alimenter le nouveau progiciel

Lors du lancement du nouveau logiciel de scolarité, les élèves déjà inscrits au CNAM doivent pouvoir s'inscrire et retrouver dans le logiciel les données de leurs études. De plus le logiciel doit être alimenté avec l'offre de formation, les diplômes, etc.

Interfacer SAP avec l'entrepôt

L'objectif des interfaces sortantes est d'identifier les données de SAP à interfacer avec l'entrepôt de données. Dans un premier temps Il a fallu préciser les données utiles qui sont aujourd'hui extraites de SISCOL. La société de service LOGICA nous a fourni la structure des fichiers extraits ainsi que les référentiels utilisés et j'ai réalisé les programmes en entrée de l'entrepôt.

7. Synthèse

Dans ce premier chapitre j'ai présenté l'établissement en essayant de donner une vue d'ensemble de ses vocations. Puis, j'y ai présenté le service de la DSI dans son ensemble et plus particulièrement l'équipe application dont je fais partie. Enfin j'espère y avoir clairement présenté les principaux aspects du projet de scolarité en m'appliquant à bien situer les parties sur lesquelles j'ai été mis à contribution. La mise en place d'un entrepôt de données ainsi que son alimentation sont des éléments indispensables à la réussite du projet auquel il a fallu attacher un soin particulier. L'objectif de la deuxième partie est dans un premier temps de présenter les concepts liés aux entrepôts de données puis dans un second les outils permettant de traiter les flux de données intégrés ou extraits de ces entrepôts.

II. Introduction aux entrepôts de données

Ne possédant qu'une connaissance élémentaire, commune à tous les ingénieurs, dans la construction et l'alimentation d'un entrepôt de données, j'ai d'abord été amené à me documenter sur les caractéristiques des systèmes d'information décisionnels (SID) par opposition aux systèmes d'information opérationnels (SIO). J'ai aussi été formé aux outils de gestion des flux de données dont je détaillerai le fonctionnement dans le quatrième chapitre.

La Figure 6 offre une visibilité sur les différents flux de données entrant et sortant d'un entrepôt de données. Il peut être vu comme une interface entre les sources de données opérationnelles et les décideurs.

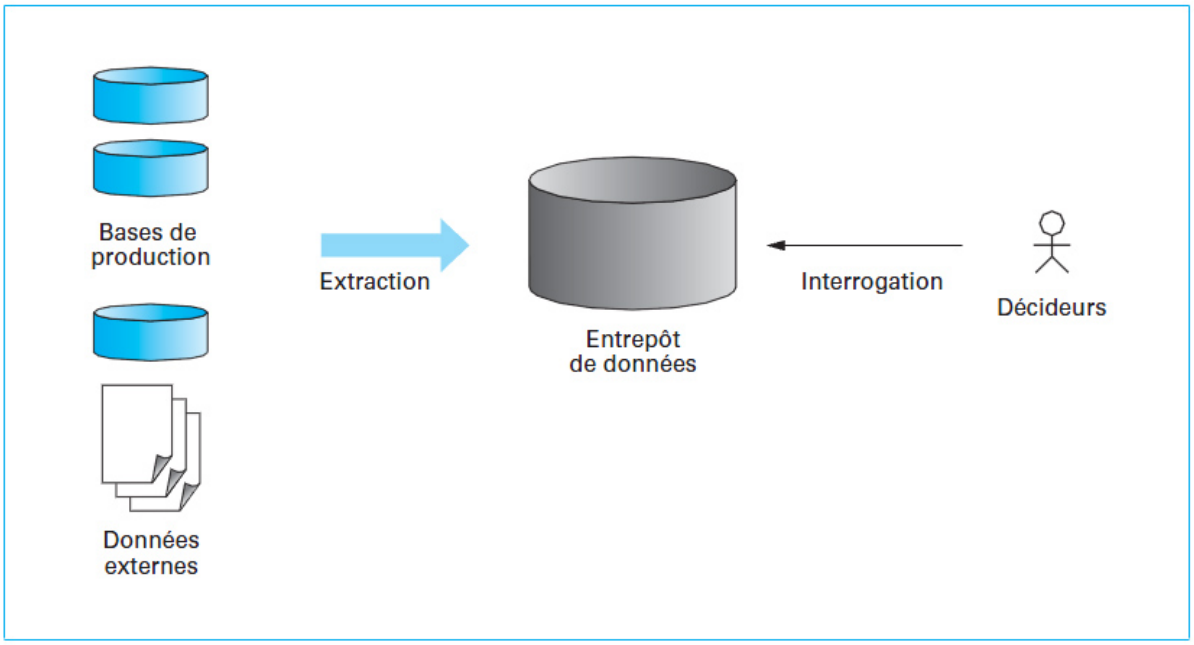


Figure 6 : Interface entre les sources d'information et les décideurs

Il est important de considérer un projet de création d'un entrepôt de données, non pas comme un projet unique, mais plutôt comme une succession de projets plus légers, focalisés sur les besoins métiers, répondant chacun à une nécessité clairement identifiée et définie. Chacun des projets s'intégrant avec le précédent et ouvrant des pistes pour les suivants.

1. L'entrepôt de données

1) Définition

Un entrepôt de données est un objet informatique se présentant sous la forme d'une base de données d'aide à la décision. Cet objet diffère d'une base de données classique de par son fonctionnement et ses utilisateurs. En effet, les bases de données classiques sont des systèmes qui gèrent des données résultant d'un très grand nombre de transactions simples effectuées à des fins de gestion courante. A contrario, les entrepôts de données sont des bases qui gèrent des données historisées permettant, par des analyses en ligne, aux décideurs de l'entreprise d'effectuer des choix ou de prendre des décisions. Les données présentes dans un entrepôt ne sont pas directement modifiables par les utilisateurs, mais elles sont dérivées des données des bases opérationnelles qui régissent les activités de l'entreprise. Les données dérivées sont intégrées périodiquement à l'entrepôt. L'utilité de cet outil est de stocker sur de longues périodes des mesures sur certaines des activités de l'entreprise, ce qui permet de les étudier ultérieurement.

D'après Ralph Kimball, le datawarehouse se représente ainsi (1):

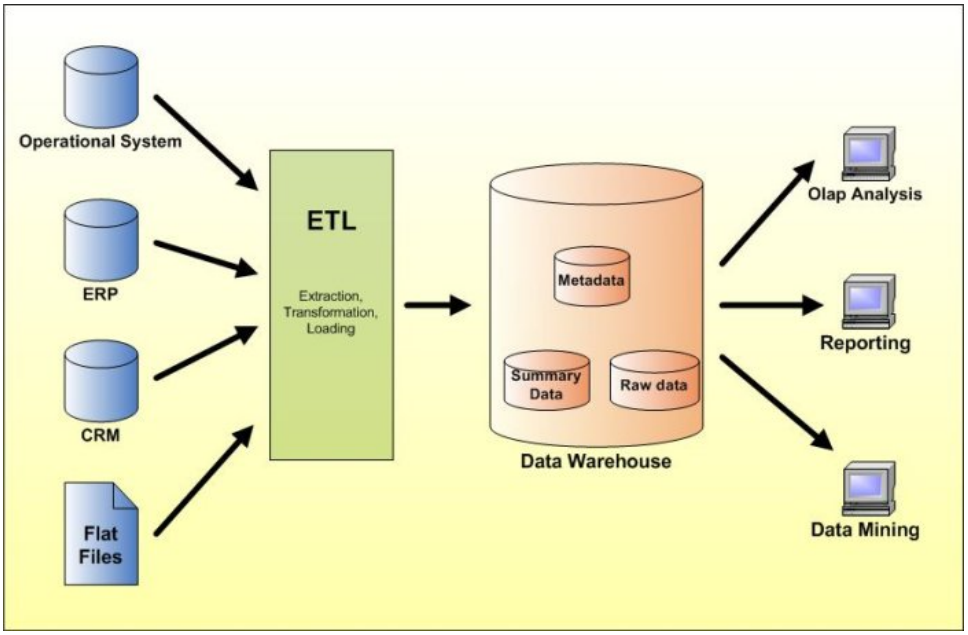


Figure 7 : Composants de base d'un entrepôt de données

Sa définition est assez large et englobe tout le processus de cette conception.

Il prend en premier lieu les systèmes sources. D'après son modèle les systèmes sources sont comparables aux systèmes de production. Nous y retrouvons les données liées à l'activité. En seconde partie nous trouvons la « zone de préparation de données ». Il définit ainsi tout un processus qui a pour but de nettoyer (purge, suppression de doublon, ….) les données provenant des systèmes sources. Ce nettoyage permet d'alimenter la phase suivante. En troisième plan nous avons le « serveur de présentation». Celui-ci est découpé en sous parties, elles-mêmes alimentées par la « zone de préparation des données ». Enfin la partie « portail de restitution » correspond à la partie utilisateur. Elle permet l'accès aux données contenues dans le « serveur de présentation ».

2) Définition générales

Le créateur du concept d'entrepôt de données, Bill Inmon, le définit comme « *collection de données orientées sujet non volatiles et historisées, organisé pour le support d'un processus d'aide à la décision* » (2).

D'un point de vue conceptuel, un entrepôt de données enregistre des collections de données agrégées provenant de différentes sources hétérogènes. La création d'un entrepôt de données implique de fournir un effort sur du long terme. D'après Claude Chrisment (3), la constitution d'un entrepôt de données principal (unique) représente un investissement trop important pour une équipe d'informaticiens au sein d'une entreprise. Il est donc nécessaire d'étaler l'activité correspondante sur plusieurs années. De plus d'après Scott Arnette (4) l'intégration doit se faire de manière itérative car les champs couverts par un sujet augmente lorsque les utilisateurs affinent leur analyse et augmente leur besoin en données. De plus il préconise l'approche « sujet par sujet » lors de la construction d'un entrepôt.

3) Caractéristiques

Toujours d'après Bill Inmon, les données d'un Entrepôt de données doivent respecter les caractéristiques suivantes :

Intégrées : Les données de l'entrepôt proviennent de différentes sources éventuellement hétérogènes. L'intégration consiste à résoudre les problèmes d'hétérogénéité des modèles, des schémas, de la sémantique.

Orientées sujet : Les données de l'entrepôt sont organisées autour des thèmes qui ont un intérêt majeur pour l'entreprise, le but de cette organisation est de disposer de l'ensemble des informations utiles sur un thème le plus souvent transversal aux structures fonctionnelles et organisationnelles de l'entreprise tels que : le client, le produit, les ventes… Cette orientation par thème va permettre à l'entreprise de développer son système décisionnel progressivement, c'est une approche par itération.

Non volatiles : Un datawarehouse veut conserver la traçabilité des informations et des décisions prises. Les données ne sont ni modifiées ni supprimées. Une requête émise sur les mêmes données à plusieurs mois d'intervalles doit donner le même résultat. Un datawarehouse définit donc à la fois un ensemble de données et un ensemble d'outils. Il s'agit de données destinées aux décideurs, qui sont souvent une copie des données de production avec une valeur ajoutée (orientées objet, agrégées, historisées). Et c'est un ensemble d'outils permettant de regrouper les données des différentes sources, de les nettoyer et de les intégrer, ainsi que d'y accéder de différentes manières (requêtes, rapport, analyse, datamining).

Historisées : La prise en compte de l'évolution des données est primordiale pour la prise de décision et notamment les prédictions. Dans un système de production ; la donnée est mise à jour à chaque nouvelle transaction. Dans un datawarehouse, la donnée ne doit jamais être mise à jour. Un référentiel temps doit être associé à la donnée afin d'être capable d'identifier une valeur particulière dans le temps.

Résumées : Les données peuvent être agrégées dans certains cas, pour optimiser la prise de décision.

Processus d'aide à la décision : Les utilisateurs doivent avoir accès aux données qui leur sont autorisées.

Disponibles pour l'interrogation et l'analyse : Les utilisateurs doivent pouvoir consulter les données réorganisées de l'entrepôt en fonction de leurs droits d'accès. De plus, l'entrepôt de données offre à l'entreprise les avantages suivants :

- Il constitue une collection de données centralisées disponibles pour l'aide à la décision (OLAP, datamining,...).
- Les évolutions des données de l'entrepôt sont conservées (historisation des données).
- Il contient un ensemble de données consolidées (données homogènes et fiables).
- Il contient des données agrégées permettant une analyse à différents niveaux de détails.
- Il permet de développer différents thèmes d'analyse (réorganisation en fonction des sujets à analyser).

2. Les datamarts

Parfois traduits par magasins de données, Ralph Kimball (1) définit les magasins de données de comme des sous-ensembles de l'entrepôt de données. Ils sont constitués de tables plus détaillées et plus agrégées permettant de restituer tout le spectre de l'activité liée à un métier. L'ensemble des datamarts de l'entreprise constitue le datawarehouse. Pour Bill Inmon (5) le datamart est issu d'un flux de données provenant de l'entrepôt de données. Contrairement à ce dernier qui présente le détail des données pour toute l'entreprise, il a vocation à présenter la donnée de manière spécialisée, agrégée et regroupée fonctionnellement.

Par ailleurs, Il existe plusieurs manières d'organiser un magasin de données. Soit par service ou fonctions de l'établissement par exemple un magasin de données pour les ressources humaines, un pour la scolarité. Soit par sous-ensemble organisationnel, par exemple un magasin de données par entité (CEP, INTEC), centres régionaux, …..

Les magasins de données étant des extraits simplifiés du détail des données de l'entreprise, ils ne présentent d'intérêt que pour des requêtes identifiées et répétitives. Il est plus facile pour le système d'interroger un magasin de données qui ne contient que le nécessaire que d'avoir à cerner et à trier toute la base. Par ailleurs, les datamarts permettent de classifier et de clarifier l'information, de manière à ce que chaque métier ait accès à des chiffres correspondant à ses attentes fonctionnelles, sans être pollué par des données contiguës. En revanche, les choix de simplification qui donnent lieu aux magasins de données rendent ceux-ci naturellement moins flexibles. Des demandes d'utilisateurs sortant de leur cadre habituel requièrent fréquemment d'interroger la base à un autre niveau, générant des coûts de développement ou la création de solutions de rechange. Des problèmes peuvent de fait survenir lorsque les magasins de données constituent l'unique moyen d'accès aux données pour l'utilisateur final.

1) Modélisation d'un magasin de données

Nous venons de présenter le cœur de la base de données décisionnelle. Le projet global peut se nommer entrepôt de données. Au sein de celui-ci nous retrouvons les magasins de données. Maintenant nous allons expliquer comment sont organisées les données au sein d'un magasin. La modélisation multidimensionnelle repose sur les concepts de fait, de dimension et de hiérarchie. Cette modélisation consiste à décrire les données analysées au travers d'un schéma en étoile.

2) Les faits

Les faits représentent les informations quantifiées de l'entreprise. Ils sont parfois nommés faits, indicateurs ou encore mesures. Ce sont les données à analyser correspondant à l'activité de l'entreprise. Les indicateurs ont la particularité d'être additifs. Ils sont contenus dans une table physique de la base de données décisionnelle. Lorsque l'on regroupe plusieurs indicateurs on parle de portefeuille d'indicateurs, de table des faits ou de table des mesures. Les indicateurs n'ont d'intérêt que s'ils sont mis en valeur par des informations. Une ligne de faits correspond aux valeurs de l'intersection des tables des dimensions. Grâce aux dimensions, nous déterminons la granularité (la finesse) des résultats contenus dans la table des faits.

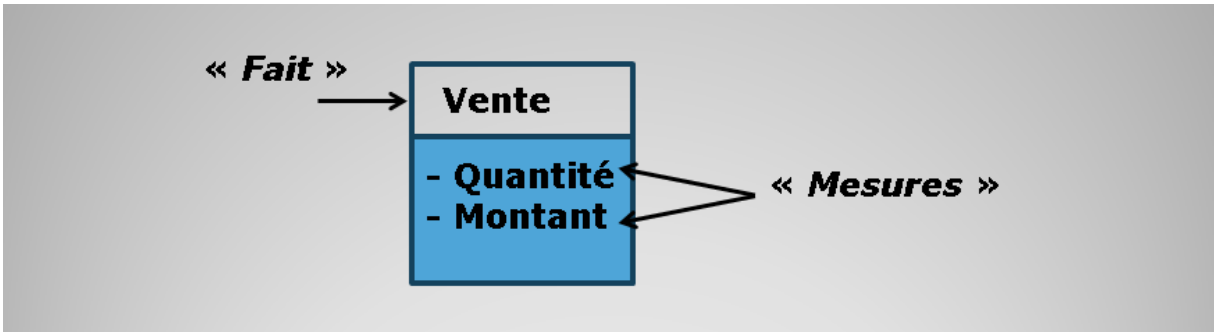


Figure 8 : Les faits

3) Les dimensions

Les axes d'analyse, selon lesquels un fait est observé, sont modélisés par des dimensions. Tout comme les faits, les dimensions sont contenues dans des tables physiques de la base de données. Ce sont les informations qui mettent en évidence les données contenues dans les tables des faits. Les dimensions comportent un ou plusieurs attributs qui sont le

plus souvent hiérarchisés. Une hiérarchie modélise les niveaux de granularité auxquels les mesures sont observées. Par exemple dans la dimension géographique pourrait avoir comme hiérarchie la région, le département et la ville (Figure 9).

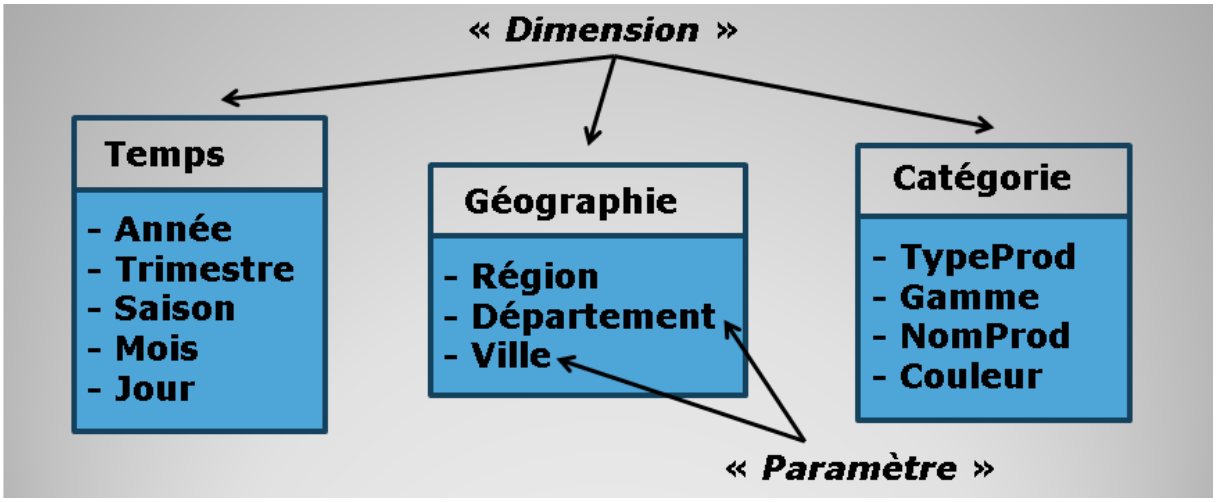


Figure 9 : Les dimensions

Les dimensions ont pour rôles :

- D'afficher les données : Ce seront les entêtes des lignes ou des colonnes pour regrouper les faits caractérisant ainsi la donnée brute contenue dans la table des faits.
- De filtrer les données afin d'obtenir un tableau correspondant aux attentes des utilisateurs.

3. Modélisation logique des données

La modélisation multidimensionnelle intègre donc des concepts spécifiques pour lesquels les notations existantes (entité-relation, UML) s'avèrent imparfaites. L'inadéquation des notations existantes est par exemple liée à la représentation des hiérarchies associées à chaque dimension. Des notations spécifiques doivent être proposées. Les figures 9 et 10 présente des exemples de schéma conceptuel décrit avec des notations spécifiques adaptées. Sur ces figures, le fait représenté concerne des VENTES dont les mesures les caractérisant sont le montant et la quantité. Les ventes sont analysées suivant trois

dimensions : TEMPS, PRODUITS et CLIENTS. Sur chacune d'elles sont définies des hiérarchies dont les paramètres sont représentés par un cercle tandis que les attributs faibles associés sont soulignés. Il existe plusieurs techniques de modélisation multidimensionnelle dont la modélisation en étoile et la modélisation en flocon de neige.

1) Modélisation en étoile

La modélisation en étoile doit son nom à sa forme. Au cœur de ce modèle se trouve la table des faits. Autour, les dimensions donnent chacune un axe d'analyse différent. Cette modélisation ne tient pas compte des formes normales car elle a uniquement l'analyse comme préoccupation. La table des faits est la seule table à contenir des jointures avec les dimensions.

Ce schéma très performant pour la restitution de données est plus gourmand en espace de stockage. Voici une représentation (Figure 10) d'un modèle en étoile sur un exemple très simple de gestion des ventes avec analyse du lieu, de la période de vente et du produit :

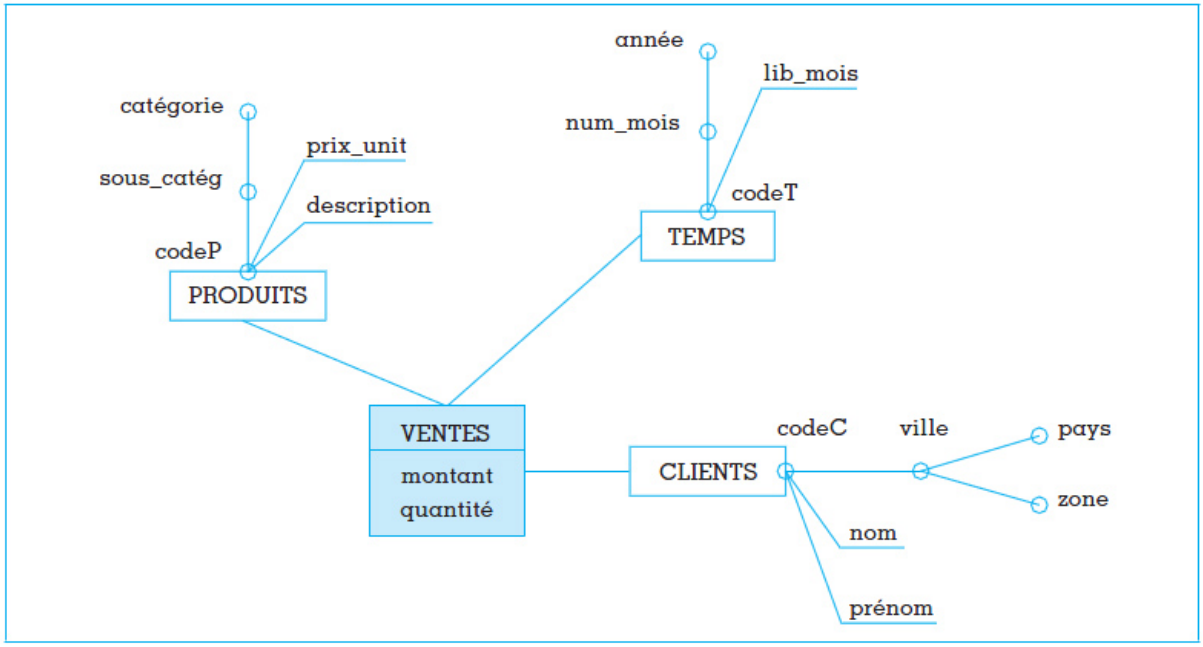


Figure 10 : exemple de table des faits et dimensions

2) Modélisation en flocon de neige

Le modèle en flocon de neige est constitué d'une table des faits au centre et des tables des dimensions autour, comme pour le modèle en étoile. La différence se situe au niveau des tables des dimensions, qui peuvent également se diviser en plusieurs branches différentes. Ces branches sont souvent utilisées pour modéliser des hiérarchies. Cependant d'après Ralph Kimball (1), elles peuvent engendrer un certain nombre de points négatifs comme la difficulté de compréhension par des non informaticiens et l'alourdissement des requêtes par un nombre grandissant de jointure. Il estime même que l'argument du gain de place n'est pas forcement fondé, lorsqu'il est comparé à la table des faits qui est très volumineuse.

3) Modélisation en constellation

Les modèles en étoile ne gèrent qu'une seule table des faits. Il est très fréquent d'avoir recours à plusieurs tables des faits, donc plusieurs étoiles pour décrire l'activité d'une entreprise. Ces différentes étoiles auront peut-être des dimensions communes. En reliant ces dimensions ensemble on obtient une constellation. En ajoutant la table des faits « prix» et la dimension « temps », la dimension « produit » est commune aux deux tables des faits. La modélisation d'une telle réalité analytique consiste à décrire un schéma en constellation (Figure 11) issu de l'intégration de plusieurs sous-schémas en étoile:

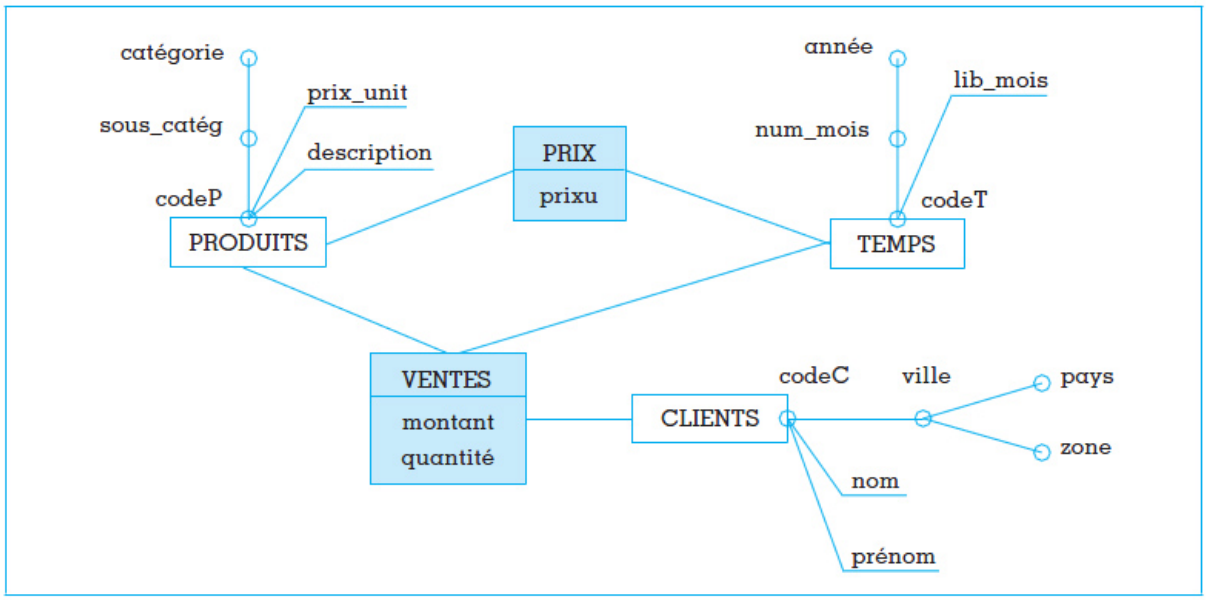


Figure 11 : Modélisation en constellation

4) Comparaison des modèles en étoile et en flocon

Le modèle en flocon offre une vue plus claire de la structure de l'information permettant notamment de déceler une hiérarchie. La normalisation de ce modèle permet de plus de diminuer la redondance, en réduisant la taille des tables de dimension. A noter que Kimball (1) a évalué le gain de place disque à 1 % de l'espace disque total.

De plus Ralph Kimball (1) préfère le modèle en étoile sur la base de deux arguments :

- la dénormalisation permet d'améliorer les performances du système lors de l'exécution des requêtes.
- le modèle est plus facile à apprendre par l'utilisateur non informaticien

4. Alimentation de l'entrepôt et ETL

1) Introduction

La notion d'ETL (Extract Transform Loading), recouvre à la fois des outils et des processus d'alimentation d'un entrepôt de données. Il s'agit d'un élément clé dans l'intégration d'applications, en particulier dans le monde de la Business Intelligence et du « datawarehousing ». Les Outils et les processus ETL sont des briques d'une infrastructure du système d'information dont la valeur ajoutée et le retour sur investissement s'expriment dans le temps en accompagnant l'évolution du système d'information global ou du système d'information décisionnel. L'utilisation des bases de données s'est répandue dans les entreprises dans le courant des années 60. Les besoins en intégration de données, liés à la nécessité de combiner et de questionner d'une manière homogène des données de plusieurs sources autonomes et hétérogènes, sont alors rapidement apparus. Ces besoins ont évolué au cours du temps mais sont toujours d'actualité : après trois décennies de recherches il n'existe toujours « pas de solution miracle ». Les questions soulevées par l'article de P. Ziegler et KR. Dittrich (6) demeurent actuelles.

2) L'Intégration ou alimentation de données

Les outils ETL gèrent toutes les étapes de la collecte des données au sein des systèmes d'information hétérogènes : SGBD, ERP, applications spécifiques, fichiers plats, bases hiérarchiques... depuis le nettoyage des données collectées, la consolidation et la mise en concordance des données éparses jusqu'à leur distribution auprès des applications cibles ou des systèmes décisionnels. Le processus ETL est une opération de migration de données qui consiste aussi à la rendre aisément consommable. Ce processus représente une part majeure des traitements et nécessite une attention régulière tout au long du cycle de vie du système, dans la mesure où il est garant de la qualité des données. Un processus ETL se décompose en trois phases : l'extraction, la préparation ou transformation et le chargement.

3) Extraction des données

Il s'agit en premier lieu d'aller chercher les données là où elles se trouvent. L'outil ETL a la capacité de se connecter aux différentes applications, bases de données ou fichiers. Pour cela, plusieurs technologies sont utilisables :

- Les passerelles fournies par les éditeurs de logiciels de gestion de bases de données.
- Les utilitaires de réplication, utilisables si les systèmes de production et décisionnels, sources et cibles, sont homogènes.
- Les outils spécifiques d'extraction. L'outil doit être à même de lire sélectivement les données sources, et donc de filtrer les données en lecture afin de n'extraire que l'information pertinente.

4) Le chargement et le transfert des données

Le chargement prend en compte la gestion du format final des données. Pour la mise en œuvre du transfert de données, on distingue deux approches possibles :

- Le transfert de fichiers : l'ETL transporte les données du système source vers le système cible via un moteur.
- Le transfert de base à base. Dans ce cas, les outils travaillent en mode connecté, d'une source de données à une cible. Les données sont extraites ensemble à la source, puis transférées à la cible en y appliquant éventuellement des transformations à la volée. Un seul processus, plus rapide, a ainsi l'avantage de pouvoir effectuer, sans rupture, les transferts et toutes les autres opérations d'alimentation.

5) Comparatif des outils actuels

Les nombreux prototypes d'extraction et de transformation des données dont ceux figurant dans le Tableau 1 : Comparaison de quelques ETL du marché (7), permettent l'extraction de structures et expressions régulières, la transformation de valeurs de données par application de fonctions de formatage, la transformation de l'ensemble des valeurs de n-uplets (lignes) et d'attributs (colonnes) d'une base de données relationnelle.

Tableau 1 : Comparaison de quelques ETL du marché

Editeur	Solution	Tarification	Principaux clients
IBM	Information Server	Dès 88 000 € (avec 1 an de maintenance comprise)	NC
Informatica	Power Center	Dès 50 000 €	EDF, Société Générale, Danone...
Oracle	Warehouse Builder et Sunopsis Data Conductor	Dès 25 000 €	Caroll, Arpège Groupe Caisse d'Epargne...
Microsoft	SQL Server Integration Services	Dès 24 000 $ (SQL Server 2005 uniquement)	NC
Talend	Open Studio	Offre Open-Source gratuite (formation expertise 1000€/jour et support dès 900 €/mois)	Groupe Accor, GMF, Direction Générale de la Comptabilité Publique...

Tableau 2 : Comparaison des ETL du marché

Editeur / Solution	Principales briques fonctionnelles	Extraction et type d'architecture	Connecteurs ERP natifs > 6
IBM / Information Server	Ordonnancement des tâches, gestion centralisée des metadonnées (*Metadata Server*), gestion des glossaires métier (*Business Glossary*), administration des scénarios d'alimentation et qualité des données (*Datastage*)...	Extraction batch temps réel et/ou via MOM. Architecture : Hub&Spoke	SAP, Oracle, PeopleSoft et Siebel
Informatica / Power Center	Ordonnancement des tâches, gestion centralisée des metadonnées, administration des scénarios d'alimentation et qualité des données, profiling et accès aux données non structurées...	Extraction batch temps réel et/ou via MOM. Architecture : Hub&Spoke	(SAP, Oracle, Siebel, PeopleSoft...)
Oracle / Warehouse Builder et Sunopsis Data Conductor	Ordonnancement des tâches, gestion centralisée des metadonnées, qualité des données...	NC	Oracle, PeopleSoft, Siebel...
Microsoft / SQL Server Integration Services	Ordonnancement des tâches, gestion centralisée des metadonnées, qualité des données...	Extraction batch temps réel et/ou via MOM	NC
Talend / Open Studio	Ordonnancement des tâches (*scheduler*), gestion centralisée des metadonnées (*MetadataManager*), gestion des glossaires métier (*Business Modeler*), administration des scénarios d'alimentation (*rAMC*) et qualité des données...	Extraction batch temps réel. Architecture : Hub&Spoke (Network Centric via MOM)	OpenBravo et SAP (en cours de développement)

Jusqu'à présent, le marché de la qualité des données concerne principalement les entreprises possédant de grands systèmes d'information (institutionnels, télécoms, transports, banques, énergéticiens, etc.). Les familles de produits commercialisés répondant à la problématique de la qualité des données présentées dans le Tableau 2 nous montrent qu'il n'existe pas encore sur le marché d'approche intégrée qui permette :

- de spécifier et vérifier des contraintes métier sur une base de données multi sources ;
- de traiter l'ensemble du processus d'analyse de la qualité des données, depuis la spécification et la détection de l'anomalie jusqu'à son explication et sa correction.

5. L'outil Talend

Bien qu'ils soient généralement invisibles pour les utilisateurs de la plate-forme décisionnelle, les ETL reprennent les données de tous les systèmes opérationnels et les traitent pour les outils d'analyse et de reporting.

Talend offre une très grande connectivité aux :

- Progiciels (ERP, CRM, etc.), bases de données, serveurs centraux, fichiers, Web Services, etc. pour couvrir la disparité grandissante des sources.
- Entrepôts de données, magasins de données, applications OLAP pour analyse, etc.
- Composants ETL avancés stockés localement, incluant des manipulations de chaînes, traitement automatique des références, etc.

L'intégration opérationnelle de données est souvent utilisée pour implémenter les programmes et routines habituels, complétée en fonction des besoins spécifiques.

Les applications de chargement et migration de données ou de synchronisation de données sont les plus répandues en matière d'intégration opérationnelle de données. Elles requièrent :

- des correspondances et transformations complexes avec des fonctions d'agrégation et de calculs pour pallier les différences dans les structures des données.
- le traitement et la résolution des conflits de données en tenant compte des mises à jour des enregistrements ou des "propriétaires des enregistrements".
- la synchronisation de données en quasi temps réel étant donné que les systèmes impliquent une latence lente.

Talend est entièrement écrit en langage de programmation Java, fonctionne sous Eclipse et s'appuie sur les langages standard de génération de codes : Java, Perl et SQL. Cela permet à notre équipe de tirer parti de son expertise dans ces technologies.

1) Fonctionnement

Afin de mieux comprendre les différents traitements sur les données reposant sur l'utilisation de l'outil Talend, il est important d'appréhender certains concepts et de bien comprendre le fonctionnement de cet ETL.

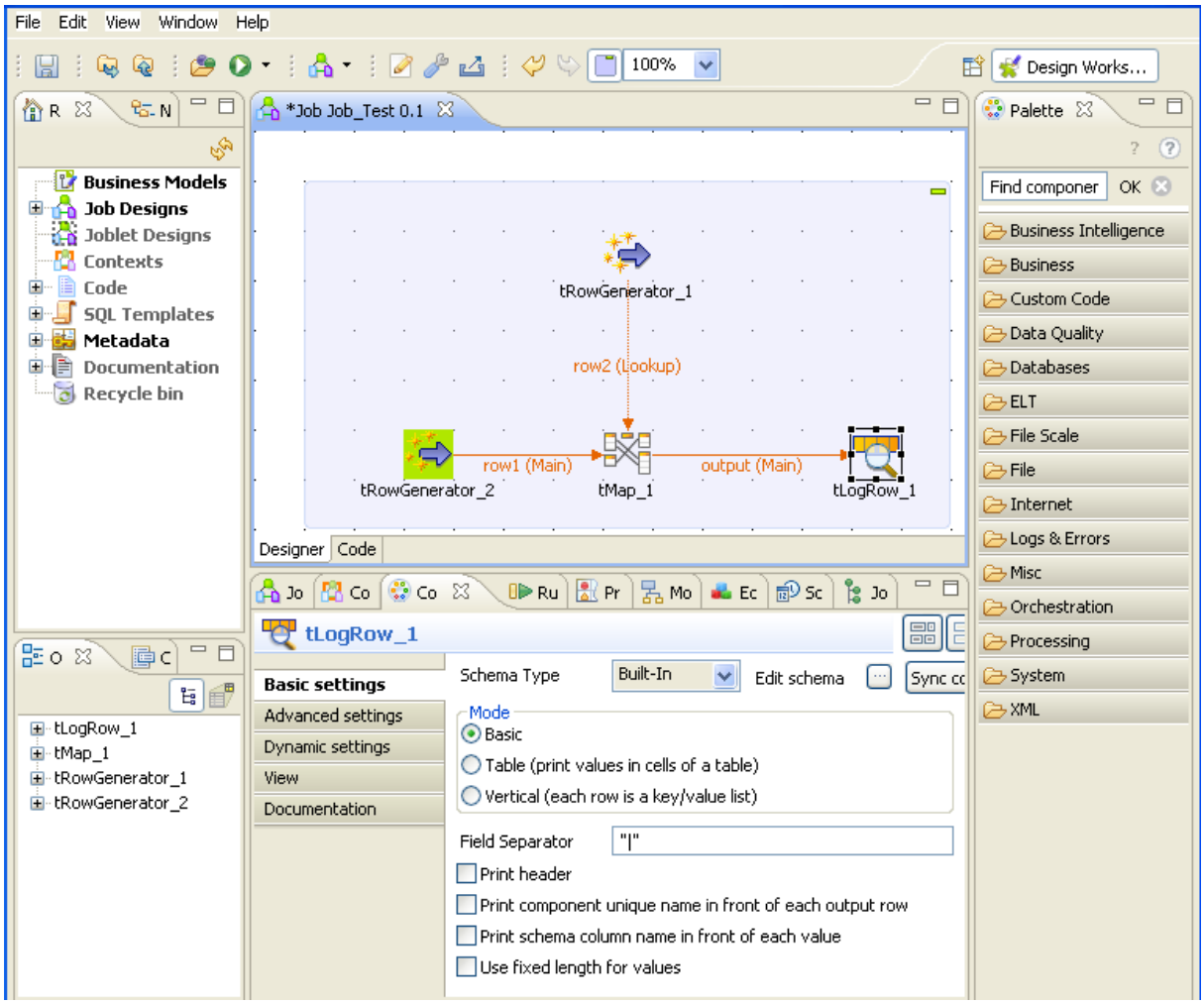


Figure 12 : Fenêtre de composition de Talend Open Studio

La fenêtre de Talend Open Studio (Figure 12) est composée des vues suivantes :

- Barres d'outils et menus
- Référentiel
- Espace de modélisation
- Diverses vues de configuration organisées en onglets
- Aperçu du schéma
- Aperçu du code

2) Le référentiel

Le référentiel (Figure 13) est une arborescence regroupant les éléments techniques disponibles pour la description des « Business model » et la conception des « Job design ». Le référentiel donne aussi un accès aux contextes et aux métadonnées ainsi qu'à toutes les routines et documentations réutilisables. Ce référentiel centralise et conserve localement tous les éléments contenus dans un projet. Le référentiel regroupe les éléments suivants sous forme d'une arborescence :

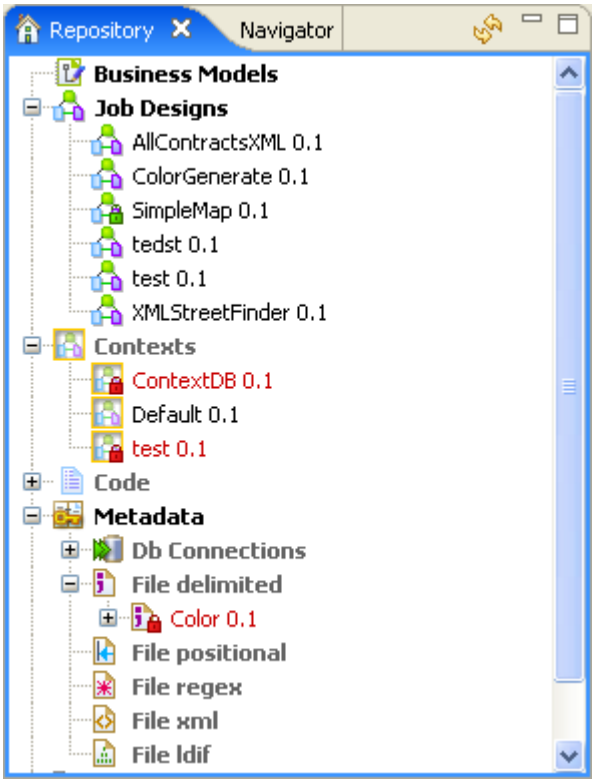


Figure 13 : Le référentiel Talend

Le « business model »

Toutes les représentations graphiques des processus métier d'un projet sont regroupées sous le nœud « Business Model » (Figure 14). Le Business Modeler intégré à Talend permet aux acteurs fonctionnels de prendre part à la conception des flux de données et de suivre de près l'avancement des développements. Dans l'espace de modélisation du « Business Modeler », l'équipe fonctionnelle modélise et documente les processus d'intégration sous forme de diagrammes qui permettront ensuite d'orienter le développement et de faire évoluer les processus en fonction des besoins.

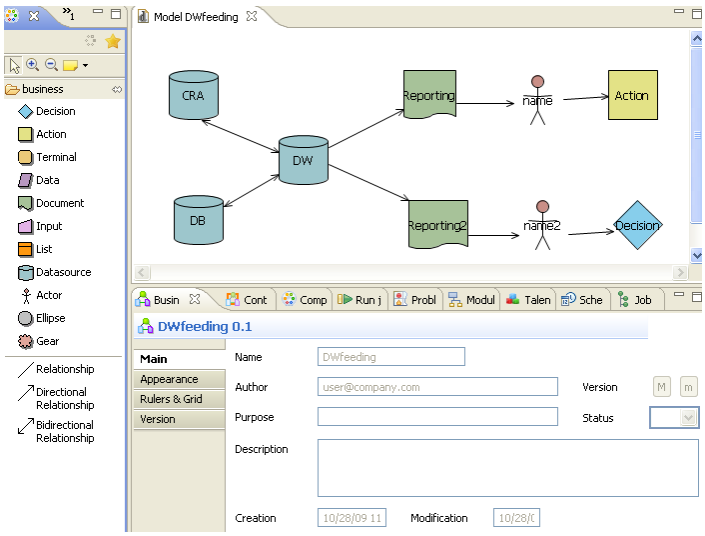


Figure 14 : « Business Model »

Les jobs

Un Job (Figure 15) constitue la couche d'exécution ou l'implémentation technique d'un business model. Il traduit les besoins métier en codes, en routines ou en programmes, puis se charge d'exécuter ces derniers. En d'autres termes, le Job permet de mettre en place votre flux de données. C'est lui qui effectue les transformations entre les données d'entrées et les données de sorties.

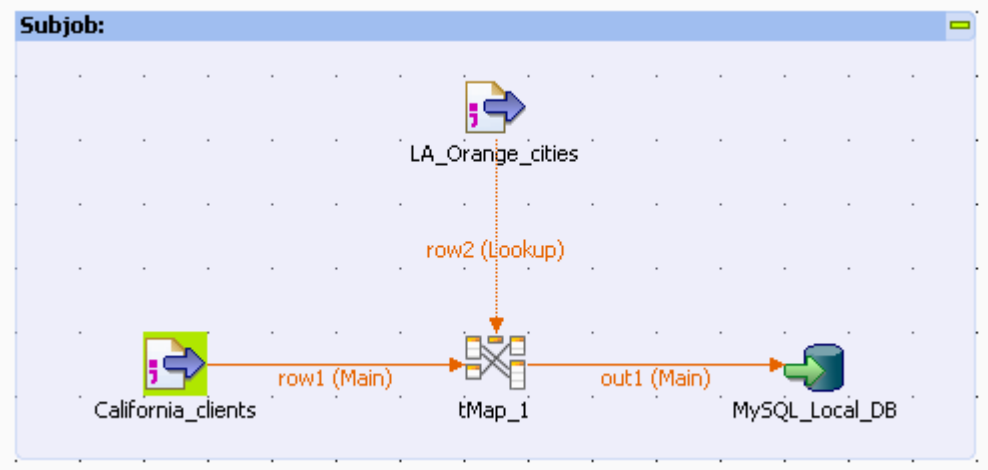


Figure 15 : Job Talend

La gestion des versions des jobs

Tout au long d'un projet les différents programmes réalisés évoluent au fur et à mesure que les besoins s'affinent. Talend propose un mécanisme de gestion des versions permettant des montés de versions et un retour sur une version stable.

Les métadonnées

Le répertoire « Metadata » regroupe les informations que l'on souhaite réutiliser dans nos différents Jobs, notamment les schémas et les informations de propriétés. C'est très utile lorsque l'on partage des sources de données ou des schémas entre les différents jobs. Il est possible de créer :

- des connexions DB ;
- des schémas de fichier ;
- des schémas LDAP ;
- des schémas génériques ;
- des schémas WSDL.

Les contextes

Le nœud contextes rassemble les fichiers contenant les variables de contexte que l'on souhaite utiliser dans les différents Jobs, tels que les chemins d'accès ou les informations de connexion aux bases de données.

Les codes

Le nœud Code correspond à une bibliothèque rassemblant toutes les routines disponibles pour ce projet. Il est possible de mutualiser ce code et le réutiliser dans divers composants ou Jobs.

Les routines

Une routine (Figure 16) est un morceau de code plus ou moins complexe généralement utilisé de façon itérative dans un ou plusieurs Jobs. Dans le référentiel un dossier System regroupe toutes les routines Talend prédéfinies.

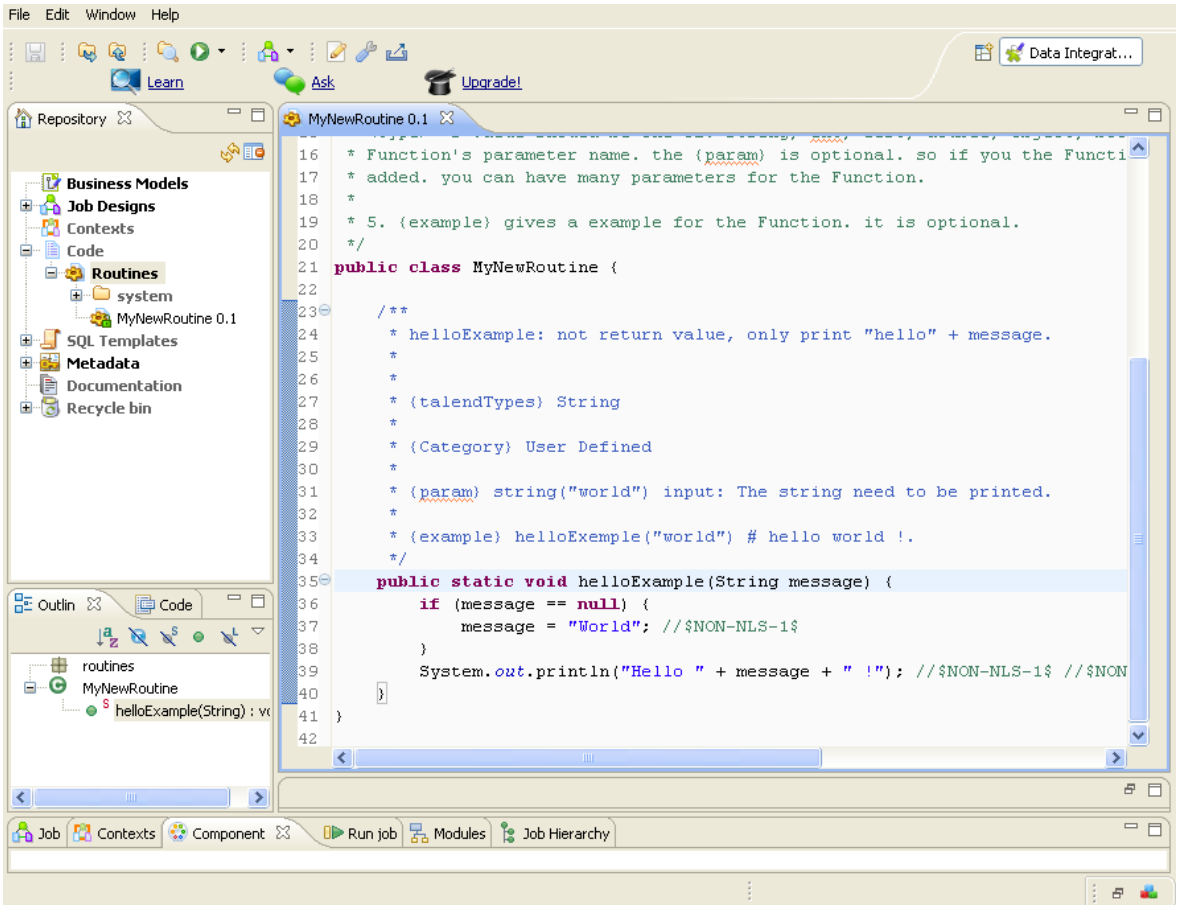


Figure 16 : Création d'une routine

3) La palette de composants

Dans Talend les actions sont modélisées grâce à la bibliothèque de composants techniques. La Palette de composants inclut plus de 200 composants et connecteurs techniques et métier, qui fournissent des fonctionnalités de base telles que les correspondances, les transformations et les recherches, des fonctionnalités avancées comme le filtrage et le multiplexage de données, l'ELT, le support de la plupart des SGBDR, formats de fichiers, annuaires LDAP, etc. La Palette de composants peut être étendue grâce à des langages standards comme Perl, Java, ou SQL.

	Le composant *tDB2Input* (connecteur DB2) fournit une connectivité native à DB2. Ce connecteur inclut des fonctionnalités avancées d'analyse de métadonnées permettant à l'équipe d'exploitation de comprendre aisément la structure de la base de données DB2 du CRM (même si toute la documentation du système existant a été perdue depuis longtemps).
	Le composant *tFileInputLDIF* (connecteur LDAP) sert à extraire de l'annuaire LDAP de l'entreprise des informations utilisateurs et permet de les utiliser comme données d'entrées vers les mappings.
	Le composant *tWebServiceInput* (connecteur services Web) permet la connexion vers un service fournissant des coordonnées géographiques : il établit les adresses des clients en tant que paramètres et retourne les adresses géocodées.
	Le composant *tMap* permet aux développeurs de mapper aisément les données issues de DB2 vers MySQL, de les enrichir avec d'autres données issues de l'annuaire LDAP et des services Web de géocodage et de transformer les données à partir du schéma DB2 vers le modèle de données Salesforce.com.
	Le composant *tFuzzyMatch* présente des fonctionnalités avancées de comparaison de données permettant d'identifier les doublons (même si les données n'ont pas une correspondance à 100%).
	Le composant *tMysqlOutput* (connecteur MySQL) permet une connectivité native à MySQL dans lequel le modèle de données Salesforce.com a été dupliqué (grâce aux fonctionnalités avancées de gestion des métadonnées de Talend Open Studio).
	Le composant *tSalesforceOutput* (connecteur métier Salesforce.com) permet une connectivité native à Salesforce.com, en exploitant ses API services Web et le langage SOQL.

Figure 17 : Principaux composants de Talend

Il existe des composants par défaut (Figure 17) permettant par exemple de filtrer des flux, de lire le contenu d'un répertoire, d'interroger une base de donnée ou encore de créer des fichiers et il est possible créer de nouveaux composants ou familles de composants correspondant à un besoin spécifique.

Ces composants sont paramétrables et les relations entre les composants définissent la nature des actions ainsi que leur déroulement. Il est toujours possible d'accéder au code généré pour le programme par les composants afin de le modifier ou de le documenter (Figure 18).

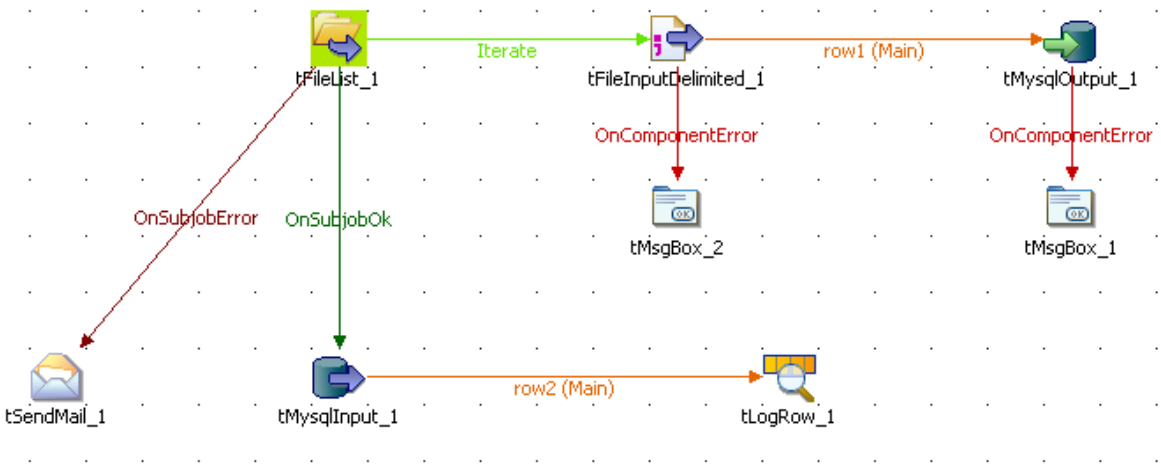


Figure 18 : Exemple de composant et de leurs connexions

4) Le job designer

Le nœud Job design rassemble tous les Jobs (les programmes) du projet courant. Le job designer de Talend fournit une vue à la fois graphique et fonctionnelle des processus d'intégration. Une boîte à outils graphique, la Palette, permet aux développeurs de disposer de tous les composants techniques et connecteurs correspondant à chacune des étapes de l'implémentation technique des besoins modélisés. Les processus d'intégration sont construits simplement en déposant ces composants et connecteurs sur l'espace de travail, en les reliant, et en définissant leurs propriétés (la plupart étant héritées des métadonnées).

i. Correspondances

L'outil TALEND repose sur les correspondances ou « mappage » des données. Ces correspondances indiquent au système la provenance des données qui vont alimenter les schémas cibles lors de l'exécution des jobs, ainsi que le type de ces données. A l'exécution du job, TALEND déduit de ces correspondances les transformations que le système doit appliquer sur les données sources afin de les intégrer au schéma cible. Chaque

correspondance peut être vue comme un lien entre une ou plusieurs colonnes des schémas sources et une colonne schéma cible. Cette section présente l'ensemble des correspondances possibles avec TALEND.

Il existe plusieurs types de correspondances :

La **correspondance atomique** est la plus simple puisqu'elle associe une colonne du schéma source à une colonne du schéma cible (Figure 19).

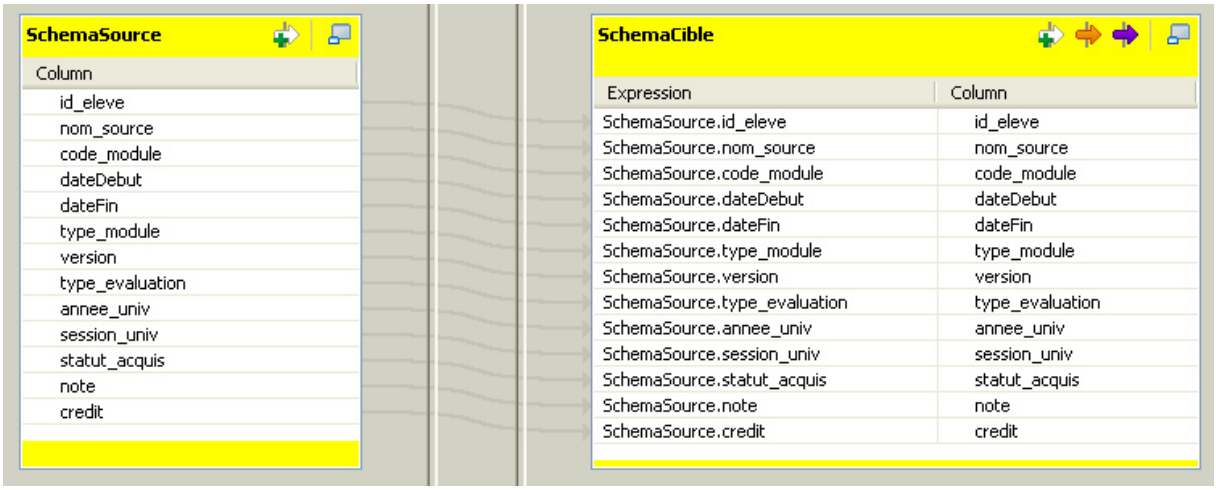


Figure 19 : Correspondance atomique

La **correspondance de type calcul** va appliquer un calcul sur une valeur source avant de copier le résultat obtenu dans la colonne cible (Figure 20).

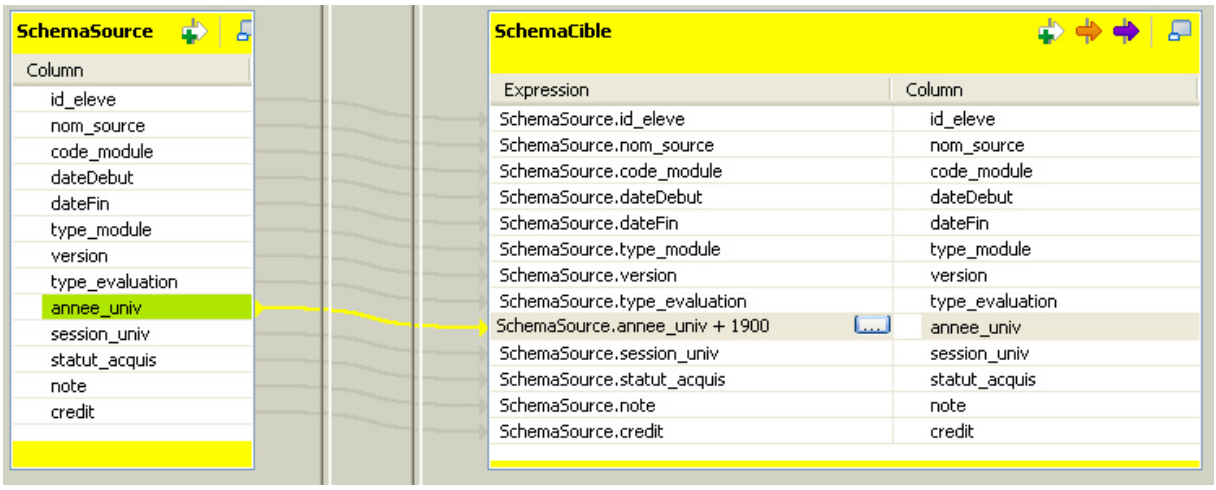


Figure 20 : Correspondance de type calcul

La **correspondance de type « valeur fixe »** est une correspondance qui va affecter pour chaque enregistrement une même valeur fixe à la colonne cible (Figure 21).

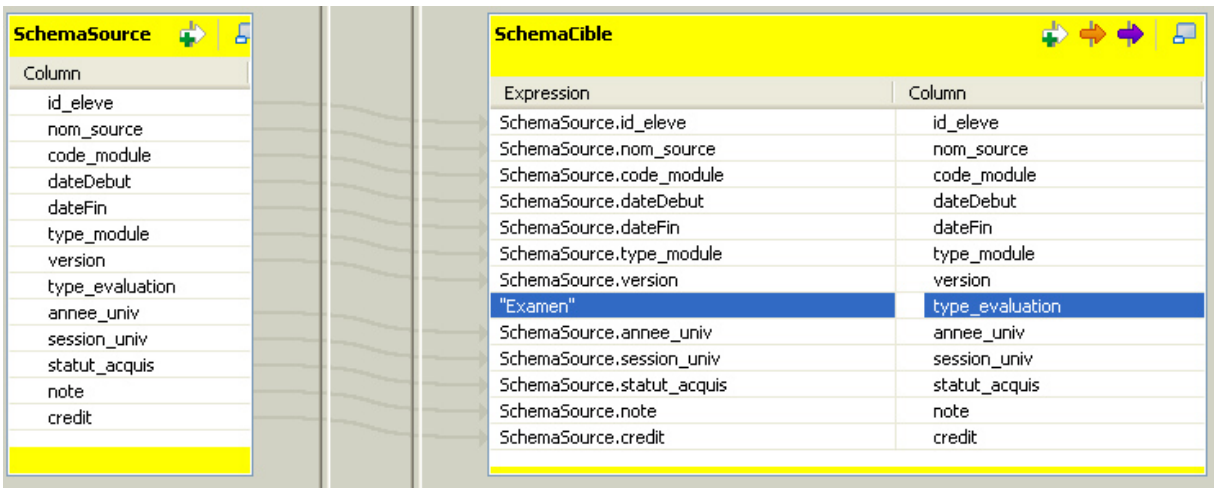


Figure 21 : Correspondance de type valeur fixe

La **correspondance de type transtypage** est une correspondance qui va effectuer une modification du type des valeurs copiées de la colonne source vers la colonne cible (Figure 22).

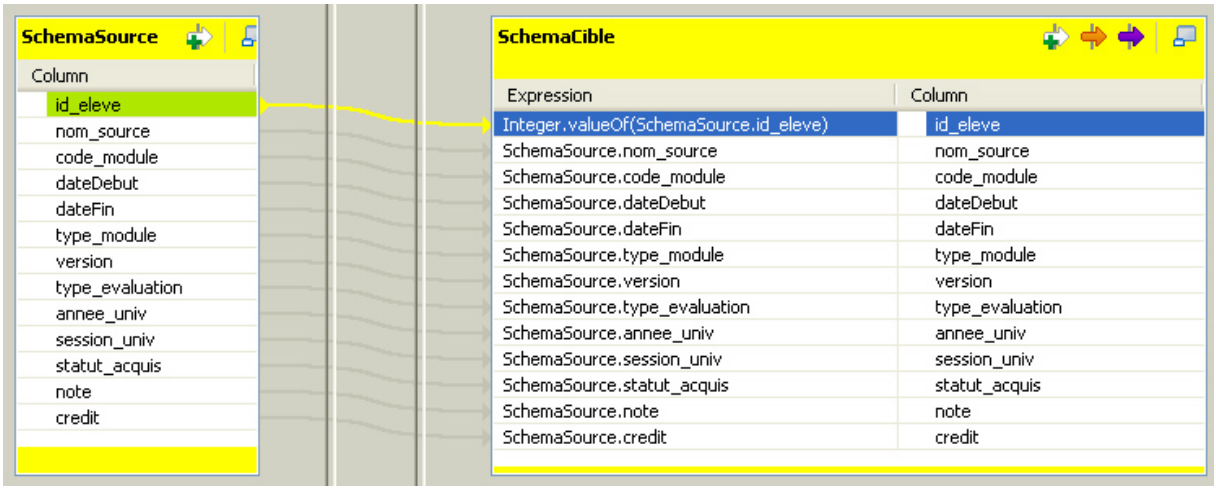


Figure 22 : Correspondance de type transtypage

La **correspondance du type référence** met en jeu deux tables sources où l'une des colonnes est définie comme clé étrangère référençant une des colonnes de l'autre table. A l'exécution du job, la valeur copiée dans la base cible sera celle de l'enregistrement référencé par la clé étrangère. En d'autres termes, une table source fait référence à une autre table source (Figure 23).

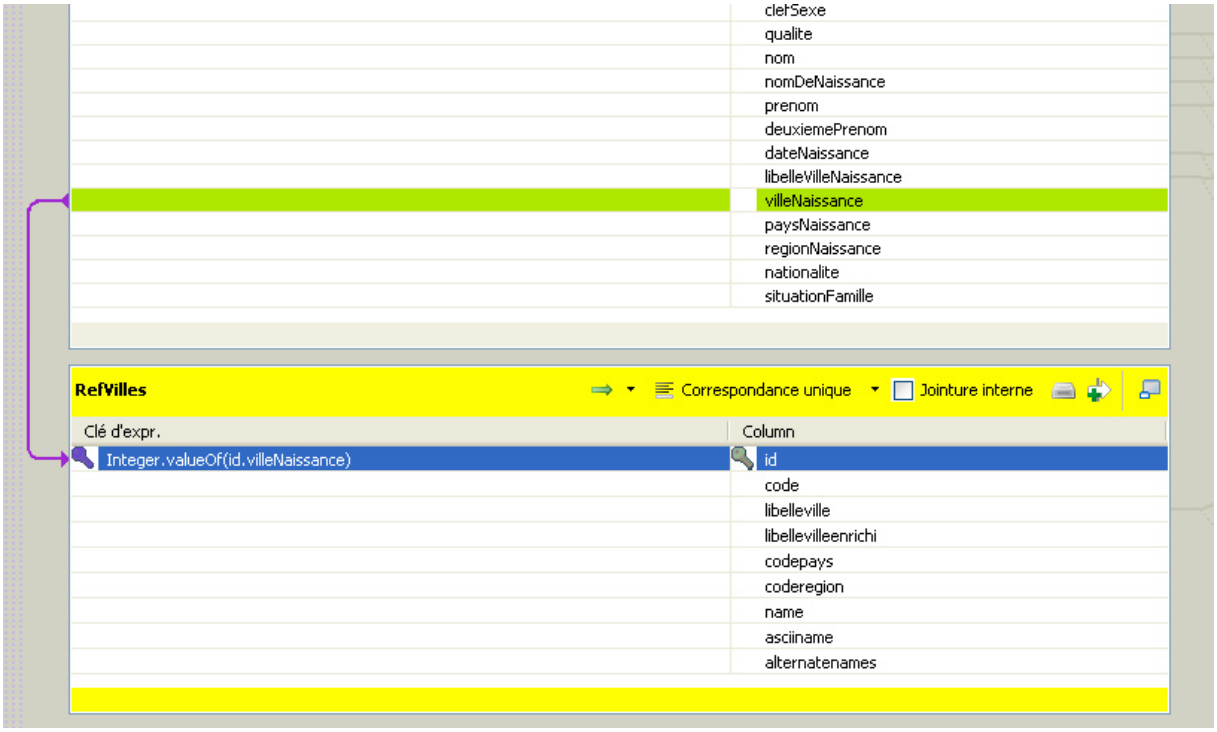


Figure 23 : Correspondances de type référence

La **correspondance de type concaténation** permet de concaténer les valeurs de plusieurs champs d'un enregistrement de la base source et de copier la chaîne ainsi obtenue dans un champ de la base cible (Figure 24).

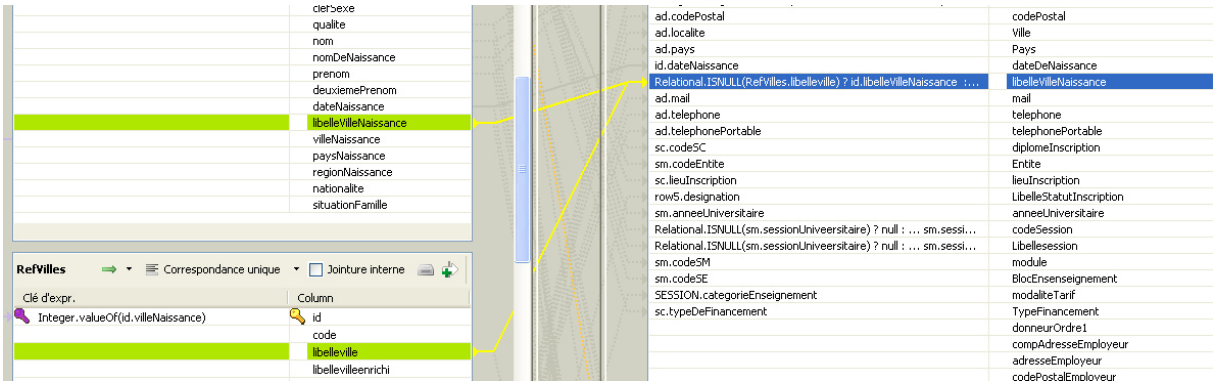


Figure 24 : Correspondance de type concaténation

ii. *Constructeur d'expression*

Pour certains jobs, il est nécessaire de rédiger du code afin de paramétrer les composants. Dans la vue Component de certains composants, une fenêtre « Expression Builder » peut nous aider à la construction de ce code (en Java ou Perl). Par exemple il est possible d'utiliser une méthode sur un attribut (Figure 25).

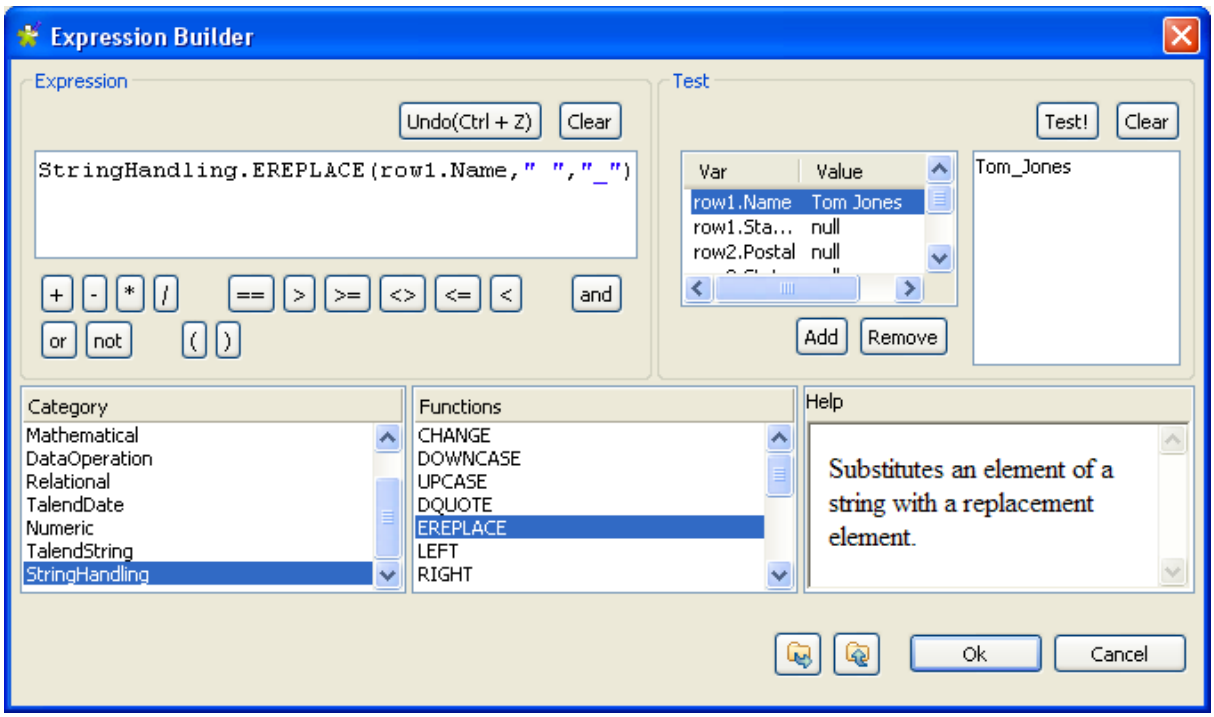


Figure 25 : Le constructeur d'expression

5) La création de composant

Dans Talend il est possible de créer des composants réalisant des tâches répondant à un besoin spécifique. Ces composants ne sont pas des classes java mais des patrons de programme java appelés « snippet ». Chaque programme est composé de trois sections (dans certain cas une seul) : un bloc de début, un programme principal et un bloc de fin. Les blocs de début et de fin ne sont exécutés qu'une seule fois alors que la partie principale peut, elle, être par exemple exécutée dans une boucle Figure 26.

Start code
```
System.out.println("I am the begin section");
for (int myvar=0;myvar<10;myvar++)
{
```

Main code
```
// here is the main part of the component,
// a piece of code executed in the row
// loop
System.out.println("I am the main section and myvas is "+myvar);
```

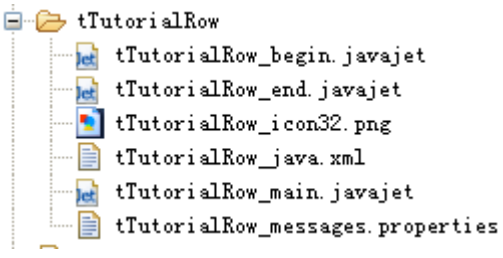
End code
```
// end of the component, outside/closing the loop
}
System.out.println("I am the end section");
```

Figure 26 : Description d'un composant

Chaque composant est un ensemble de 6 fichiers (Figure 26) déposé dans le répertoire des composants de Talend :

- <component>_<language>.xml : la description du composant dans une langue ;
- <component>_icon32.png : l'icône du composant;
- <component>_messages.properties : les labels affichés ;
- <component>_begin.<language>jet : Le code de début
- <component>_main.<language>jet : la partie principale ;
- <component>_end.<language>jet : le code de fin

Figure 27 : Arborescence des fichiers d'un composant



Bien entendu, Il est possible de paramétrer la génération du code final à l'aide de fichier XML. Ces fichiers sont lus et leurs contenus sont injectés dans les « templates » afin de produire le code java qui sera exécuté.

6) Génération de code

La Figure 28 représente le processus permettant de générer la classe qui sera effectivement exécutée. Chaque composant possède un certain nombre de paramètres renseignés par le programmeur. Ces paramètres permettent de personnaliser les unités de code java formellement définies (« snippet » ou « template ») puis générées en code java utilisable et incorporées dans un module plus large (le job), la classe java qui sera compilée et exécutée.

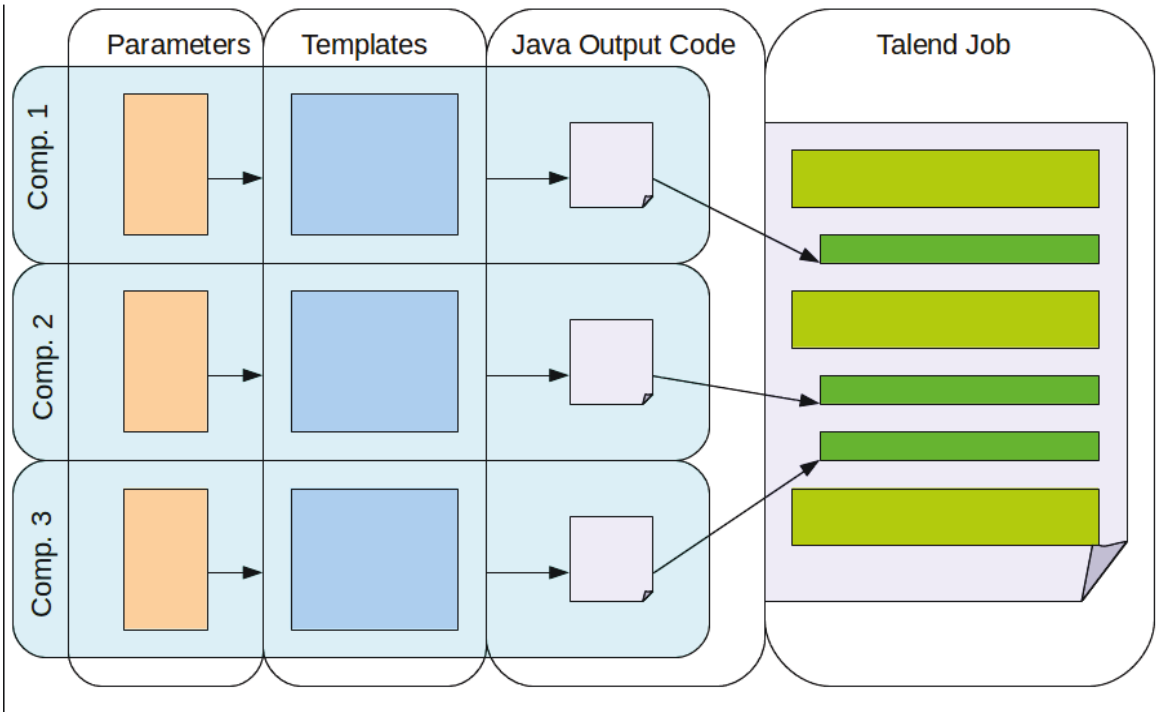


Figure 28 : Processus de génération de code Java

6. Synthèse

Dans ce chapitre nous venons de poser les briques qui nous permettent d'appréhender les notions liées aux bases de données décisionnelles et à l'alimentation de celle-ci. Un projet de création d'un entrepôt de données repose sur un certain nombre de points clés. Nous avons vu aussi de quelle manière un entrepôt de données peut être vu comme un ensemble de magasin de données. De plus nous avons vu plus particulièrement de quelle manière l'ETL va nous permettre d'alimenter un entrepôt à partir des bases de données de production.

Nous allons maintenant aborder les problématiques liées à la migration des données ainsi qu'à leur mise en qualité. Durant ce processus le rôle de l'ETL et de celui qui le programme sont importants, afin de garantir la qualité des données, de les travailler, de les nettoyer, et de vérifier leur intégrité avant de les intégrer à l'entrepôt. Ce sont tous ces processus que je propose de développer dans ce troisième chapitre.

III. Etude de la reprise de données

Dans ce chapitre je présente les différentes problématiques à prendre en compte lors de la reprise de données de scolarité du CEP, de l'Intec et de l'école d'ingénieur. En effet j'ai effectué une étude préalable à l'aide d'un « business model » afin de mieux cerner le périmètre du projet. Je me suis attaché dans un premier temps à décrire de manière générale le processus permettant d'effectuer la reprise de données puis dans un second temps d'expliquer les méthodes adoptées lors de ces différentes phases de ce processus de reprise. Enfin j'y décris de manière plus précise les problématiques spécifiques liées à la réconciliation de données.

1. Business model de la reprise de données des applications de scolarité

Afin de préciser le cheminement suivi par les données depuis leur extraction de la source à leur intégration dans l'entrepôt, j'ai d'abord proposé un « business model » (Figure 29) décrivant l'ensemble des processus constituant la reprise de données. L'alimentation de l'entrepôt se base sur différentes sources de données allant de l'utilisation de simples fichiers textes à différents types de bases de données.

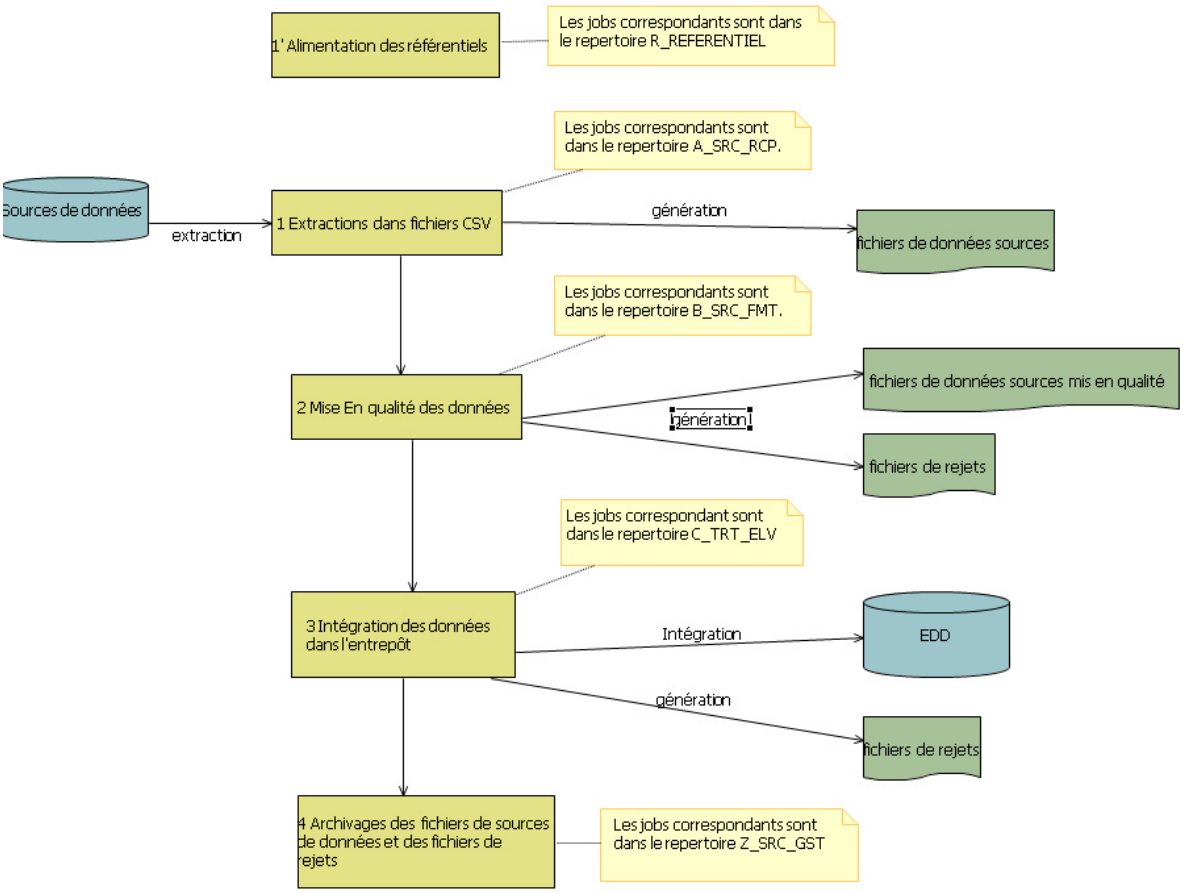


Figure 29 : Business model reprise de données

Dans tous les cas, le cheminement utilisé pour amener l'information dans l'entrepôt est le même :

1) Alimentation des référentiels

Les référentiels sont fournis sous forme de fichier plat par l'équipe Logica en charge du paramétrage de SAP. Ils sont déposés dans un répertoire destiné à cet effet puis les jobs Talend alimentent les tables correspondantes. Lors d'une mise à jour des tables de référence, les jobs sont rejoués avec un nouveau jeu de fichiers extraits de SAP.

2) Extraction des données

Les données sources sont reçues soit directement sous forme de fichier, soit sont récupérées dans les bases de données des applications sources par le biais de vue ou directement en interrogeant les bases.

3) Intégration de ces données dans l'ODS

Cette étape se base sur les fichiers sources de données déposés dans les répertoires précisés à l'étape précédente pour les injecter dans l'ODS.

4) Mise en qualité des données dans l'ODS

Cette étape consiste à préparer la donnée avant de l'intégrer dans l'entrepôt. Cela revient à utiliser les valeurs des référentiels et à faire le tri entre les données qui peuvent être intégrées à l'entrepôt de celles qui ne peuvent pas l'être. Cette étape n'impacte que la base de données ODS.

5) Extraction des données mise en qualité dans l'ODS dans des fichiers destinés à être intégrés dans l'entrepôt de données

Cette étape extrait de l'ODS les données destinées à être intégrées dans l'entrepôt, c'est-à-dire celles qui sont « validées » pour être stockées dans l'entrepôt. Pour ce faire, on extrait de l'ODS dans des fichiers les données visées.

6) Extraction des données de rejets dans des fichiers

Au cours de cette étape sont extraites les données rejetées vers des fichiers. Cela nous permet notamment de corriger les données ou dans les sources ou bien d'adapter les programmes de reprise afin de prendre en compte des cas particulier.

7) Intégration des données dans l'entrepôt

Lors de cette étape les données extraites de l'ODS sont effectivement intégrées à l'entrepôt. Les jobs traitent l'insertion des données (toutes les valeurs de référentiels ont été valorisées lors de la mise en qualité) dans l'entrepôt avec leur suivi de version. En outre, il faut contrôler que les données injectées ne sont pas déjà présentes dans l'entrepôt afin de savoir si l'on crée une première version ou si l'on en ajoute une nouvelle. Dans le cas d'un ajout il faut tenir compte de la priorité des sources. Il faut évidemment s'assurer de ne pas insérer un enregistrement identique à celui déjà présent dans l'entrepôt. Lors de ces traitements des fichiers temporaires sont générés afin de stocker temporairement des enregistrements en vue de traitements particuliers.

8) Extraction des données de l'entrepôt dans des fichiers textes en direction de SAP

Lors de cette étape, en se basant sur les spécifications fonctionnelles fournit par l'équipe Logica, sont générés les fichiers texte à destination de SAP. Ces fichiers sont au format UTF8 avec en caractères de fin de ligne : CR et LF et ne doivent pas dépasser 50 000 lignes.

Ces étapes nécessitent l'utilisation de répertoire pour stocker les fichiers générés/utilisés tout au long du processus décrit. Ces répertoires sont définis dans des variables de contexte, elles-mêmes définies dans un fichier de contexte afin de permettre de passer facilement de l'environnement de développement à celui de production.

2. Etapes de la reprise de données

Durant le projet, ma principale mission fut de proposer des méthodes de détection et de correction des erreurs introduites lors de la saisie des informations par les utilisateurs des différentes applications.

Le processus de mise en qualité se décompose en plusieurs traitements intermédiaires (Figure 30) garantissant un niveau de qualité de donnée acceptable. J'ai utilisé L'ETL Talend présenté en paragraphe 2.5 pour réaliser les opérations suivantes. Je propose dans un premier temps d'expliquer les méthodes adoptées lors de ces différentes phases.

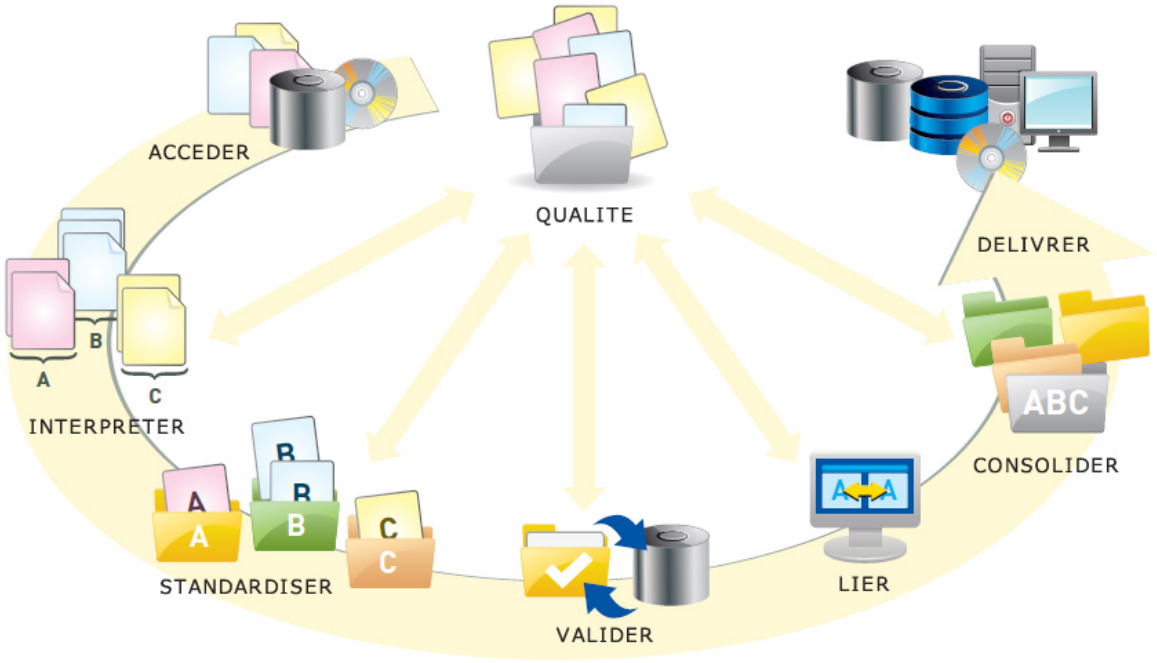


Figure 30 : Processus de mise ne qualité des données

1) Accéder aux données.

Le but de ce processus est de récupérer les données de production. Généralement stockées dans plusieurs bases de données ou des fichiers, seuls les gestionnaires et les utilisateurs de l'application peuvent nous guider afin de me permettre de mener à bien cette étape. Une première étude m'a permis d'avoir la liste, la structure et la localisation de chaque donnée à reprendre.

Deux principales approches permettent un accès unifié à des sources de données hétérogènes l'une virtuelle souvent appelée approche par médiateur et une matérialisée appelée approche par entrepôt. Dans l'approche virtuelle, basée sur une hiérarchie de médiateurs correspondant à des vues virtuelles au-dessus des extracteurs, les données ne sont stockées que dans leur source d'origine. En, revanche, avec l'approche matérialisée les données sont effectivement extraites, nettoyées, intégrées et stockées dans un entrepôt. Les requêtes sont posées directement sur les données de l'entrepôt, un des problèmes majeurs à résoudre dans cette approche est celui de la répercussion dans l'entrepôt des mises à jour effectuées sur les sources. Les outils de type ETL permettent d'intégrer des données en s'appuyant sur une approche matérialisée.

La première étape fut de répertorier les différentes sources de données englobées par le champ fonctionnel du projet. Pour chaque source j'ai mené des entretiens avec les gestionnaires des différentes applications concernées afin de constituer d'une « bibliothèque » de requêtes permettant d'interroger et d'extraire les données pertinentes.

Puis nous avons établi des règles permettant d'identifier pour chaque donnée extraite:

- la source de la donnée (GRAFIC EISCOL UTINTEC).
- la clé constituant l'identifiant unique de la donnée dans la source
- la date d'extraction de la donnée

Les données sont extraites vers une zone de préparation des données permettant la mise en qualité avant l'intégration dans l'entrepôt. Dans notre cas, cette zone appelée ODS est une base MySQL constituée de table temporaire. Ce procédé permet d'étanchéifier les zones opérationnelle (d'où viennent les données) et décisionnelle.

2) Interpréter les données

Interpréter les données implique une très bonne connaissance des sources de données, afin de connaître la structure et la sémantique de chaque information. Encore une fois seule les utilisateurs fonctionnels peuvent nous aider lors de cette étape.

Toutes les données sources n'ont pas systématiquement d'intérêt pour la base de données décisionnelle. Le processus d'interprétation a pour mission de filtrer les données utiles, de s'assurer de la présence effective de la donnée dans la source et de vérifier la pertinence des données entreposées.

Les données « floues » (i.e. dont il n'est pas possible de décrire la structure et/ou de fournir la définition précise) ne sont pas retenues. Elles seront prises en compte à mesure lorsque leur sémantique apparaîtra clairement dans les spécifications fonctionnelles détaillées.

3) Standardiser des sources de données hétérogènes

Standardiser les sources de données consiste à adapter la représentation (le schéma) d'un objet commun à chaque application en une représentation commune (le schéma cible). L'usage de schémas intermédiaires permet de conserver une donnée sous sa forme source, de la traiter et de conserver sa correspondance sous sa forme cible. Il a donc fallu définir des formats d'entrée pour toutes les données à intégrer dans l'entrepôt. La Figure 31 page 63 présente un exemple de modèle de données final adopté pour tous les dossiers administratifs**.** L'usage Schéma générique permet ensuite de standardiser les traitements réalisés sur les objets communs à chaque source. (Par exemple lors de la consolidation et la validation d'un objet personne).

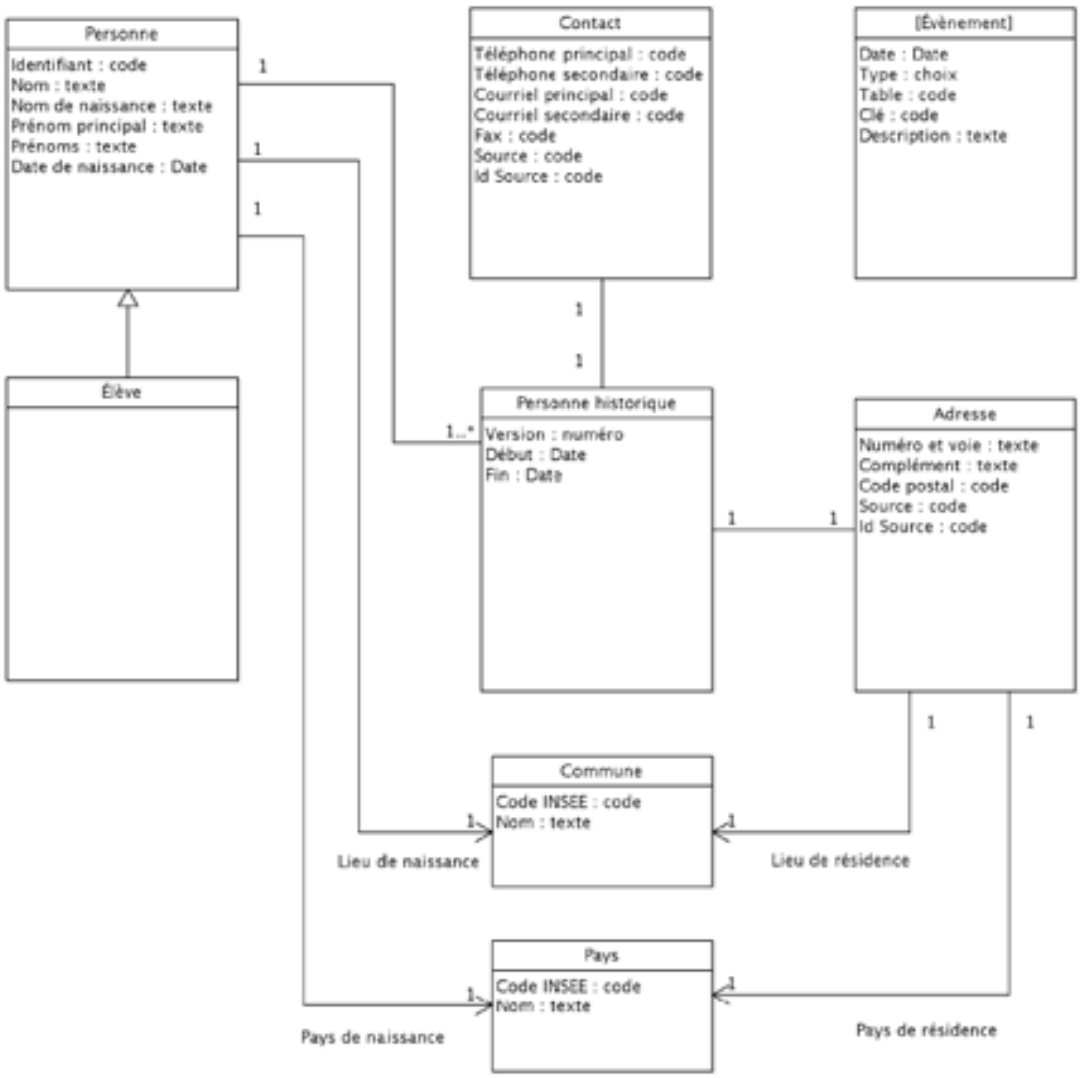


Figure 31 : Modèle de donnée du dossier administratif d'un élève

N.B : Ce modèle de données représente la vision fonctionnelle de l'EDD. Il est la spécification de ce que les applications utilisatrices peuvent trouver dans l'EDD. Ce n'est donc pas le schéma physique des tables. L'EDD physique comporte des informations supplémentaires, nécessaires aux traitements d'alimentation et d'administration des données.

4) Valider des données

Certaines données doivent être validées d'un point de vue syntaxique et sémantique. Par exemple l'adresse de courrier électronique doit répondre à une syntaxe spécifique. Le numéro de téléphone ne doit pas comporter de caractère alphabétique et les villes habituellement en saisie libre doivent désormais correspondre à des référentiels. La mise en place et l'utilisation de référentiel dans l'entrepôt permet de valider les données entre les sources. Certains référentiels utilisés par SAP sont importés dans l'entrepôt puis utilisés lors de la mise en qualité des données. C'est le cas notamment lors des traitements réalisés sur les villes et les pays. Lors de la reprise des dossiers pédagogiques les nomenclatures permettant de désigner les unités d'enseignement ont été systématiquement validées à l'aide de l'offre de formation fournit par la DNF. Les différents symptômes rencontrés lors de la validation des données sont détaillés dans le paragraphe 3.3 traitant de la réconciliation de données**.** Les instances dont certains attributs essentiels ne se pas validés sont rejetées et font l'objet d'un traitement particulier.

5) Gérer les rejets

J'ai mis en place un système de traçabilité nous permettant d'appliquer un ou plusieurs syndromes à une instance. Par exemple si une donnée importante, comme les villes de naissance entrant dans le calcul du numéro INE, n'est pas validée, il est nécessaire de pouvoir conserver les enregistrements rejetés afin de ne pas perdre de données entre ce qui existe dans les applications sources et ce qui est centralisé dans l'entrepôt. Actuellement les données à rejeter sont définies lors de la mise en qualité dans l'ODS. Plusieurs solutions de stockage ont été envisagées:

- Les extraire dans des fichiers
- Les stocker dans l'entrepôt
- Les stocker dans une base dédiée

Les rejets sont extraits dans des fichiers textes et sont destinés à prendre la mesure des volumes rejetés ainsi qu'à adapter les programmes à certains cas particuliers. Chaque

enregistrement y est consigné et accompagné d'une explication sur le motif du rejet permettant d'améliorer les programmes de reprise en fonction des erreurs rencontrées.

6) Lier les données

Dans de nombreux cas les données enregistrées concernent le même objet (par exemple, un auditeur, les diplômes) résidant dans les systèmes de sources multiples. Ces dossiers doivent d'abord être liés et consolidés avant d'être chargés dans l'entrepôt de données. L'intégration avec des logiciels de qualité de données est souvent le seul moyen réaliste de faire correspondre ces enregistrements. Chaque système peut contenir sa propre représentation d'une même donnée. La plupart du temps il s'agit de repérer les doublons entre les sources notamment lors de la reprise des dossiers administratifs de l'élève. Les problématiques rencontrées lors de la recherche de doublons sont détaillées dans le 3.3.5**.**

7) Consolider les données

Dans un premier temps nous avons défini des critères d'unicité communs à chaque source permettant d'apparier certaine données. Pour un auditeur le nom, le prénom, ainsi que la date et le lieu de naissance servent de clé lors de la consolidation des données élèves. Par la suite, il faut filtrer et trier chaque donnée relative à l'élève (comme son adresse ou son passé pédagogique) présent dans les différentes sources afin de les consolider dans un seul dossier. Les problématiques rencontrées lors de la consolidation des données sont détaillées dans le 3.3.3**.**

8) Intégrer les données dans l'entrepôt de données

Lors du chargement des données dans l'entrepôt, il a fallu définir un protocole tenant compte de la reprise incrémentale. Si la donnée évolue dans le temps il faut mettre en place un système de gestion de version des données. De plus, nous devions conserver une correspondance entre la donnée dans la source et la donnée dans l'entrepôt en conservant l'identifiant dans chaque source et l'identifiant dans l'entrepôt d'une donnée.

3. Problématiques liées à la réconciliation de données

1) Enjeux

Afin de mener au mieux le processus de reprise de données il m'a été demandé de proposer une méthode de reprise en détaillant plus particulièrement les processus mis en œuvre lors de la mise en qualité des données extraites des applications sources. En effet, j'ai dû proposer un mode opératoire permettant de rapprocher les différentes sources de données avant de les intégrer dans l'entrepôt. Cette section détaille les problèmes rencontrés lors de la mise en qualité des données.

2) Problématiques

Le "nettoyage des données" appelé "mise de la qualité des données" a pour but de résoudre les problèmes de cohérence des données (8). Ces incohérences peuvent être locales à un enregistrement par exemple lors d'une erreur de frappe dans la saisie d'un champ ou locales à une source par exemple lorsqu'un auditeur possède plusieurs dossiers administratifs dans la même source. De plus, des incohérences peuvent aussi survenir lors de la mise en commun des différentes sources de donnée lorsque par exemple une personne présente dans plusieurs sources possède une adresse différente dans chacune des sources.

Lors de l'étude préliminaire des données un certain nombre de symptômes ont été identifiés et répertoriés :

- la présence de données fausses dès leur saisie ;
- la persistance de données obsolètes ;
- la confrontation de données exactes, sémantiquement identiques, mais syntaxiquement différentes.

Le problème des adresses et des noms de clients est un des problèmes pratiques les plus cruciaux des entrepôts de données, cette donnée étant d'une part de la plus haute importance, et d'autre part à la fois subjective, sans format fixe et volatile.

3) Détection, correction et nettoyage des données

Le nettoyage de données par transformation fait partie des stratégies d'amélioration de la qualité des données (9) (10) qui consiste à choisir et appliquer des transformations sur des jeux de données pour résoudre différents problèmes de format et d'incohérence, soit au sein d'une même source de données, soit entre plusieurs sources de données hétérogènes.

Les problèmes candidats au nettoyage peuvent être répartis en problèmes mono-sources et multi-sources, au niveau du schéma ou des instances. Nous verrons dans les tableaux d'exemples suivants que la classification d'une erreur dépend essentiellement des contraintes qui auront pu être définies. Le Tableau 3 présente des erreurs détectées au moyen de contrôles de cohérence sur les sources de données INTEC et CEP.

Tableau 3 : Exemple de violation de contraintes au niveau schéma d'une source

niveau	Symptôme	Données	Description
Attribut	Valeur incohérente	Elève101 (date naissance = 31/02/1976)	Valeur erronée
Enregistrement	Violation des contraintes de cohérence entre attributs	Adresse13 (Ville = Barcelone, Pays = France)	La contrainte entre ville est pays n'est pas satisfaite
Enregistrement	Violation de contrainte d'unicité	Elève22 (Nom = Leblanc, INE = 0G5DRJ001G2) Elève36 (Nom = Felicien, INE = 0G5DRJ001G2)	L'unicité pour le numéro INE n'est pas satisfaite
Source	Violation de contrainte d'intégrités référentielles	Activite (Catégorie socio-professionnelle = BIATOSS)	Valeur non définie pour la catégorie socio-professionnelle

Le Tableau 4 présente des exemples d'erreurs au niveau instance. La détection de ces erreurs peut être effectuée avec :

- des contrôles de cohérence non référencés comme contraintes d'intégrité de la base;
- des tests de vraisemblance ;
- au moyen de critères empiriques établis lors de la phase d'analyse des données.

Tableau 4 : Exemples de violation de contraintes au niveau « instance » d'une source

niveau	Symptôme	Données	description
Attributs	Valeur manquantes	Elève (Email =xxxxxx)	Valeur non disponible
	Erreurs typographiques	Adresse13 (Ville = Mareseille)	Erreur phonétique
	Abréviations	Adresse13 (voie =Bd de Strasbourg)	
	Valeurs imbriquées	Elève (deuxième prénom = Jean, Charles, Albert)	Valeur multiples saisies dans un seul attribut
	Erreurs d'attributs	Adresse13 (Ville = 85100)	
Enregistrement	Violation de dépendances entre attributs	Adresse13 (Ville = Toulouse, code postal = 85100)	Les valeurs ne correspondent pas
Type d'enregistrement	Transpositions	Elève11 (nom = Dupond, prénom = jean) Elève42 (nom = Dupont, prénom =Jean)	Problème lié à la saisie libre
	Doublons	Elève11 (nom = Dupond, INE= 0G5DRJ001S4) Elève42 (nom = Dupont, INE = 0G5DRJ001S4)	Même élève entré deux fois avec un nom différent
	Enregistrements contradictoires	Elève53 (nom = Armand, date naissance = 16/09/1965) Elève78 (nom = Armand, date naissance = 06/09/1965)	Plusieurs dates de naissance sont possibles pour l'élève Armand
Source	Mauvais référencement	Activité (Catégorie socio-professionnelle = Cadre)	La catégorie socio-professionnelle existe mais elle est fausse pour cet enregistrement

Les problèmes multi-sources au niveau du schéma de représentation des données peuvent se scinder en deux catégories:

- les conflits de noms qui surviennent lorsqu'un même nom est donné à deux objets différents (homonymes) dans chacune des sources, ou lorsque des noms différents sont donnés au même objet (synonymes) ;
- les conflits de structure qui peuvent être très variés et proviennent de représentations différentes d'un même objet dans les différentes sources.

Par exemple, pour représenter le nom d'une personne, l'INTEC utilise la terminologie *nom patronymique* et *nom épouse*, le CEP *nom de naissance* et *nom d'usage* et l'école d'ingénieur *nom (pour le nom de naissance)* et *nom d'usage* ce qui nous a posé beaucoup de problème lors de la détection de doublons.

Les problèmes multi-sources au niveau « instance » (Tableau 5) peuvent être dus à des représentations différentes des données, à des différences d'agrégation (de regroupement), à l'évolution des usages de saisie et de description au cours du temps. La résolution de ces problèmes implique l'intégration des deux schémas, ainsi que le nettoyage de chaque source.

Tableau 5: Problèmes multi-sources au niveau instance

Symptôme	Description
différence de codages	"M/F" ou "1/2" pour le sexe d'une personne
différence d'unités	un prix en Francs dans une source et en € dans l'autre
différence de granularité	un nombre d'heures travaillées par semaine dans une source et par mois dans une autre
différence de fraîcheur	Un âge de 25 ans dans une source et de 26 ans dans une autre mise à jour plus récemment
imprécision	un poids de 54 kg dans une source et de 54,2 dans une autre
utilisation de synonymes	"sans emploi" et "chômeur"
différentes façons d'écrire la même donnée dans un texte libre	une même adresse peut être "4, av. du Gal. De Gaulle" dans une source et "4, avenue du général de Gaulle" dans une autre
différence de contenu dans un texte libre	une adresse contenant dans une source le nom du destinataire et pas dans l'autre

4) Résolution des problèmes spécifiques

Dans cette section, nous nous intéressons plus particulièrement au cas des doublons, des valeurs manquantes et des valeurs aberrantes en s'appuyant, dans une certaine limite, sur certaine méthodes et techniques issues des travaux de recherche dans le domaine.

i. Elimination des doublons

Dans le cas de l'intégration de plusieurs sources d'information (en l'occurrence l'intégration de bases de données relationnelles) dans un entrepôt de données, il est nécessaire d'associer plusieurs tables au moyen de jointures pour lesquelles on ne dispose pas de clés communes exactes. Lors de la recherche de doublons sur une seule table, il est nécessaire de procéder par auto jointure : bien que les clés puissent identifier de façon unique chaque enregistrement de la table, plusieurs enregistrements peuvent pourtant décrire la même réalité ; par exemple deux enregistrements peuvent décrire la même personne avec des dates de naissances différentes. Ainsi pour détecter les doublons, Koundas (11) recommande une technique de jointure approximative. D'après notre exemple, il est nécessaire d'apparier les données entre les tables pour pouvoir renseigner tous les champs de l'entrepôt. Les noms et adresses sont décrits de différentes façons (par exemple, « Avenue du Général de Gaulle » ou « av. Gal Gaulle ») et il peut être difficile de faire l'appariement sur les noms ou adresses. Si, en revanche, le numéro INE est le même, on pourra supposer qu'il s'agit bien de la même personne, c'est pourquoi il s'avère nécessaire d'abord de standardiser certains attributs comme les adresses puis d'examiner les informations qui corroborent ou non une hypothèse d'appariement sur l'ensemble des attributs disponibles. Parfois très spécifique à l'application, la technique de jointure approximative consiste à regrouper et trier les enregistrements par « paquets » (ou groupes) selon une fonction de hachage sur les valeurs d'un ou plusieurs attributs (par exemple, utilisant les premières lettres ou les consonnes des noms propres). Les enregistrements qui se trouvent dans les mêmes groupes sont candidats à l'appariement et, pour chaque paire de candidats, une distance de similarité est calculée. Seules les paires de plus haut score sont effectivement appariées ou assimilées à des doublons.

La méthode classique de jointure approximative et de détection de doublons est présentée dans le Tableau 6.

Tableau 6 : Méthode de recherche de doublons

Méthode générique de recherche des doublons
1. Prétraitement des données (standardisation des attributs, des abréviations, structuration des adresses, etc.). 2. Choix d'une fonction permettant de réduire l'espace de recherche par : • tri ou hachage selon une clé ; • examen par fenêtrage multiple. 3. Choix d'une fonction de comparaison permettant d'exprimer la distance entre les paires telle que : • identité stricte, distance simple ou complexe ; • distance pondérée par la fréquence ou dirigée par des règles ; • distance d'édition, distance de Jaro, Jaro-Winkler ; • comparaison N-gram, Q-gram ; • Soundex ; • TF-IDF ; • coefficient de Jaccard, etc. 4. Choix d'un modèle de décision : • méthodes probabilistes : avec/sans ensemble d'apprentissage ; • méthodes basées sur des règles et connaissances du domaine. 5. Vérification de l'efficacité de la méthode.

Pour des domaines d'attributs textuels, l'appariement des chaînes de caractères peut être calculé par une distance comptabilisant le nombre d'opérations d'édition (telles que l'ajout, la suppression d'un caractère ou le changement de lettre) nécessaires pour transformer une chaîne de caractères en une autre. Par exemple, « SRH » et « RH » ont une distance d'édition de 1. Les chaînes de caractères dont la distance d'édition est inférieure à un seuil fixé seront alors appariées. L'ensemble des algorithmes d'appariement de chaînes de caractères est détaillé dans l'ouvrage de G.Navarro (12).

ii. *Gérer les valeurs manquantes*

Les données manquantes sont classées en trois grandes catégories :

- les données manquantes complètement aléatoires : les enregistrements ayant une donnée manquante ne peuvent pas être distingués de ceux ayant une donnée renseignée. La probabilité qu'une donnée soit manquante ne dépend ni des valeurs des variables observées ni de la valeur non observée ;
- les données manquantes aléatoires ou ignorables : le fait d'avoir une donnée manquante dépend d'autres caractéristiques observées, mais pas de la valeur manquante qui aurait pu être renseignée. La probabilité qu'une donnée soit manquante dépend des valeurs des variables observées mais non de sa vraie valeur ;
- les données manquantes informatives non aléatoires et non ignorables : le fait d'avoir une donnée manquante n'est pas aléatoire, ne peut pas être déduit des autres variables et dépend de la valeur manquante (qui aurait pu être renseignée). La probabilité qu'une donnée soit manquante dépend de sa vraie valeur.

Selon leur type, le traitement des données manquantes peut se faire selon trois approches :

- en ne considérant que les données complètes : seuls les enregistrements ayant tous les attributs renseignés et complets sont analysés. Facile à mettre en œuvre, cette approche n'est praticable que sur un faible nombre de données manquantes complètement aléatoires ;
- en n'analysant que les données disponibles : Elle n'est valable que si les données manquantes sont complètement aléatoires;
- par imputation : la valeur manquante est remplacée par une valeur observée dans un autre enregistrement ayant les mêmes caractéristiques. Cette dernière méthode assez simple nécessite une métrique pour choisir les variables d'appariement et le calcul d'une distance. L'enregistrement le plus « proche » est alors retenu.

4. Synthèse

L'intégration de données est un vaste sujet étudié maintenant depuis plus d'une trentaine d'années. Les recherches sont nombreuses, tout comme les produits existants. Les frontières entre le domaine de l'intégration de données, de la mise en qualité et de l'informatique décisionnelle ne sont pas encore clairement définies. Dans ce chapitre nous avons pu voir de quelle manière la reprise de données a été menée en détaillant les différentes étapes de l'accès aux données sources jusqu'à l'intégration de celle-ci dans la base cible. De plus nous avons clairement identifié à quels types de contrainte et de problème j'ai été confronté lors de la mise en qualité des données. Le chapitre suivant décrit les solutions que j'ai pu apporter afin de les résoudre. Les quatre principaux postes à surveiller étaient :

- *l'hétérogénéité et la diversité des données multi-sources à intégrer ;*
- *le volume de données manipulées ;*
- *la richesse et la complexité des données ;*
- *la dynamique de rafraîchissement des données et leur qualité.*

IV. Reprise de données production

Ce chapitre présente les réalisations des principales tâches liées à l'intégration et à l'exploitation des données de l'entrepôt. J'y décris les développements ainsi que le paramétrage nécessaire permettant l'extraction des données, leur mise en qualité et leur insertion dans l'entrepôt. J'ai pris comme exemple la reprise des données administratives des auditeurs de l'INTEC et du CEP et la reprise des dossiers pédagogiques de l'INTEC et de l'école d'ingénieur. Par ailleurs j'y détaille les traitements particuliers permettant de gérer les enregistrements rejetés. Enfin j'y explique les processus permettant l'intégration des données dans l'entrepôt.

1. Reprise des dossiers administratifs du CEP et de l'INTEC

1) Extraction des sources du CEP

A partir des vues réalisées par le gestionnaire de l'application, j'ai pu extraire l'ensemble des auditeurs présent dans GRAFIC. Toutes les vues sont basées sur le même schéma défini lors de la conception détaillée. Certaines vue concernent les dossiers de l'année en cours, les dossiers déjà issus d'une reprise (lors de la mise en place de GRAFIC) et les dossiers concernant des inscriptions à des diplômes. Les données extraites sont ensuite « mappées » au schéma adopté dans l'entrepôt pour le processus de mise en qualité (Figure 32). Ce schéma possède des champs supplémentaires afin de stocker les données standardisées ainsi que des informations sur la qualité de l'enregistrement.

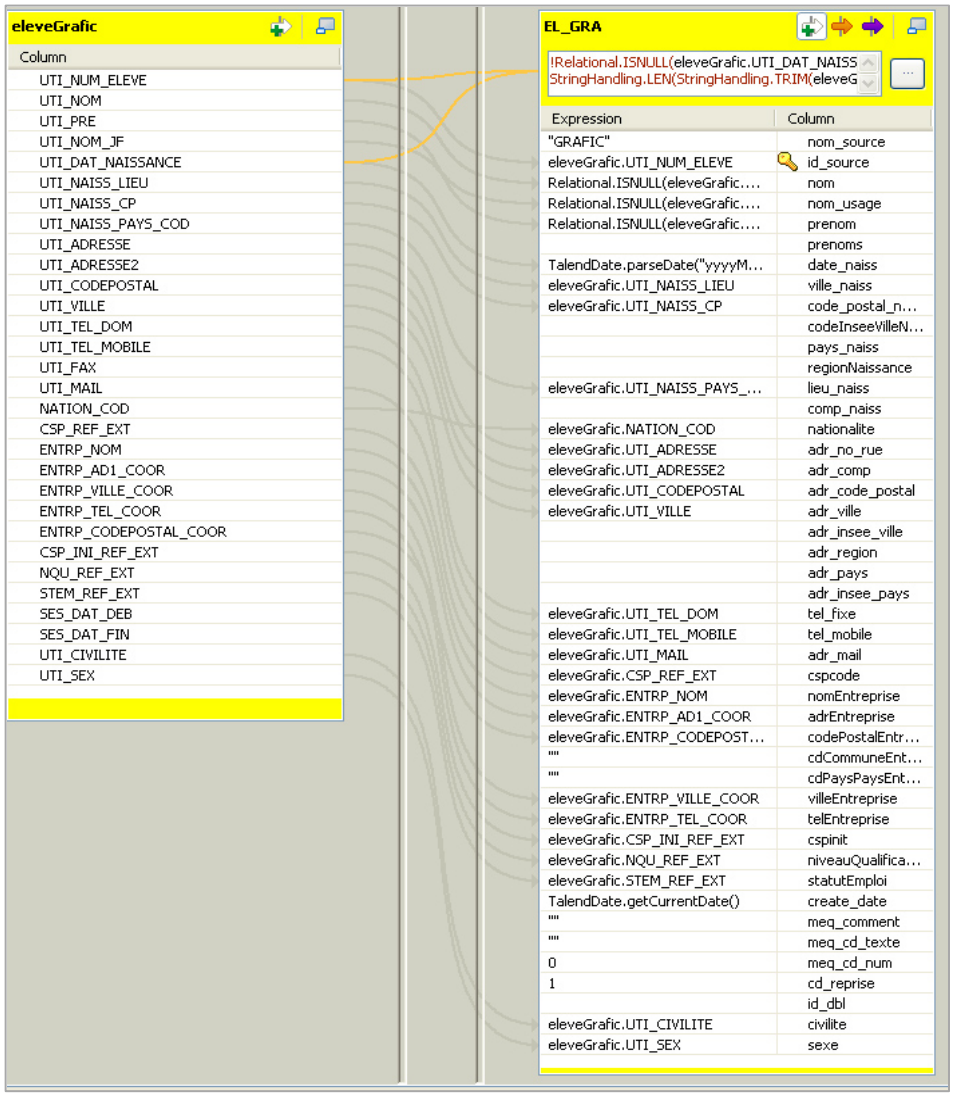


Figure 32 : Correspondance des données

2) Mise en qualité des critères d'unicité

Dans la Figure 33 on peut voir l'enchainement des différentes étapes de la mise en qualité des critères d'unicité des élèves repris de GRAFIC. D'abord le contrôle de la date de naissance, puis le contrôle des noms et prénoms. En cas de rejet, l'enregistrement est marqué d'un syndrome dont le fonctionnement est détaillé au chapitre 4.4.

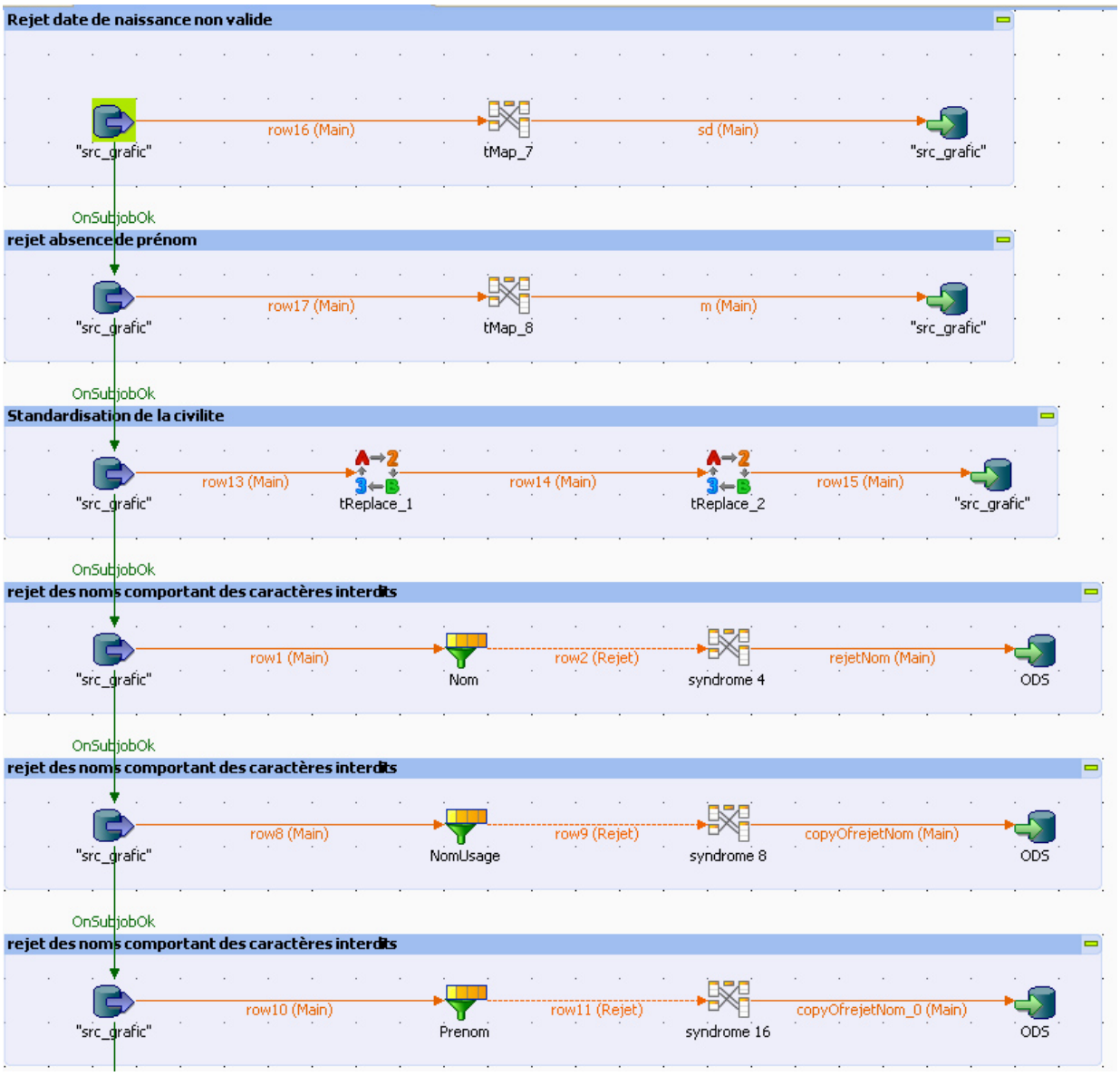


Figure 33 : Mise en qualité des critères d'unicité

La Figure 34 montre le principe de filtrage du champ nom si celui-ci ne respecte pas une expression régulière rejetant les caractères interdits. Les enregistrements non valides sont marqués du syndrome correspondant en vue d'un traitement spécifique.

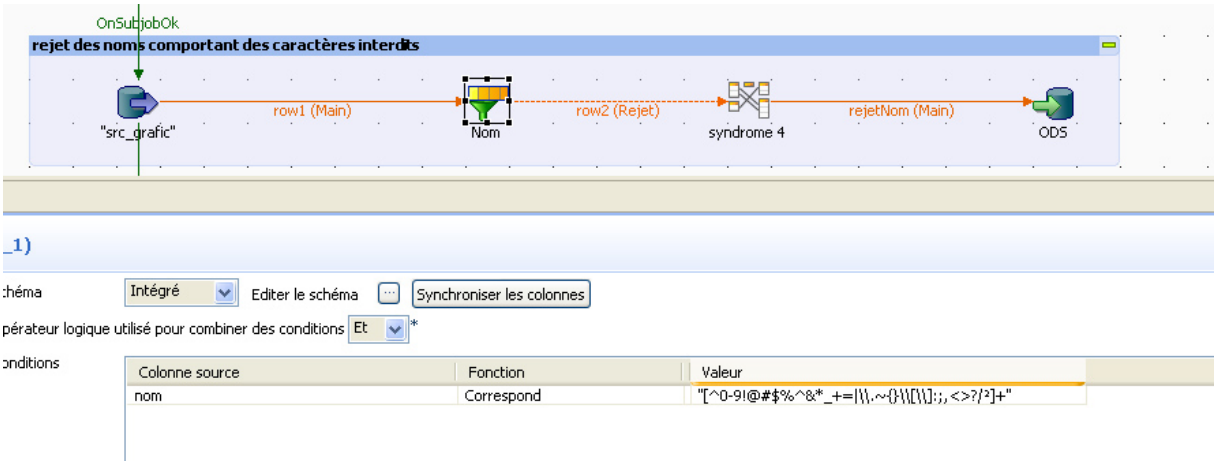


Figure 34 : Filtrage sur une expression régulière

La Figure 35 montre de quelle manière la source est dédoublée en comparant chaque enregistrement de la même source.

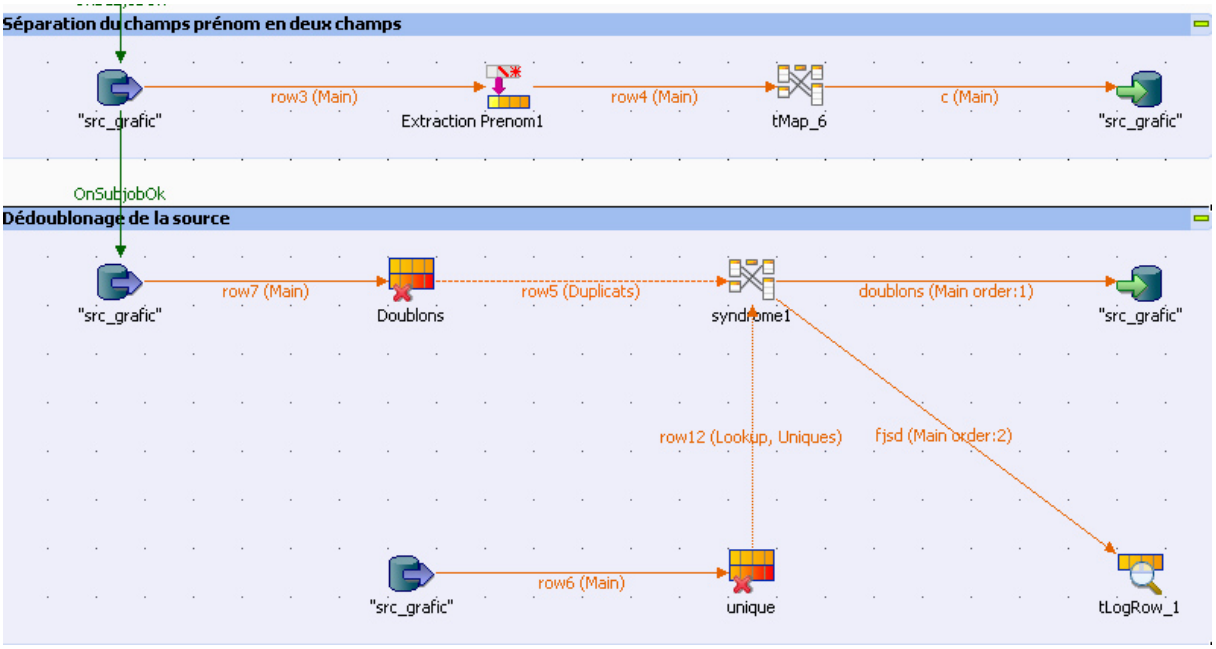


Figure 35 : Dédoublage simple de la source

Une fois les données standardisées, je procède à une détection des doublons basée sur une jointure sur le nom, le premier prénom et la date de naissance (Figure 36). Le numéro de doublons est conservé avec l'enregistrement rejeté car les données pédagogiques présentes sur le dossier seront consolidées avec le dossier effectivement repris.

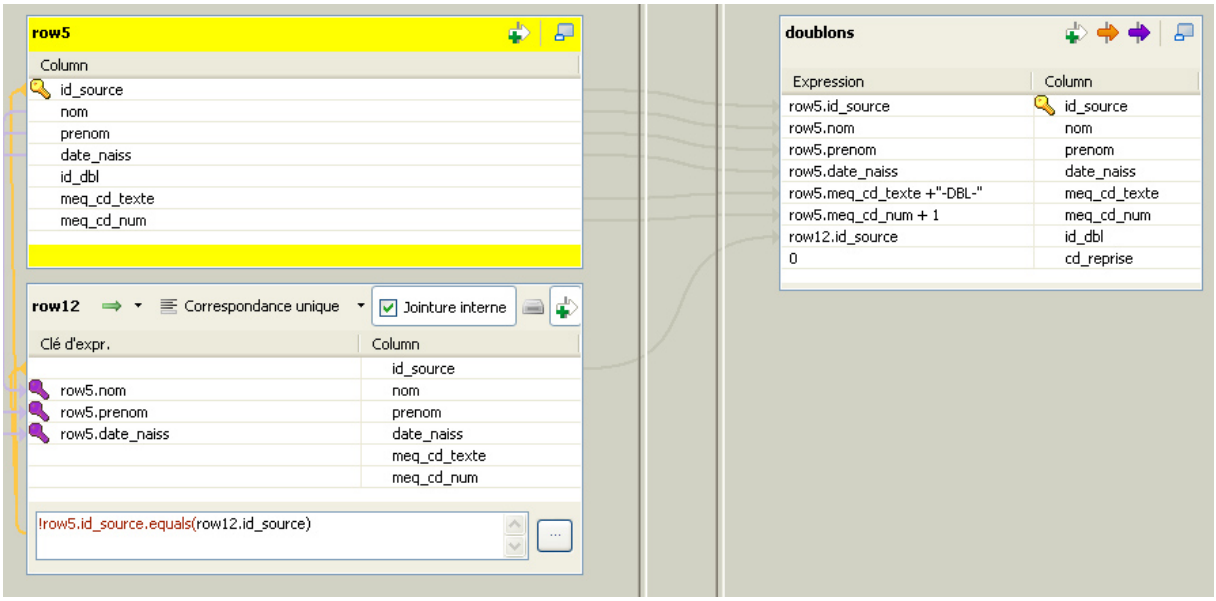
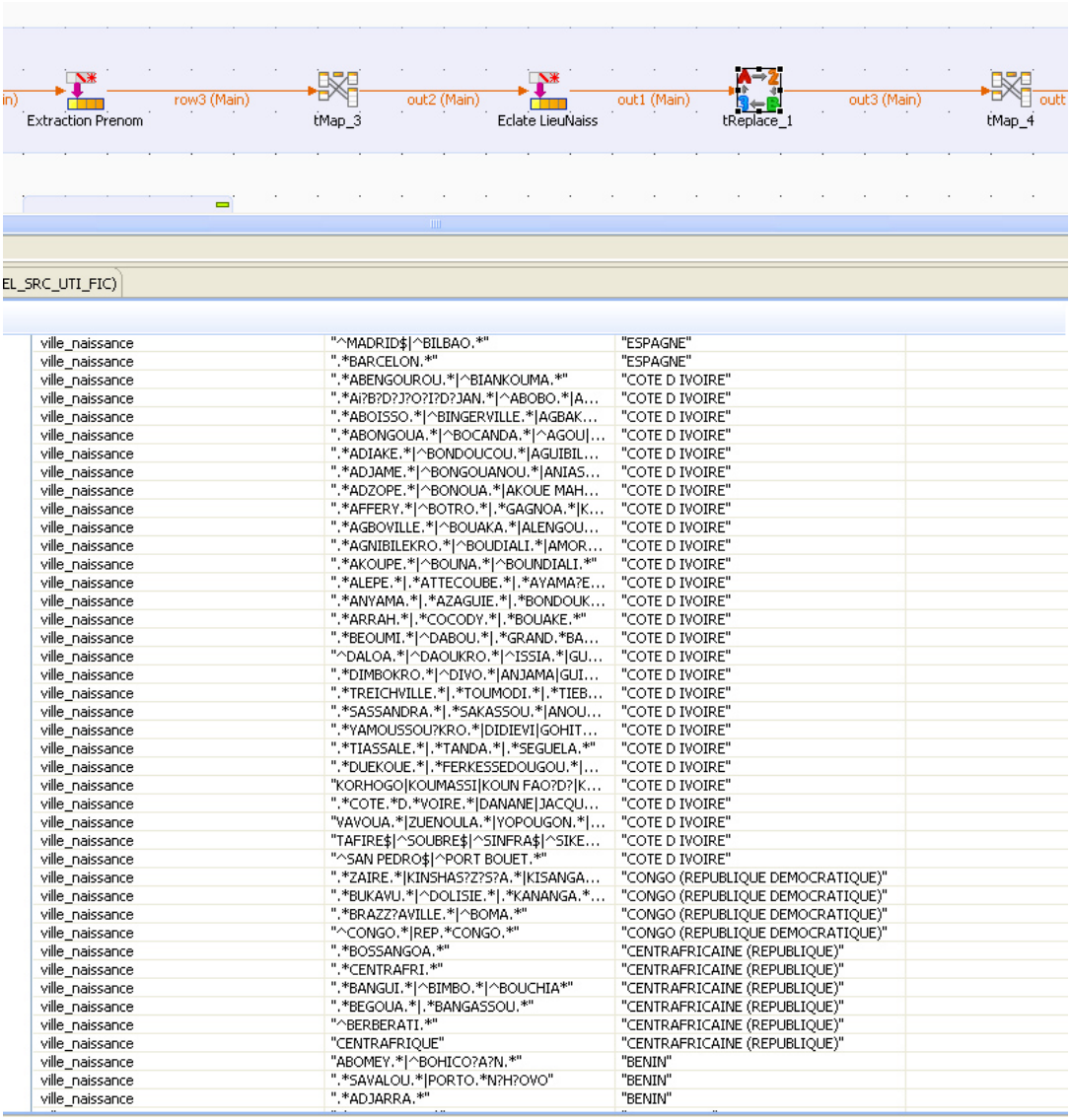


Figure 36 : Détection des doublons par jointure

3) Reprise des dossiers administratifs de l'INTEC

La reprise des dossiers pédagogique de l'INTEC est identique à la reprise des dossiers de GRAFIC. En revanche, en raison du nombre important d'élèves étrangers un effort plus particulier a été fourni afin de mettre en qualité les données relatives aux lieux de naissance et aux adresses. En effet, il n'existait pas de référentiel des villes et des pays dans UTINTEC et la saisie de ces informations étaient libres. J'ai utilisé un composant de remplacement dans lequel j'ai écrit des centaines d'expressions régulières afin de corriger les erreurs de saisie les plus courantes (Figure 37).



ville_naissance	"^MADRID$|^BILBAO.*"	"ESPAGNE"
ville_naissance	".*BARCELON.*"	"ESPAGNE"
ville_naissance	".*ABENGOUROU.*|^BIANKOUMA.*"	"COTE D IVOIRE"
ville_naissance	".*Ai?B?D?J?O?I?D?JAN.*|^ABOBO.*|A...	"COTE D IVOIRE"
ville_naissance	".*ABOISSO.*|^BINGERVILLE.*|AGBAK...	"COTE D IVOIRE"
ville_naissance	".*ABONGOUA.*|^BOCANDA.*|^AGOU|...	"COTE D IVOIRE"
ville_naissance	".*ADIAKE.*|^BONDOUCOU.*|AGUIBIL...	"COTE D IVOIRE"
ville_naissance	".*ADJAME.*|^BONGOUANOU.*|ANIAS...	"COTE D IVOIRE"
ville_naissance	".*ADZOPE.*|^BONOUA.*|AKOUE MAH...	"COTE D IVOIRE"
ville_naissance	".*AFFERY.*|^BOTRO.*|.*GAGNOA.*|K...	"COTE D IVOIRE"
ville_naissance	".*AGBOVILLE.*|^BOUAKA.*|ALENGOU...	"COTE D IVOIRE"
ville_naissance	".*AGNIBILEKRO.*|^BOUDIALI.*|AMOR...	"COTE D IVOIRE"
ville_naissance	".*AKOUPE.*|^BOUNA.*|^BOUNDIALI.*"	"COTE D IVOIRE"
ville_naissance	".*ALEPE.*|.*ATTECOUBE.*|.*AYAMA?E...	"COTE D IVOIRE"
ville_naissance	".*ANYAMA.*|.*AZAGUIE.*|.*BONDOUK...	"COTE D IVOIRE"
ville_naissance	".*ARRAH.*|.*COCODY.*|.*BOUAKE.*"	"COTE D IVOIRE"
ville_naissance	".*BEOUMI.*|^DABOU.*|.*GRAND.*BA...	"COTE D IVOIRE"
ville_naissance	"^DALOA.*|^DAOUKRO.*|^ISSIA.*|GU...	"COTE D IVOIRE"
ville_naissance	".*DIMBOKRO.*|^DIVO.*|ANJAMA|GUI...	"COTE D IVOIRE"
ville_naissance	".*TREICHVILLE.*|.*TOUMODI.*|.*TIEB...	"COTE D IVOIRE"
ville_naissance	".*SASSANDRA.*|.*SAKASSOU.*|ANOU...	"COTE D IVOIRE"
ville_naissance	".*YAMOUSSOU?KRO.*|DIDIEVI|GOHIT...	"COTE D IVOIRE"
ville_naissance	".*TIASSALE.*|.*TANDA.*|.*SEGUELA.*"	"COTE D IVOIRE"
ville_naissance	".*DUEKOUE.*|.*FERKESSEDOUGOU.*|...	"COTE D IVOIRE"
ville_naissance	"KORHOGO|KOUMASSI|KOUN FAO?D?|K...	"COTE D IVOIRE"
ville_naissance	".*COTE.*D.*VOIRE.*|DANANE|JACQU...	"COTE D IVOIRE"
ville_naissance	"VAVOUA.*|ZUENOULA.*|YOPOUGON.*|...	"COTE D IVOIRE"
ville_naissance	"TAFIRE$\|^SOUBRE$|^SINFRA$|^SIKE...	"COTE D IVOIRE"
ville_naissance	"^SAN PEDRO$|^PORT BOUET.*"	"COTE D IVOIRE"
ville_naissance	".*ZAIRE.*|KINSHAS?Z?S?A.*|KISANGA...	"CONGO (REPUBLIQUE DEMOCRATIQUE)"
ville_naissance	".*BUKAVU.*|^DOLISIE.*|.*KANANGA.*...	"CONGO (REPUBLIQUE DEMOCRATIQUE)"
ville_naissance	".*BRAZZ?AVILLE.*|^BOMA.*"	"CONGO (REPUBLIQUE DEMOCRATIQUE)"
ville_naissance	"^CONGO.*|REP.*CONGO.*"	"CONGO (REPUBLIQUE DEMOCRATIQUE)"
ville_naissance	".*BOSSANGOA.*"	"CENTRAFRICAINE (REPUBLIQUE)"
ville_naissance	".*CENTRAFRI.*"	"CENTRAFRICAINE (REPUBLIQUE)"
ville_naissance	".*BANGUI.*|^BIMBO.*|^BOUCHIA*"	"CENTRAFRICAINE (REPUBLIQUE)"
ville_naissance	".*BEGOUA.*|.*BANGASSOU.*"	"CENTRAFRICAINE (REPUBLIQUE)"
ville_naissance	"^BERBERATI.*"	"CENTRAFRICAINE (REPUBLIQUE)"
ville_naissance	"CENTRAFRIQUE"	"CENTRAFRICAINE (REPUBLIQUE)"
ville_naissance	"ABOMEY.*|^BOHICO?A?N.*"	"BENIN"
ville_naissance	".*SAVALOU.*|PORTO.*N?H?OVO"	"BENIN"
ville_naissance	".*ADJARRA.*"	"BENIN"

Figure 37 : Correction de la ville de naissance

Il a par ailleurs fallu veiller à ne pas introduire d'erreur lors des substitutions. Environ 25% des villes de naissance étaient en erreur et ne correspondaient pas aux référentiels. Grâce au travail de mise en qualité seuls 3 % des dossiers n'ont pas été repris en raison de l'absence de ville de naissance. La plupart du temps ce sont les pays de naissance qui se trouve dans le champ ville naissance. Comme les critères d'unicité retenus pour les élèves étrangers ne tiennent pas compte de la ville de naissance mais uniquement du pays, il a été décidé de remplacer directement la ville par le pays et d'effectuer les correspondances sur ce champ.

Il existe un composant natif dans Talend permettant d'effectuer une rechercher d'une valeur dans les colonnes d'entrée spécifiées et de la remplacer par une autre. La recherche peut porter sur une expression régulière. Par exemple il est possible de corriger la plupart des erreurs de saisie sur la ville d'Abidjan en filtrant le champ ville de naissance lorsqu'il correspond à l'expression régulière « .*Ai?B?D?J?O?I?D?JAN.* » puis en replaçant la ville erronée par l'orthographe correct permettant d'en déduire le pays ainsi que le code INSEE de ce dernier.

Ensuite les champs pays et ville mis en qualité sont comparés aux champs libellé de la table INSEE des pays et de la table INSEE des villes. C'est ce code INSEE qui est conservé avec l'enregistrement (Figure 38 et Figure 39).

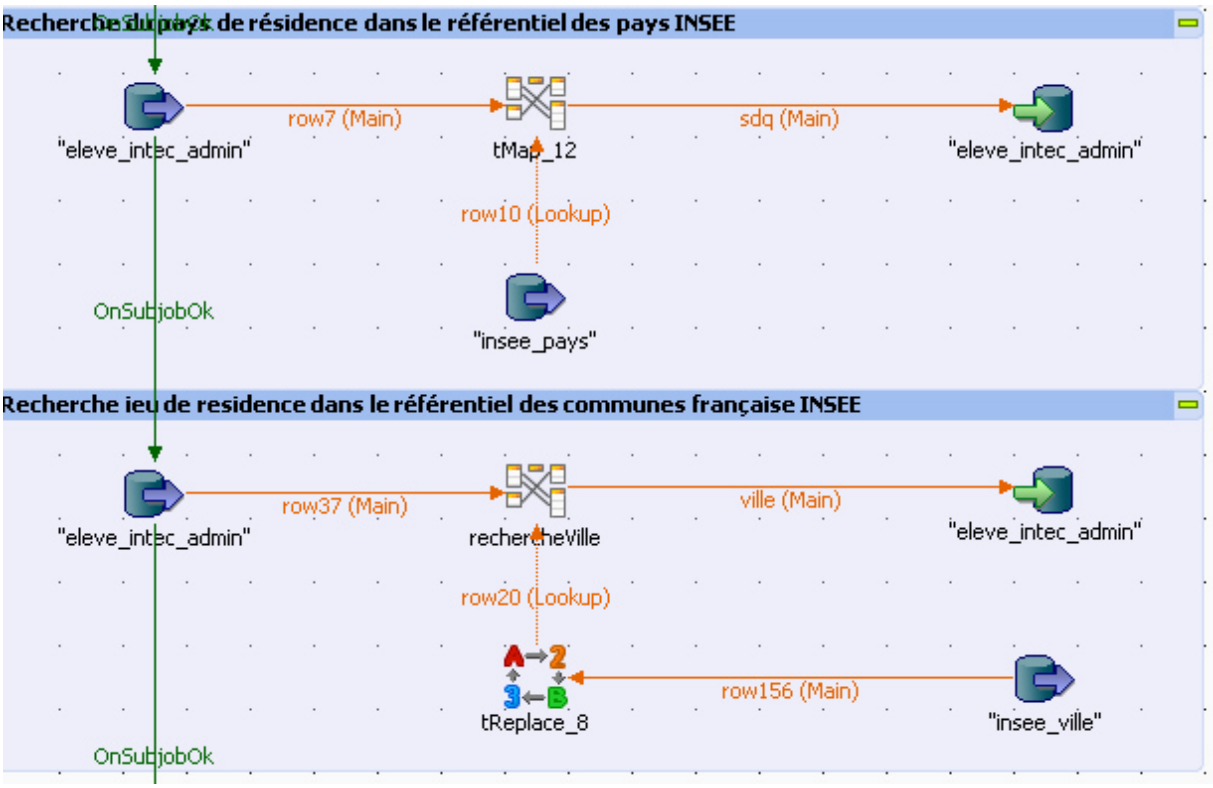


Figure 38 : Correspondance code INSEE du pays

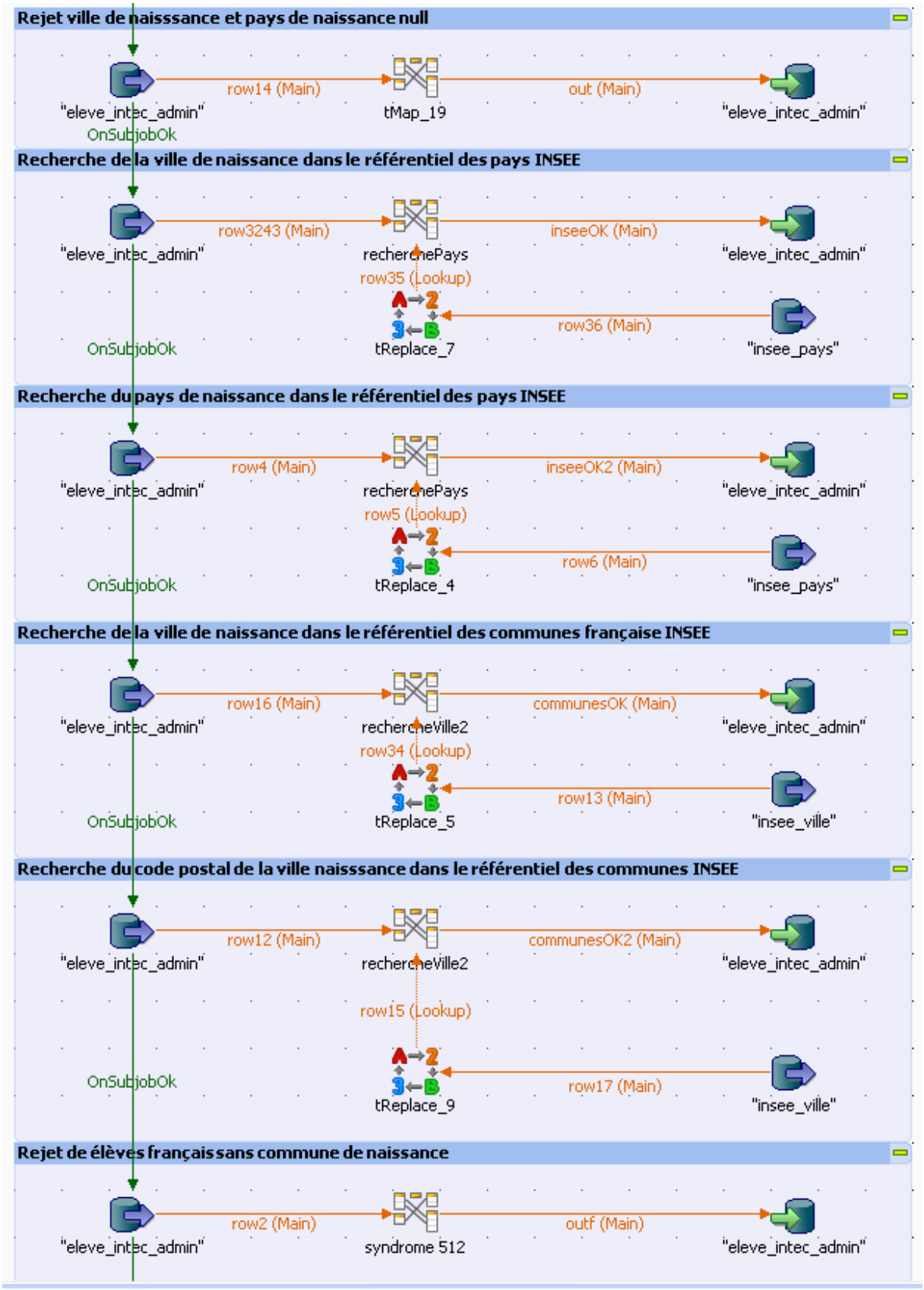


Figure 39 : Mise en qualité des villes et des pays

2. Reprise des dossiers pédagogiques de l'INTEC

Un dossier élève source est composé de 3 parties : la partie administrative, la partie pédagogique et la partie financière. Dans le cas de dossiers multiples c'est la partie administrative source la plus récente qui est retenue dans sa totalité. Cependant il faut reprendre les parties pédagogiques de chacun des dossiers. La première étape fut de remplacer les identifiants des élèves repérés comme doublons par ceux qui sont effectivement dans l'entrepôt.

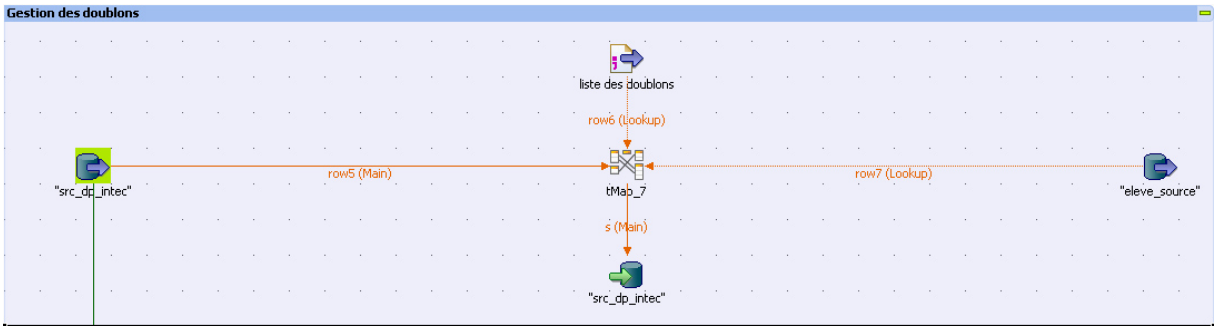


Figure 40 : Gestion des doublons

La mise en qualité des dossiers pédagogiques comporte plusieurs étapes (Figure 41):

- rejeter les dossiers des élèves qui n'ont pas été repris en raison de l'absence des critères d'unicité ;
- Remplacer les doublons par le dossier administratif effectivement repris
- séparer les modules acquis avant le passage à la réforme LMD en 2005 (objet SU) et les modules acquis après (objet SM) ;
- Faire correspondre les codes des unités d'enseignement utilisés par l'INTEC par ceux utilisés dans l'entrepôt. Les modules correspondant dans l'entrepôt son codé « TEC » suivi du code du module dans la source. Les modules de type SU sont ensuite suivis de « _SU ». Par exemple TEC_526_SU ;
- Valider les codes auprès des référentiels présents dans l'entrepôt ;
- Rejeter et marquer les données non valides.

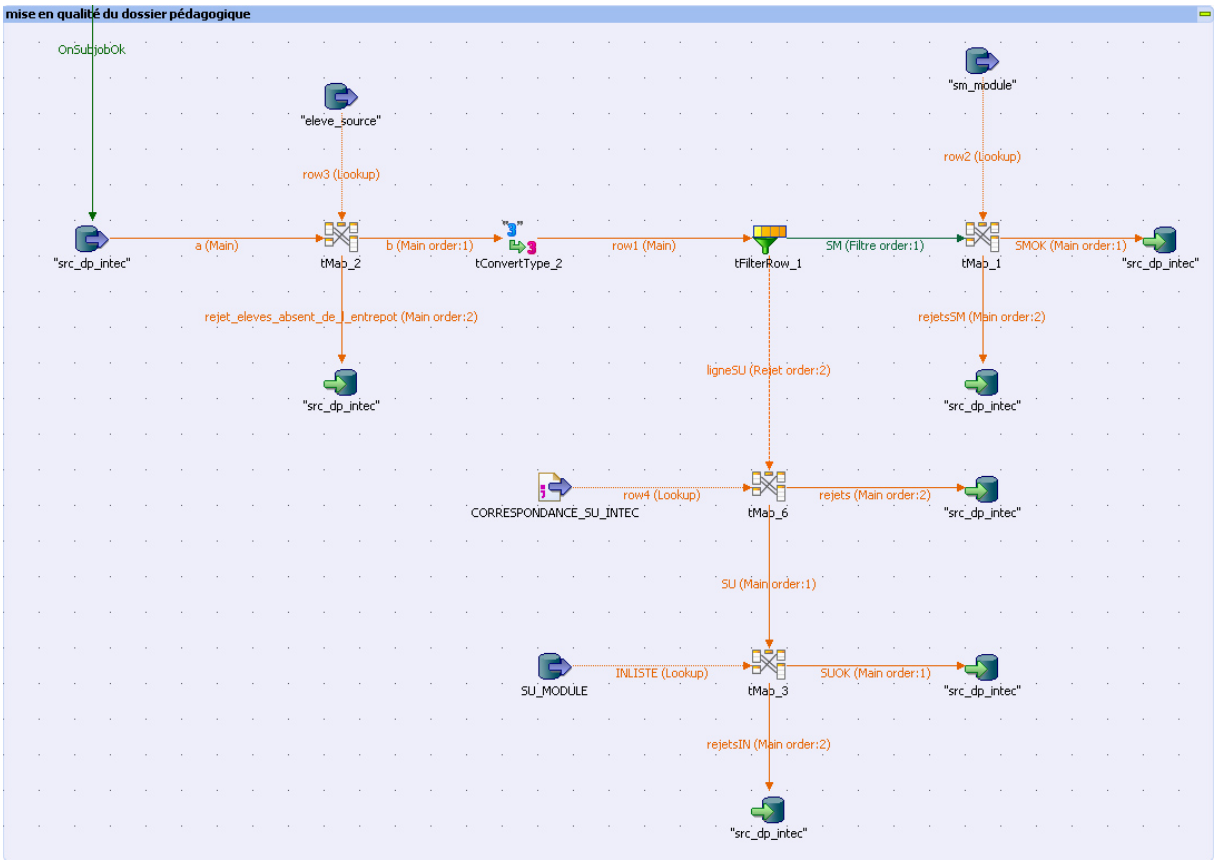


Figure 41 : Mise en qualité du dossier pédagogique de l'INTEC

J’utilise une jointure sur le code ainsi que les dates de début et dates de fin afin de valider l’existence d’une unité d’enseignement (Figure 42).

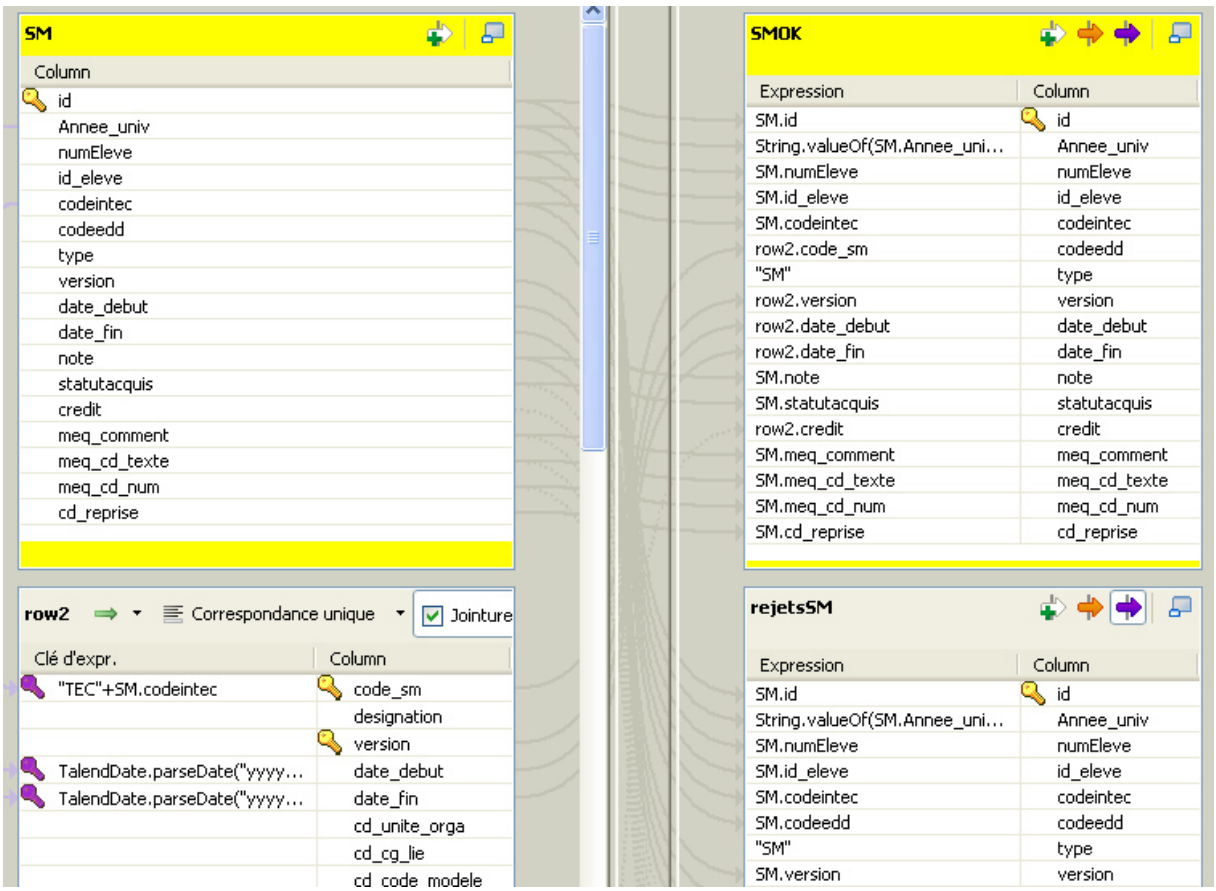


Figure 42 : Validation des SM à partir du référentiel dans l'entrepôt

3. Reprise EICNAM

1) Modalités de reprise

Le logiciel de gestion des diplômes de l'école d'ingénieur est utilisé pour instruire les demandes de délivrance des diplômes d'ingénieur CNAM ainsi que d'autres diplômes gérés par la DNF et dénommés cycle C et économiste CNAM. Le gestionnaire attribue des positions aux élèves permettant de suivre leur évolution dans leur parcours. Le gestionnaire saisit aussi les notes obtenues à l'examen probatoire ainsi qu'à la soutenance finale. Les résultats obtenus lui permettent ensuite de valider l'obtention d'un diplôme et d'éditer les documents attestant de sa réussite.

La première étape fut de remplacer les codes diplôme utilisés par EICNAM par les codes diplôme utilisés par la DNF. Par exemple aux codes EICNAM des diplômes informatique « S066A » et « S066Z », correspondent les 5 codes diplômes BDO (CYC12p-1, CYC14p-1, CYC15p-1, CYC45p-1, CYC47p-1). Puis le processus de délivrance des diplômes de l'EICNAM passe par plusieurs étapes repérées par des codes dénommés « code position ». Le tableau ci-contre liste les différents codes de la délivrance d'un diplôme et donne pour chaque code un état du diplôme (Obtenu, En cours, Abandon). La reprise des données de l'EICNAM exploite les « codes position » figurant dans le dossier de l'élève dans l'application EICNAM. A chaque rencontre des positions Diplômé et Saisie note mémoire dans un dossier il est généré un acquis externe de type diplôme (objet EQ dans SAP). Pour les positions de la phase admissibilité il est généré un acquis (SM dans SAP) de nom EI5Axx

code positio	libellé positi	Etat diplôme
00	Candidature r	Abandon
01	Provisoire	Abandon
02	Candidature a	Abandon
03	Echec C1	En cours
04	Echec C2	En cours
05	Echec C3	En cours
06	Délai supplém	Abandon
07	D (Délai Dépas	Abandon
08	DD (Délai Dép	Abandon
09	Décédé	Abandon
10	En soutenance	Abandon
11	Diplômé	Obtenu
14	Recu à la sout	Abandon
15	Candidature a	Abandon
16	Candidature a	Abandon
17	Candidature a	Abandon
18	Succès à l'épr	En cours
19	Succès à l'ens	En cours
20	Abandon cand	Abandon
21	Saisie note va	En cours
22	Saisie note pr	En cours
23	Désistement	Abandon
24	Echec à la sou	En cours
31	Dossier confo	Abandon
32	Dossier envoy	Abandon
33	dossier retour	Abandon
91	Saisie note mé	Obtenu
92	Admissible	En cours
93	Admissible so	En cours
94	Refus admissi	En cours
95	Refus admissi	En cours
96	Refus admissi	En cours

(admissible), EI5Bxx (admissible sous conditions), EI5Txx (Refus admissibilité) où xx représente les 2 derniers chiffres du diplôme CYCxx préparé. Pour les positions de la phase admission il est généré un acquis SM de nom UA5Axx (Position Admis) EI5Rxx (non admis), EI5Sxx (admis sous conditions) où xx représente les 2 derniers chiffres du diplôme CYCxx préparé. De plus il est généré une inscription au cursus (objet SC) avec son équivalent en code diplôme DNF. Pour les élèves qui auraient comme dernière position les positions 21 (saisie note valeur C) et/ou 22 (Saisie note probatoire) sans avoir de position 97 seront repris selon le même principe que la position 97 (Admis).

2) Réalisation

La reprise des dossiers administratifs des élèves de l'école d'ingénieur suit le même principe que celui décrit précédemment pour les élèves issus de GRAFIC et de l'INTEC. En revanche la reprise des dossiers pédagogiques devait prendre en compte le déversement des informations dans SISCOL. A partir des modalités de reprise, décrites dans le paragraphe précédent, j'ai réalisé plusieurs programmes permettant de convertir les données sources.

La première (Figure 43) étape fut de remplacer les codes des diplômes utilisés par l'école d'ingénieur par ceux par la DNF. Puis il a fallu vérifier que ces codes des cursus (Objet SC) et méta-diplômes (objet CQ) sont présents dans les référentiels utilisés par SISCOL.

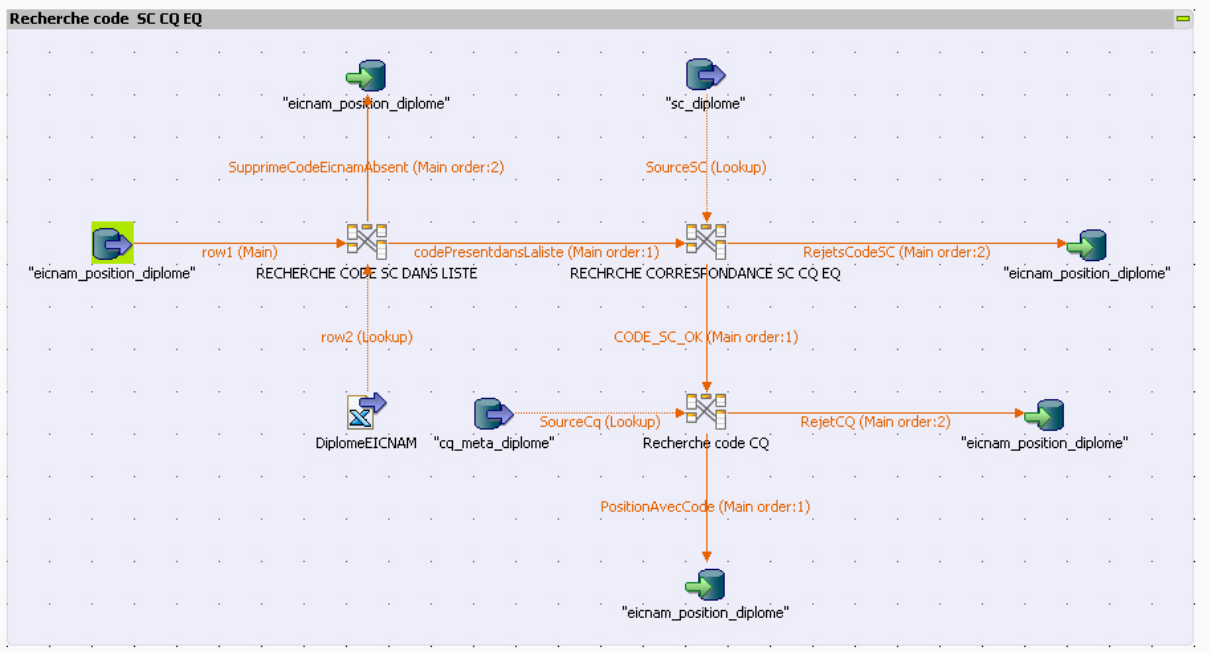


Figure 43 : Ajout des codes diplôme

Pour les élèves ayant déjà obtenu leur diplôme (Figure 44) seul un acquis hors-CNAM (objet EQ) est repris. Les autres positions du dossier de l'élève sont rejetées.

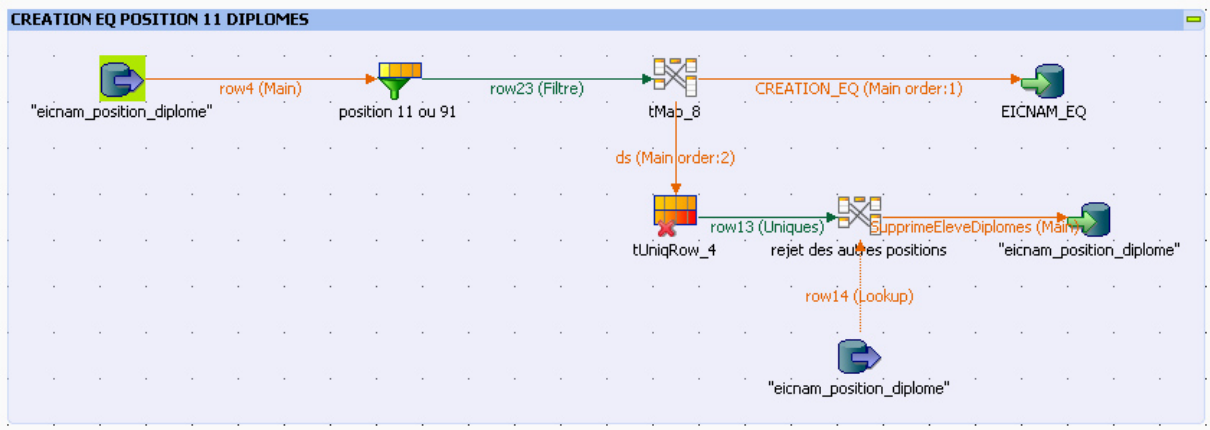


Figure 44 : Reprise des diplômes obtenus

Pour les élèves en cours d'obtention (Figure 45) et en fonction des positions présentes dans le dossier il a fallu créer soit une inscription au diplôme (objet SC), soit des unités d'activité (objet UA) soit des unités d'enseignement (objet UE).

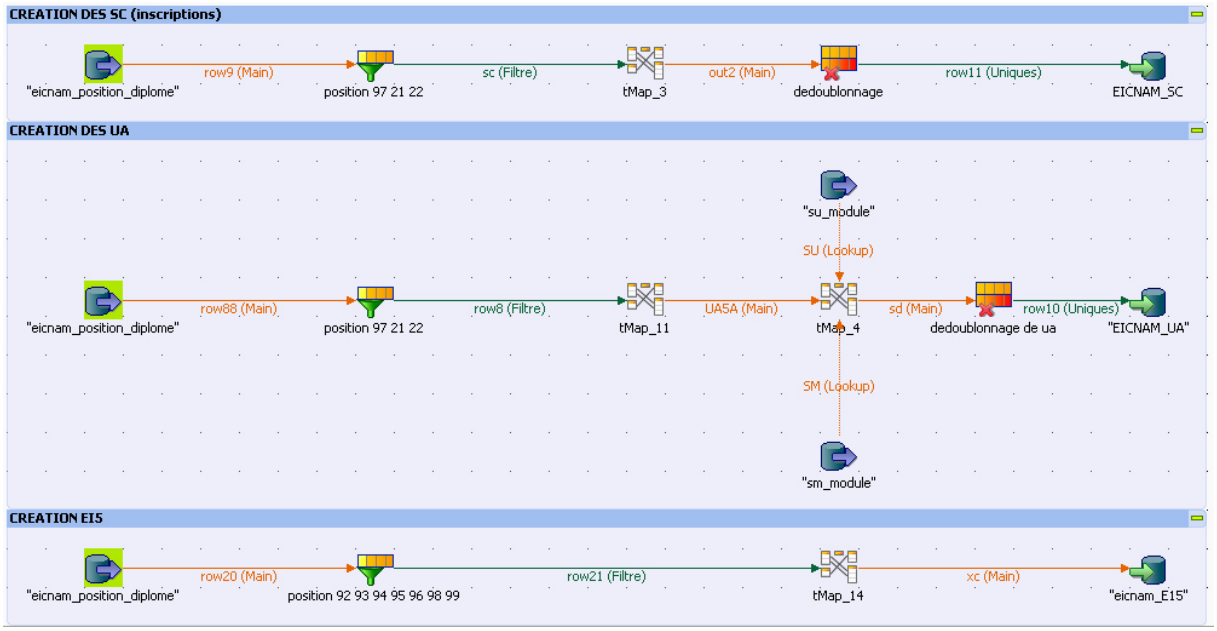


Figure 45 : Reprise des diplômes en cours d'obtention

La reprise des dossiers des élèves inscrits au cursus informatique (Figure 46) m'a demandé un traitement particulier en raison de l'inscription à plusieurs diplômes (CYC12p-1, CYC14p-1, CYC15p-1, CYC45p-1, CYC47p-1). En effet ne connaissant pas les options des cursus en informatique, il a été décidé d'inscrire tous les élèves en informatique aux 5 cursus puis c'est le gestionnaire qui supprimera les cursus non valides.

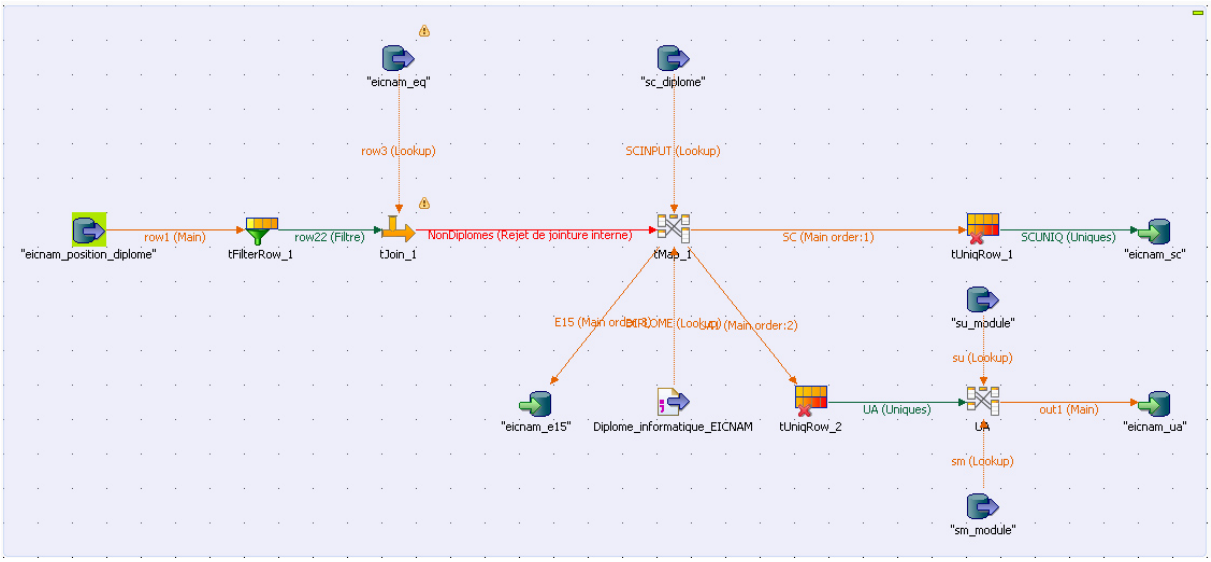
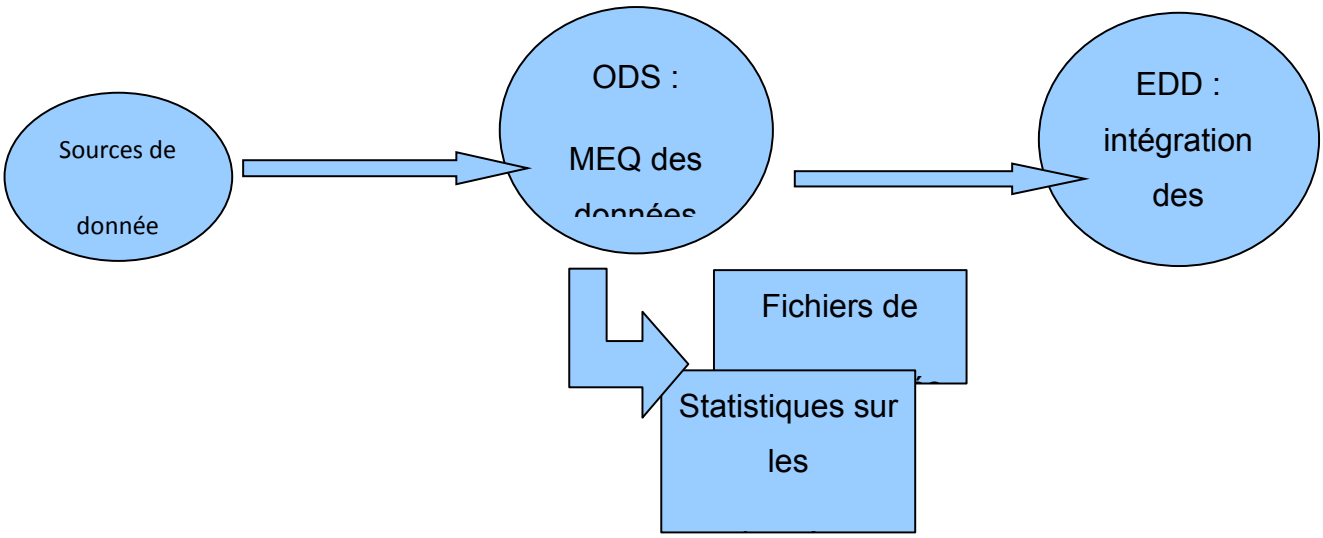


Figure 46 : Reprise des diplômes informatique

4. Gestion des rejets

Ce paragraphe a pour but de présenter le mode opératoire utilisé afin de fournir des statistiques sur la qualité des données. Nous utilisons une base de données intermédiaire entre les sources de données et l'entrepôt de données, cette base, similaire à un ODS, permettra de traiter les données entrantes pour soit les mettre en qualité, soit pouvoir clairement préciser pour chaque ligne non intégrable, quels sont les problèmes rencontrés. Le but de ces traitements étant de pouvoir pointer les champs qui produisent des rejets en raison de leur non qualité ainsi que le volume de données concernées.

1) Graphique général



Les enregistrements contenant des données rejetées sont marqués. Un champ commentaire permet de préciser en toutes lettres les problèmes rencontrés et deux autres champs sont dédiés à la codification du syndrome. Ceci nous permet de dire pour chaque source de données la volumétrie des problèmes recensés. Les données rejetées sont consignées dans des fichiers permettant ensuite de corriger les problèmes directement dans les sources ou d'adapter les programmes de reprise.

2) Description de la structure des tables

Pour chaque flux de données entrant plusieurs champs sont ajoutés permettant de marquer chaque enregistrement :

- La date d'arrivée des données dans l'ODS ;
- Un commentaire décrivant le problème rencontré sur un enregistrement ;
- Un code texte du problème rencontré ;
- Un code chiffre du problème rencontré ;
- Un statut de reprise qui précise si la ligne est à reprendre ou non ;
- Un identifiant de doublon s'il existe.

3) Alimentation de ces champs

Tout au long de cette mise en qualité les champs vont être renseignés. Lorsqu'un problème qualité est rencontré :

- le champ « commentaire » est complété avec un texte qui précise le problème rencontré, compréhensible par la population fonctionnelle qui sera amenée à corriger les données ;
- Le code texte se voit attribuer un code de 2 ou 3 caractères entouré par des tirets « - ». Ce code est unique pour une occurrence. Par exemple un code « DBL » pour préciser que l'enregistrement en cours de traitement est le doublon d'un enregistrement précédemment rencontré. Lorsqu'un enregistrement ne passe pas par un contrôle, il est mis à jour et complété avec le code du contrôle qui a échoué ;
- Le code chiffre a le même fonctionnement que le champ « code texte » mais avec des valeurs numériques qui sont additionnées afin de faire de la comparaison bit à bit et d'effectuer aisément des requêtes sur les types de problèmes rencontrés ;
- Le code reprise est initialisé à 1, lorsqu'un contrôle bloquant la reprise échoue, on positionne la valeur du champ à 0 ;
- Identifiant doublons est initialisé à la valeur « nulle ». Lorsqu'on effectue le test d'unicité et que l'enregistrement courant est un doublon d'un enregistrement

précédemment rencontré, le champ est renseigné avec l'identifiant de l'enregistrement précédemment rencontré.

4) Exemples

Prenons quelques champs d'un flux « élève administratif ».

Structure du flux :

Nom	prénom	pays	ville	adresse mail
Toto	tototo	France	Parys	toto@mail

Contrôles effectués :

1 Nom non vide :

Code texte : -NVD-

Code chiffre : 1.

2 Prénom non vide :

Code texte : -PNV-

code chiffre : 2

3 Pays dans référentiel INSEE :

code texte : -PIN-

code chiffre : 4

4 Ville dans référentiel INSEE

code texte : -VIN-

code chiffre : 8

5 adresse mail correctement formatée (XXX@XXX.XXX)

code texte : -MAL-

code chiffre : 16

5) Déroulement

Notre enregistrement test va passer avec succès les contrôles 1 ; 2 ; 3 et va échouer sur les contrôles 4 et 5. Après les contrôles 1, 2 et 3 les 3 champs sont encore vides.

Après contrôle 4 :

- commentaire : La ville « Parys » ne trouve pas d'équivalence dans notre référentiel INSEE.
- Code texte : « -VIN- »
- Code chiffre : 8

Après contrôle 5 :

- commentaire : La ville « Parys » ne trouve pas d'équivalence dans notre référentiel INSEE. L'adresse mail « toto@mail » n'est pas une adresse correcte.
- Code texte : « -VIN--MAL- »
- Code chiffre : 24 (8+16).

Pour cet enregistrement on a donc en lecture directe dans le champ « commentaire » le détail des différents problèmes rencontrés. Les deux autres champs vont permettre de faire des requêtes pour récupérer les enregistrements selon les critères voulus. Récupération des enregistrements dont le prénom est vide et le pays non identifié : avec le code texte :

*select * from matable where meq_cd_texte like '%-PNV-%-PIN-%'*

Avec le code chiffre (on additionne les codes chiffres des contrôles pour lesquels on veut récupérer les lignes qui ont échoué):

*select * from matable where (meq_cd_num & 6)>0*

6. Intégration des données élèves dans l'entrepôt

La réalimentation de l'entrepôt de données peut être effectuée de manière :

- complète : toutes les données sources sont chargées,
- incrémentale : seules sont chargées les nouvelles données sources par rapport au précédent chargement.

Dans le cas d'une réalimentation incrémentale, l'ETL doit être capable d'identifier les « nouvelles » données. Il existe, pour cela, plusieurs possibilités, qui sont présentées par Michele Bokun et Carmen Taglienti (13):

- si les données sources sont datées, le système peut se reposer sur ces informations,
- le système peut effectuer des comparaisons de données entre les sources et la cible,
- des triggers peuvent être mis en place au niveau des sources de données. Ceux-ci se déclenchent à la mise à jour des données et stockent ainsi les changements effectués dans un espace réservé,
- les logs de transactions peuvent être analysés afin de tracer les changements,

Un processus de reprise de données joué plusieurs fois sur les mêmes jeux de données implique de mettre en place des mécanismes permettant de gérer les versions des données ajoutées. De plus, il faut gérer les versions des données ajoutées pour chaque enregistrement.

La Figure 47 décrit la phase d'ajout d'un auditeur mis en qualité. D'abord l'identifiant de l'auditeur dans la source est comparé à ceux qui ont déjà été repris. S'il existe, l'auditeur est stocké dans un fichier comme existant pour un traitement ultérieur sur les données d'adresse, employeur, etc. Puis ce sont les critères d'unicités, nom, prénom date de naissance qui sont comparés avec ceux des auditeurs déjà présents dans l'entrepôt. Si un doublon entre source est détecté l'auditeur est marqué comme existant mais l'identifiant de la nouvelle source est ajouté dans l'entrepôt. Sinon l'auditeur est créé dans l'entrepôt et ajouté au fichier des élèves créés qui sera utilisé lors de l'ajout des données d'adresse (Figure 48), de contact, d'employeur etc.

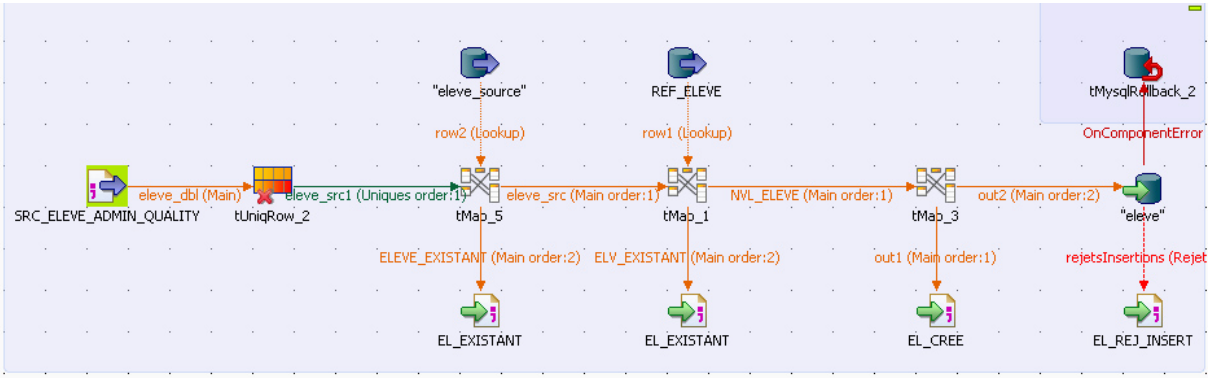


Figure 47 : Processus d'ajout d'un nouvel élève dans l'entrepôt

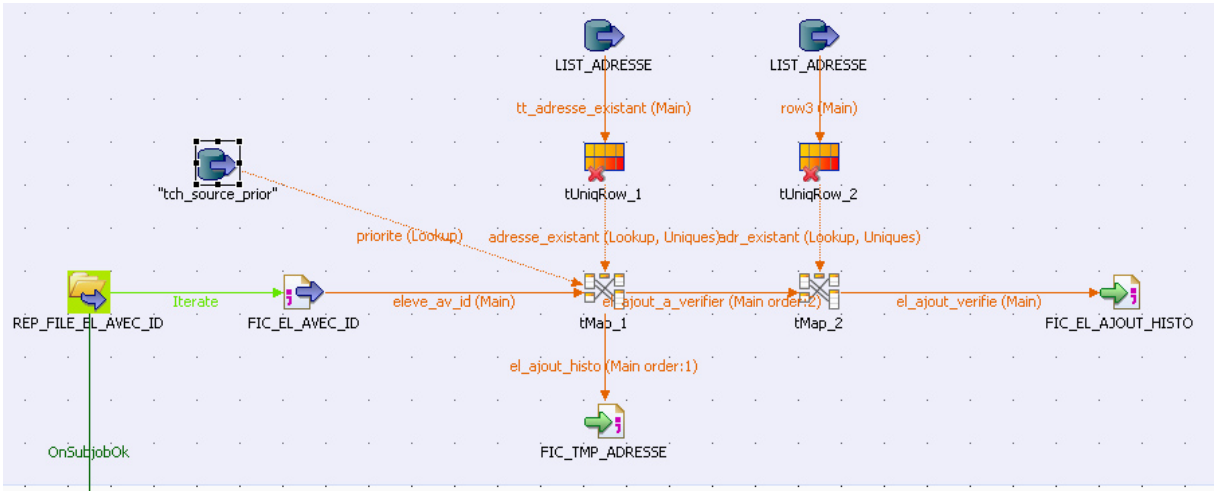


Figure 48 : Processus d'ajout d'une adresse

7. Synthèse

Dans ce chapitre j'ai pu décrire la manière dont les données ont été effectivement reprises puis mises en qualité avant d'être ajoutées à l'entrepôt de données. Le Tableau 7 donne quelques chiffres permettant de mieux mesurer l'impact des différents processus de mise en qualité sur les données reprises. On y voit notamment le nombre important de doublons dans les sources mais aussi entre les sources ainsi que l'importance du travail de la mise en qualité des villes de naissance. 70% des dossiers élève du CEP, 30% de ceux de l'école d'ingénieurs ainsi que 27% de ceux de l'Intec ne permettaient pas d'identifier la ville de naissance dans nos référentiels. 98% des dossiers du CEP et de l'école d'ingénieurs ont été repris et 88% pour l'Intec en raison du nombre d'étudiants nés à l'étranger.

Tableau 7 : impact de la mise en qualité sur la reprise de données élèves

	UTINTEC	EICNAM	CEP
nb total dossiers dans la source	119093	21443	253853
dossier occurrence double	614	118	1738
dossier occurrence triple	8	5	15
dossier occurrence > 3	3	0	0
Dossiers uniques intégrables	118468	21320	252100
Rejets pour absence de nationalité	0	0	8538
Rejets pour identifiant source inexploitable	0	0	7
Rejets pour nom prénom inexploitables	54	349	130
Rejets sur lieu de naissance	31988	5722	180457
Rejets sur lieu de résidence	11011	6404	1962
Dossiers repris avant mise en qualité	75415	8835	61004
Nombre de dossiers mis en qualité	29217	12185	186957
Dossiers intégrables à l'entrepôt de données après mise en qualité	104632	21020	247961
dossier commun GRAFIC - UTINTEC	5658		5658
dossier commun GRAFIC - EICNAM		7678	7678
dossier commun UTINTEC - EICNAM	88	88	
dossier commun GRAFIC - UTINTEC - EICNAM	54	54	54
Nombre de dossiers intégrés dans l'entrepôt	98974	13342	247961
Total	360277		

V. Interfaçage avec SAP

1. La migration de données vers le système SAP

1) Motivations

Lors de la mise en place de la nouvelle application de scolarité et de la migration vers le nouveau système SAP, les données de productions des anciennes applications doivent être préservées. Le but de la migration de données est de transférer les données existantes dans le nouvel environnement. Celles-ci doivent être transformées sous un format approprié pour être intégrées au nouveau système et ce, tout en préservant l'information stockée dans l'ancien système.

2) Processus de migration de Données

Lors d'une fusion ou de l'acquisition d'une nouvelle application, les applications redondantes sont la plupart du temps abandonnées mais les données qu'elles contiennent doivent être préservées dans le système subsistant. Lors de la migration l'ancien et le nouveau système ont besoin de coexister durant une certaine période. Le processus de migration des données (Figure 49) inclus une transformation de la représentation des données utilisées dans les applications sources afin qu'elles correspondent à un format attendu par le système cible.

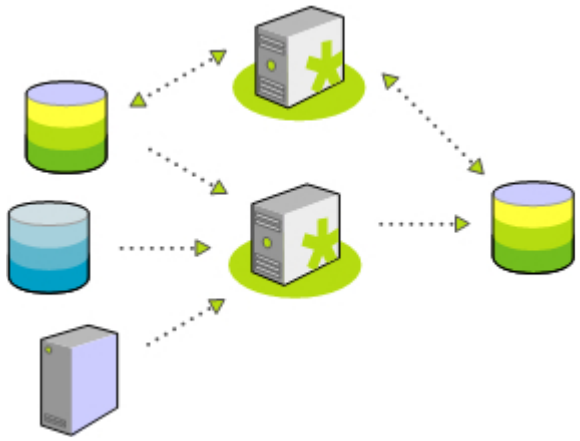


Figure 49 : Processus de migration des données

3) Les Challenges de la Migration de Données

La migration de données est souvent vue comme un aspect accessoire de la migration d'applications ou de systèmes. Il y a cependant plusieurs défis à relever pour effectuer une migration de données.

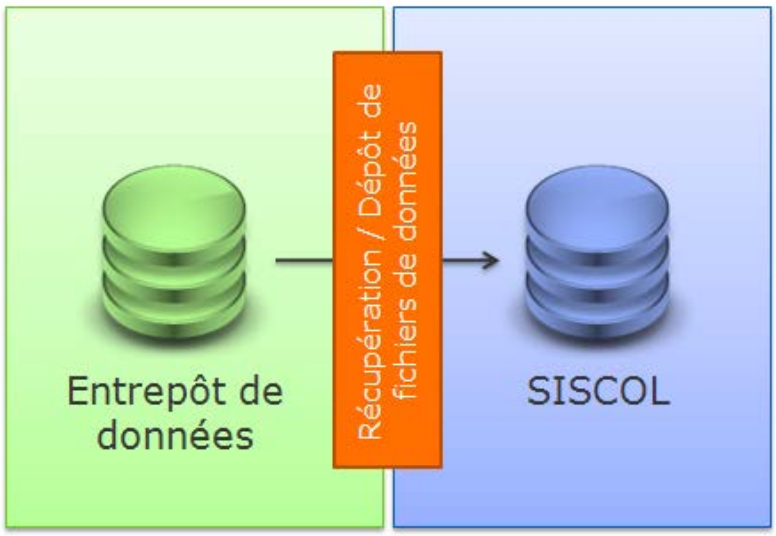
- Les migrations impliquent souvent des volumes de données importants. Dans notre cas, la migration concerne tout l'historique des transactions de l'établissement effectué dans les applications de scolarité. Elle implique aussi le traitement de larges quantités de données individuelles notamment lors de la mise en qualité de celles-ci.

- Les migrations ont souvent lieu dans des environnements hétérogènes où les structures des données source et cible sont très différentes. De plus, les structures de données sont la plupart du temps mal documentées voir pas documentées. enfin, des mappages et transformations complexes sont requis avec à l'aide de clés d'agrégation et des règles de calculs ou de transformations qu'il faut définir.

- Dans beaucoup de cas, la cohérence doit être maintenue entre l'ancien et le nouveau système après que les données aient été migrées. C'est, par exemple, le cas quand de nombreuses applications travaillent à partir des mêmes bases de données mais ne sont pas migrées en même temps. Ou lorsque le nouveau système est implémenté de manière progressive auprès des utilisateurs. Dans ces cas de figure, il devient généralement nécessaire de réaliser des synchronisations bidirectionnelles entre l'ancien et le nouveau système.

2. Interfaces entrantes SAP

1) Objectifs



Le but des interfaces entrantes est de fournir des fichiers plats permettant le chargement des données de l'entrepôt issues de la reprise afin de les intégrer dans SAP. Ces fichiers permettent d'alimenter un programme spécifique développé par l'équipe LOGICA qui réalise la création des objets dans SAP. Ces fichiers doivent respecter un format répondant aux spécifications fonctionnelles définit par la société de service. Un système de gestion des d'erreur permet d'identifier puis de livrer à nouveau les données erronées une fois corrigées. Lors de la création des fichiers, j'ai dû respecter d'autres contraintes fonctionnelles comme :

- vérifier que les élèves qui n'ont pas de dossier pédagogique n'ont également pas de dossier administratif ;
- Le numéro d'étudiant est défini sur 9 caractères numériques et doit être compris entre 100000000 et 199999999 ;
- La taille des fichiers ne doit pas dépasser 50.000 lignes ;
- toutes les dates début des objets sont alimentées à '19000101' et les dates fin à '99991231' ;
- L'absence de donnée disponible pour un champ se traduit dans le fichier par un champ vide c'est-à-dire 2 tabulations consécutives ;
- Ces fichiers sont au format UTF8 avec en caractères de fin de ligne : CR et LF.

2) Dossier administratif

Les données administratives du dossier élève sont gérées dans des info-types (ou regroupement d'informations de même nature) de l'objet ST (Etudiant) et des tables relatives au partenaire financier (BP-Business Partner) représentant l'élève.

Les info-types concernées par la reprise des dossiers administratifs sont l'identité, les coordonnées, le numéro INE, les données de groupe, les données sur l'activité professionnelle, les données liées au handicap au statut de séjour et au visa. J'ai choisi d'illustré le principe de l'intégration des données dans SAP à travers l'étude de la création du fichier contenant les données liées à l'identité des auditeurs.

Le Tableau 8 donne les champs nécessaires à la création des objets élève dans SAP. Il y apparait notamment les champs à fournir obligatoirement qui correspondent aux critères d'unicités de l'entrepôt ainsi que le type de donnée attendu, la longueur maximale des champs et l'usage d'une clé appartenant à un référentiel. A partir de ces informations j'ai pu créer un schéma de fichier cible dans Talend.

Tableau 8 : Schéma du fichier identité

ellé SAP du cha	Type d'aff.	IT	Nom technique	Type	Longueur	Référentiel?
uméro d'étudia	Obl.	1000	SHORT	CHAR	12	NON
Date de début	Obl.	1000	BEGDA	DATS	8	NON
Date de fin	Obl.	1000	ENDDA	DATS	8	NON
Clef de sexe	Fac.	1702	GESCH	CHAR	1	'1' : Masculin '2' : Féminin
Qualité	Fac.	1702	ANRED	CHAR	1	oui
Nom	Obl.	1702	NACHN	CHAR	40	non
Nom de naissance	Obl.	1702	BIRTHNAME	CHAR	40	non
Prénom	Obl.	1702	VORNA	CHAR	40	non
2e prénom	Fac.	1702	MIDNM	CHAR	40	non
Date de naissance	Obl.	1702	GBDAT	DATS	8	non
Ville de naissance (code INSEE)	Obl.	1702	ZXXX	CHAR	10	oui
Pays de naissance	Obl.	1702	GBLND	CHAR	3	oui
Région de naissance	Obl. / Fac.	1702	GBDEP	CHAR	3	oui
Nationalité	Fac.	1702	NATIO	CHAR	3	oui
Situation de famille	Fac.	1702	FAMST	CHAR	1	oui

La Figure 50 présente le programme que j'ai réalisé afin de créer les fichiers plats à partir des sources de données historiques élèves (version des données élèves), données élèves. La Figure 51 décrit les principales correspondances entre les sources et le schéma cible effectuées au niveau du premier composant Tmap_1.

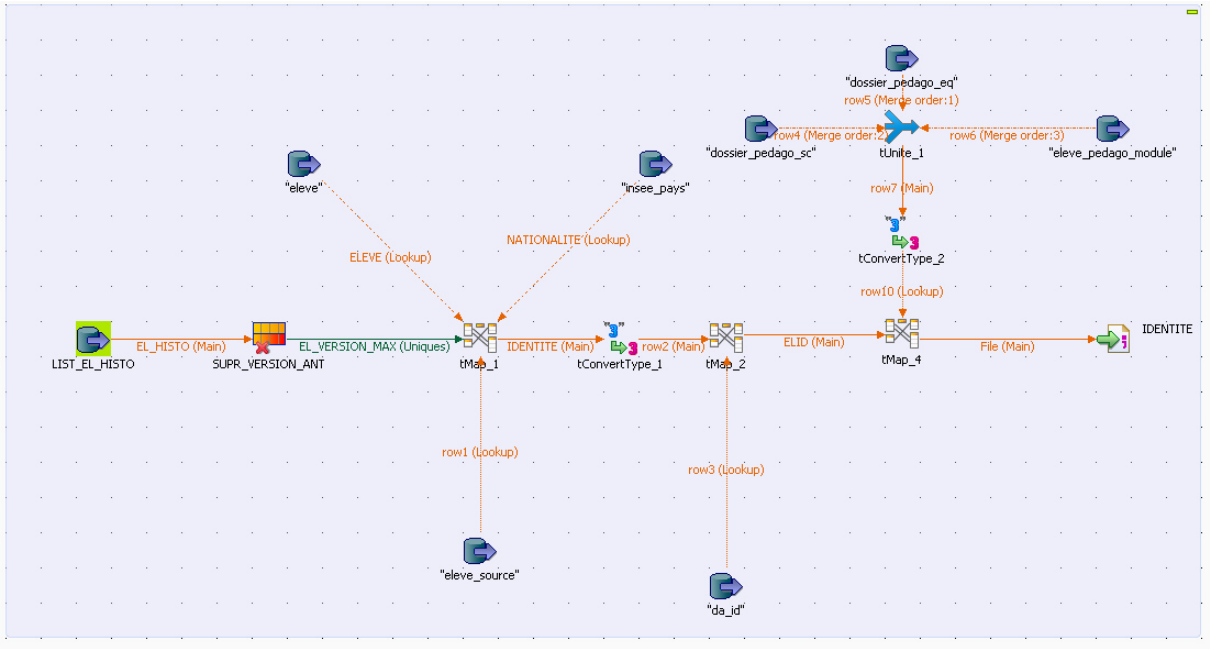


Figure 50 : job reprise identité

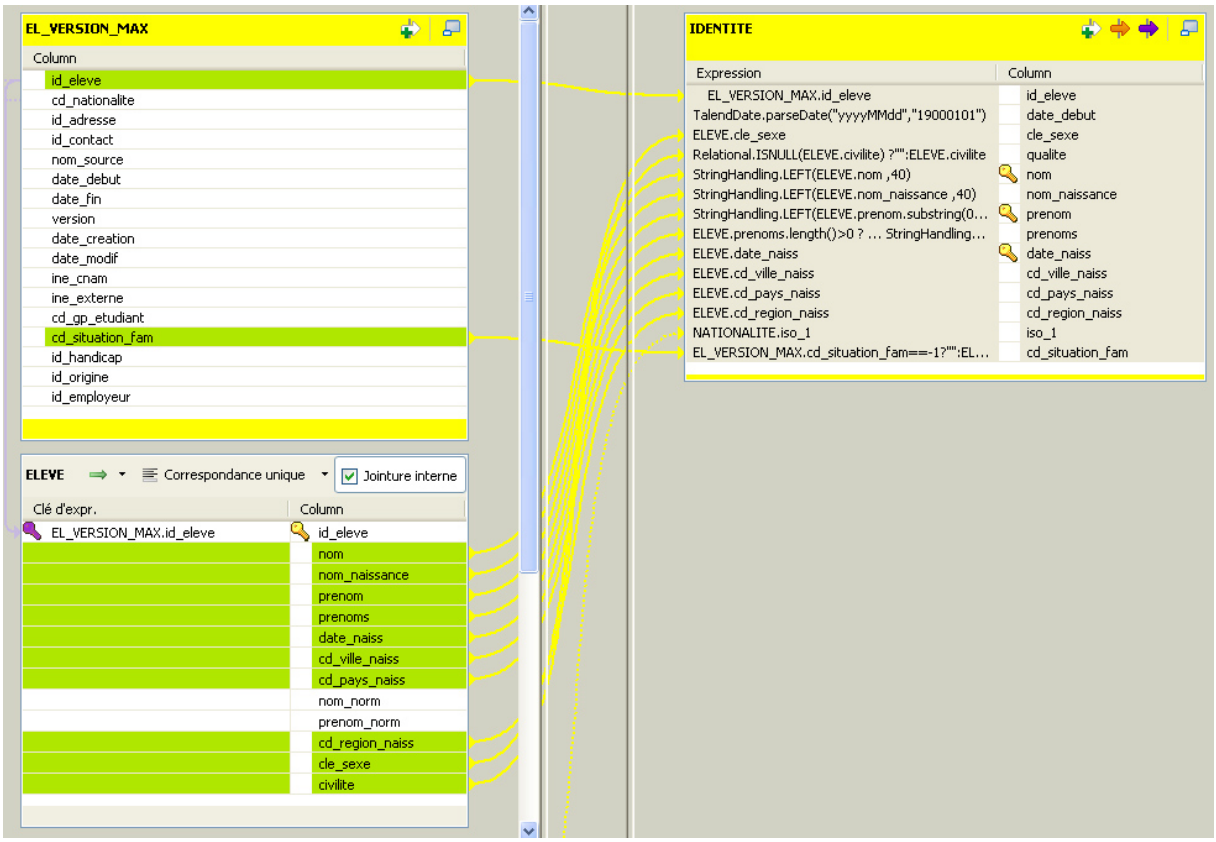


Figure 51 : Correspondance des données source vers le schéma cible

La jointure réalisée dans le Tmap_2 permet de fournir les auditeurs qui ne sont pas encore présents dans SAP. A l'aide d'une jointure sur données pédagogiques (Inscriptions aux cursus, les diplômes obtenus et les inscriptions au UE) je filtre les élèves possédant un dossier pédagogique.

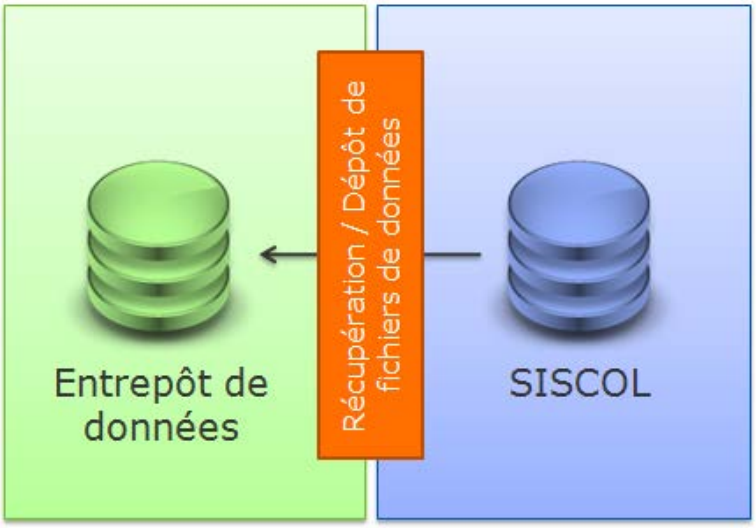
3) Recette des fichiers générés

Scripts Shell permettant de vérifier que les fichiers fournis répondent aux formats et aux contraintes des programmes d'intégration dans SAP. Ces scripts réalisent opération comme des tris ou des comparaisons afin de valider la structure du fichier et la cohérence des informations fournies. Vous trouverez en annexe un exemple de script.

3. Interfaces sortantes de SAP

1) Objectif

L'objectif des interfaces sortantes est d'identifier les données de SAP à interfacer avec l'entrepôt de données. Dans un premier temps Il a fallu préciser les données utiles qui sont aujourd'hui extraites de SISCOL. LOGICA nous a fourni la structure des fichiers extraits ainsi que les référentiels utilisés et j'ai réalisé les jobs en entrée de l'entrepôt.



On entend par interface, la mise en place d'une solution qui permette :

- un échange de données avec un système externe, en l'occurrence l'entrepôt de données ;
- l'extraction de données, sous un format défini, pour mettre à disposition un ensemble de données défini ;
- l'intégration de données, sous un format défini, pour mettre à disposition des utilisateurs de SISCOL les informations nécessaires à la réalisation des tâches qui leur sont affectées ;
- Prise en compte des modifications et des évolutions des interfaces ;
- Système d'information (Email, fichier erreur) en cas d'erreur lors de l'importation des données dans l'entrepôt.

Dans le cadre du projet SISCOL, des interfaces ont été réalisées afin de retourner l'ensemble des informations administratives et pédagogiques des élèves, mais aussi les informations liées aux partenaires financiers (entreprises et organisations), aux sessions de formation, etc... L'ensemble des données interfacées sont détaillées dans la partie

dictionnaire de données. Les échanges de données entre SISCOL et les systèmes externes passent par l'entrepôt de données, dans lequel les transcodages sont effectués.

2) Dictionnaires de données

Les données interfacées sont de plusieurs natures :

- les données administratives d'un élève comme son adresse, ou les informations sur son activité professionnelle.
- les données pédagogiques sur sa scolarité au CNAM depuis la mise en exploitation SISCOL (objet SM) et avant la mise en exploitation de SISCOL (Objet SU). Les données relatives au cursus suivi au CNAM
- des données relatives à la planification provenant de SISCOL
- Business Partner (entreprises et organisations)
- les sessions de formation, les états de services (temps de travail exécuté par les enseignants) et l'occupation des salles

Les données sont retournées selon le principe du delta des enregistrements ou delta sur la personne.

3) Besoin fonctionnels

Les interfaces sortantes des données administratives des élèves réalisées dans le cadre de SISCOL selon 7 extracteurs de données :

- Identité
- INE
- Données d'études
- Coordonnées
- Activité professionnelle
- Handicap
- Visa / Séjour

Les interfaces sortantes des données pédagogiques des élèves réalisées dans le cadre de SISCOL selon 8 extracteurs de données :

- Accompagnateur pédagogique
- Diplôme visé
- Diplôme obtenu dans SISCOL
- Inscriptions aux cursus
- Inscriptions aux modules
- Témoins de blocage
- Modules externes (Parcours hors SISCOL)
- Diplômes externes (Parcours hors SISCOL)

Les interfaces sortantes des sessions de formation regroupent 4 extracteurs de données :

- Ouverture des sessions
- Etats de services sur des enseignements non planifiés (objets EL)
- Etats de services sur des enseignements planifiés (objets E)
- Occupation des salles

De plus, deux extracteurs sont ajoutés à l'interface afin de transmettre les informations relatives aux types de formation.

Afin de permettre au CNAM d'exploiter les données d'assiduité des élèves de manière autonome, une interface sortante de l'assiduité des élèves est créée.

Les aspects impliqués dans cette interface sont :

- La présence ou l'absence de l'élève aux séances auxquelles il est inscrit
- L'éventuelle annotation indiquée pour cette séance

Afin de permettre de compléter les informations stockées dans l'entrepôt de données, une interface est mise en place, permettant d'extraire les données relatives aux partenaires, ainsi que les liens concernés.

Les aspects impliqués dans cette interface sont :

- Les liens entre l'inscription d'un élève et les partenaires payeurs
- Les informations générales des partenaires
- Les adresses des partenaires

Afin de permettre de compléter les informations stockées dans l'entrepôt de données, une interface est mise en place, permettant d'extraire les données relatives aux conventions spécifiques.

Dans ce document, le terme convention désigne une convention ou un avenant. En effet, un avenant est représenté dans SISCOL comme une convention étant liée à une convention « mère ».

Les données sont regroupées dans 7 fichiers, de la manière suivante :

- Un fichier contient les données objet (code et libellé) et l'entité de gestion
- Un fichier contient les lots d'inscriptions liés
- Un fichier contient la convention père
- Un fichier contient le centre de convention
- Un fichier contient les descriptions de la convention
- Un fichier contient les données de tarification
- Un fichier contient la validation de la convention

4) Protocole de chargement

Chaque nuit un programme extrait les données ajoutées ou mises à jour dans le logiciel SISCOL et génère des fichiers plats. Ces fichiers suivent des schémas définis par la société de service. Dès qu'une modification est détectée sur un groupe de données :

- soit tout le groupe de données est transmis en incluant les modifications effectuées (delta sur la personne)
- soit seules les informations modifiées dans le groupe sont transmises (delta sur les enregistrements)**.**

Les exemples ci-dessous explicitent les principes du delta sur la personne et du delta sur les enregistrements.

i. Delta sur la personne :

Dès qu'une modification est détectée sur un enregistrement, tout l'infotype, avec son historique, est transmis à raison d'une ligne par enregistrement. Cet historique est alors conservé dans l'entrepôt de données.

ii. Delta sur les enregistrements :

Dès qu'une modification est détectée sur un enregistrement de l'infotype, une ligne est créée, contenant l'enregistrement modifié.

Les dates de début et de fin sont calculées selon le modèle suivant :

- dès qu'un nouvel enregistrement débute pour l'un des infotypes, la date début de la ligne est égale à celle de cet infotype,
- dès qu'un enregistrement se termine pour l'un des infotypes, la date de fin de la ligne est égale à la date de fin de cet infotype,

Si deux lignes sont identiques, elles sont supprimées de l'entrepôt.

Dans le cas du delta sur la personne l'historique du groupe de données est géré entièrement dans SISCOL tandis que dans le cas du delta sur les enregistrements un historique doit être géré dans l'entrepôt.

Tableau 9 : liste les 15 groupes de données du dossier élève

Domaine	Thème	Sous-thème	Type de Delta
Dossier administratif	Identité + ID		**Delta sur les enregistrements**
	INE		**Delta sur les enregistrements**
	Données d'études		**Delta sur les enregistrements**
	Coordonnées	Adresses	**Delta sur la personne**
		Téléphones	**Delta sur la personne**
		Adresses mail	**Delta sur la personne**
	Activité professionnelle		**Delta sur la personne**
	Handicap		**Delta sur les enregistrements**
	Visa/séjour		**Delta sur la personne**
Dossier pédagogique	Accompagnateur pédagogique		**Delta sur la personne**
	Diplôme visé		**Delta sur les enregistrements**
	Diplôme postulé (succès, échec, en cours)		**Delta sur la personne**
	Inscription à un cursus		**Delta sur les enregistrements**
	Inscription à un module		**Delta sur les enregistrements**
	Témoin de blocage		**Delta sur les enregistrements**
Parcours hors-siscol	Module externe		**Delta sur les enregistrements**
	Diplôme externe		**Delta sur les enregistrements**

5) Réalisation

Les jobs réalisant l'importation des données extraites de SAP sont classés par type d'objet (Figure 52). Lors de la reprise de données j'ai eu tendance à créer des programmes réalisant beaucoup de traitement et de transformation sur les données et je me suis aperçu que cela diminuait les performances lors de l'exécution. J'ai préféré simplifier les programmes afin qu'ils ne réalisent que des tâches simples pour ensuite les regrouper dans un programme principal par catégorie et enfin en un seul programme d'importation nommé main qui est exécuté chaque nuit comme défini dans le protocole de chargement.

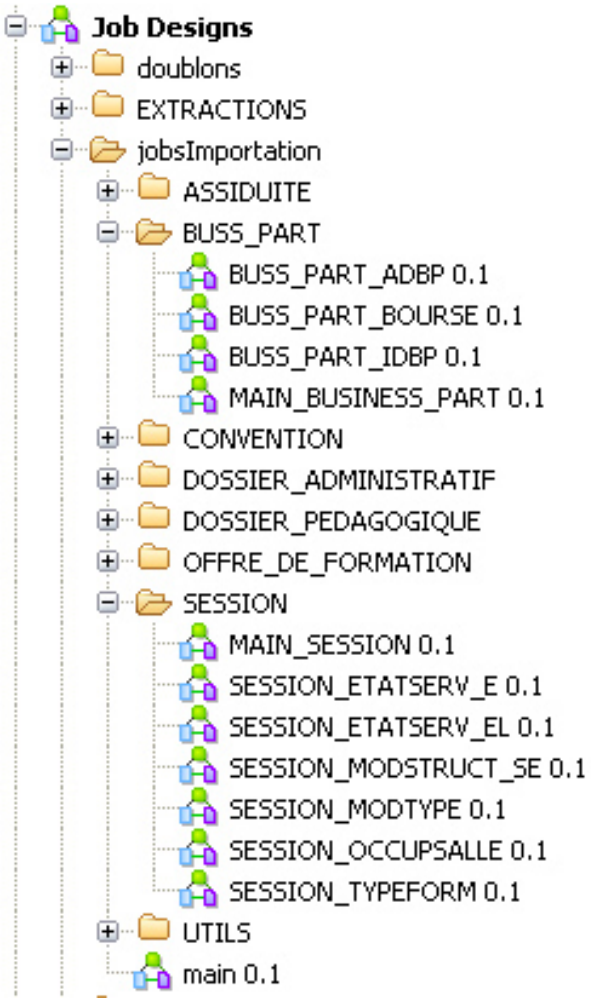


Figure 52 : Organisation des jobs

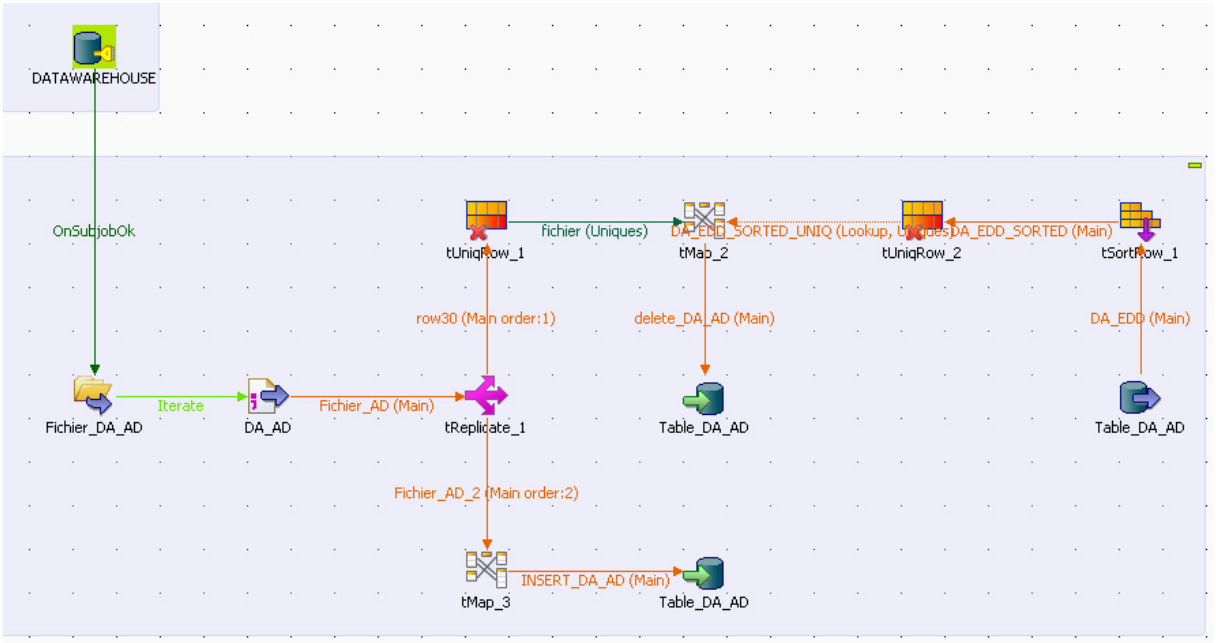


Figure 53 : Importation des données d'adresse

Chaque job réalise l'intégration des données d'un objet. Par exemple les données d'adresse (Figure 53) sont transmises suivant le delta à la personne (voir paragraphe 5.3.4), il s'agit donc de supprimer pour chaque auditeur le dernier enregistrement de l'entrepôt. Puis ajouter le contenu du fichier dans la table des adresses. Pour chaque objet l'ajout des données dans l'entrepôt demande un traitement spécifique répondant aux spécifications. Puis ces sous jobs sont packagés par domaine fonctionnel (Figure 54) puis tous les domaines sont exécutés dans un programme global (Figure 55) qui est exécuté chaque nuit. Un système de log permet de tracer les éventuelles erreurs et les fichiers traités sont archivés.

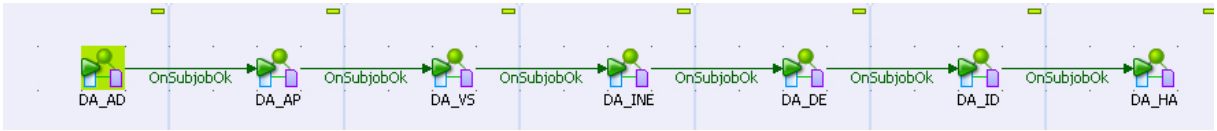


Figure 54 : Enchainement des jobs du dossier administratif

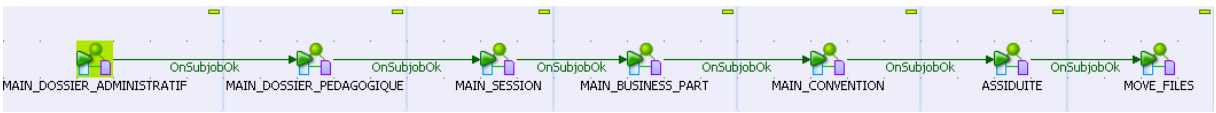


Figure 55 : Enchaînement de jobs de tous les domaines

4. Synthèse

Les exemples décrits dans ce chapitre expliquent de quelle manière j'ai pu exploiter les fonctionnalités de TALEND afin de répondre aux différents besoins tels que la migration, l'intégration et la synchronisation des données opérationnelles ainsi que le chargement des données dans l'entrepôt. Nous avons vu quels étaient les protocoles à respecter afin d'interfacer le système SAP avec l'entrepôt de données. Contrairement à la reprise de données, j'ai structuré mes programmes afin d'ils réalisent des tâches élémentaires dans chaque domaine fonctionnel. Puis je les ai incorporés à des programmes plus riches réalisant l'intégration complète des données. En procédant ainsi, on gagne en performance et cela facilite la maintenance et les évolutions. L'ensemble des données utiles à l'établissement présentes dans le nouveau progiciel de scolarité ne sont pas encore entièrement interfacées avec l'entrepôt.

VI. Conclusion

L'entrepôt de données a pour but l'aide à la décision et la qualité des données présentées à l'utilisateur est primordiale. Le nettoyage des données nous a permis d'identifier les données incorrectes, incomplètes et de les remplacer avant de les exploiter. Cette mise en cohérence du système d'information repose notamment sur l'usage de référentiel commun à tout l'établissement. De plus Il existe désormais un historique des données de l'établissement. L'entrepôt de données est un nouveau secteur d'activité permettant la centralisation des données de l'établissement et la production de rapport à l'aide d'outils de « reporting » comme Power pivot de Microsoft récemment mis en place par nos équipes.

Notre principal objectif, la reprise de données de scolarité, est un succès et les inscriptions de la rentrée 2010-2011 ont pu démarrer normalement avec le nouveau progiciel de scolarité. L'interfaçage de l'entrepôt de données avec ce progiciel permet aujourd'hui d'alimenter celui-ci, chaque jour, avec les données saisies par les gestionnaires de la scolarité. De nombreuses applications comme la listes des inscrits ou encore les attestations de réussite aux examens sont alimentées avec ces données. De plus, des fichiers contenant une synthèse des données sont générés à partir des données présentes dans l'entrepôt puis sont mis à la disposition des services qui le souhaitent.

Des évolutions sont d'ores et déjà envisagées. Lors de la mise en place de l'entrepôt nous avons choisi, pour des raisons de coût et de mise en œuvre, de stocker nos données dans une base MySQL. En raison du volume croissant de données il est en projet de migrer l'entrepôt de données vers un serveur de données Microsoft. En effet, le CNAM se tourne de plus en plus vers le moteur SQL Server 2000 pour supporter ses applications. Lors de cette migration je devrai revoir l'ensemble des processus d'alimentation et d'extraction des données de l'entrepôt afin de les adapter à attaquer ce type de base de données. Par ailleurs, la version 2 du logiciel de scolarité s'accompagne de la mise en place d'un nouveau portail d'inscription en ligne reposant sur le produit SharePoint de Microsoft. Il faudra certainement l'interfacer avec l'entrepôt de données. Enfin il est prévu d'intégrer les

données de scolarité des différents centres régionaux afin de fournir des données consolidées au niveau national.

Aujourd'hui je réalise qu'il m'aurait été difficile de mener à bien mes tâches dans ce projet sans de solides bases théoriques et de bonnes pratiques de travail acquises tout au long de mes études au C.N.A.M. de Paris. Je suis heureux d'avoir pu contribuer à la réussite d'un projet long et ambitieux au sein de l'établissement qui m'a formé. J'espère avoir doté celui-ci d'un entrepôt de données qui répond aux attentes des dirigeants et surtout d'avoir consolidé les données de tous ses auditeurs dans ce dernier.

Sur le plan professionnel le projet m'a bien sûr permis de me familiariser avec certains concepts et outils liés à l'informatique décisionnelle. Je me suis documenté sur le sujet et j'ai été formé à l'utilisation des ETL. Il m'est désormais possible de profiter au mieux de cet outil notamment lors de la manipulation de données et la production de rapport. Je pense qu'actuellement je suis la seule personne de mon service pouvant apporter une réelle expertise dans ce domaine. De plus, il est toujours enrichissant de participer à un projet important en coordination entre les personnels du CNAM et les équipes d'une société de service. Il faut suivre des spécifications fonctionnelles, rédiger des documents de travail et de suivis de charge, préparer des interventions orales lors des réunions avec les différents acteurs.

Dernièrement, j'ai été formé sur SAP et plus particulièrement à la programmation ABAP afin de renforcer les équipes de développement de la version 3 du logiciel de scolarité. A partir de l'année prochaine, je vais travailler en collaboration avec les équipes de la société Logica notamment afin de développer certains « web services » permettant l'interaction entre le portail SharePoint et le système SAP. Cette démarche s'inscrit dans une logique de transfert de compétences aux équipes de la DSI du CNAM afin de devenir, à terme, autonome sur la maintenance du produit.

Bibliographie

1. **Kimball, Ralph and Reeves, Laura.** *Concevoir et déployer un Data Warehouse.* s.l. : Eyrolles, 2000.

2. A definition of Data Warehousing. *Intranet Journal.* [Online] [Cited: 06 23, 2011.] http://www.intranetjournal.com/features/datawarehousing.html.

3. *Entrepôts de données.* **CHRISMENT, Claude, et al., et al.** s.l. : Techniques de l'Ingénieur, 2005.

4. **Scott, Arnett.** Data Warehousing. *The key for a successful implementation white paper.* 2010.

5. Data mart. *Wikipedia.* [Online] [Cited: 06 2001, 27.] http://fr.wikipedia.org/wiki/Datamart.

6. **Ziegler, Patrick and Dittrich , Klaus R.** *Three decades of Data Integration: all problems solved.* [Online] 2004. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.6593.

7. Les offres d'ETL. *Site du journal du Net.* [Online] 2006. [Cited: septembre 5, 2011.] http://www.journaldunet.com/solutions/0702/070221-panorama-etl/1.shtml.

8. **Métais, Elisabeth and Sèdes, Florence.** Appariement d'informations dans les entrepôts de données : quelques approches pour le filtrage flexible. *Information-Interaction-Intelligence.* s.l. : Cépaduès-éditions, 2002, Vol. 2.

9. **Piattini, Mario , Calero, Coral and Genero, Marcela F.** *Information and database quality.* s.l. : The Kluwer International Series on Advances in Database Systems, 2002.

10. **Johnson, Théodor and Dasu, tamraparni.** *Exploratory data mining and data cleaning.* s.l. : Wiley, 2003.

11. *Approximate join concepts and techniques.* **KOUDAS, N and SRIVASTAVA, D.** 2005. International conference on very large database.

12. **NAVARRO , G.** *A guided tour to approximate string matching.* s.l. : ACM Computer Surveys, 2001.

13. **Bokun, Michele and Taglienti , Carmen.** Incremental Data Warehouse Updates. *Information Management Magazine.* 1998.

14. **Franco, Jean-michèle.** *Le Data Warehouse.* s.l. : Eyrolles, 1997.

15. *Potter'sWheel: An Interactive Data Cleaning System.* **Raman, Vijayshankar and Hellerstein, Joseph M.** 2001. International Conference on Very Large.

16. *LOF: Identifying density-based local outliers.* **BREUNIG, Mark, et al., et al.** 2000. Conference ACM SIGMOD. Vols. p. 93-104.

Annexe

Exemple de scripts réalisés lors de la recette des fichiers

```
#!/bin/bash

#################################################################
####                Variables                ###
#################################################################
path_logs="T:/SCOLARITE/fichiers/RECETTE/fichiers_logs"
path_fichiers="T:/SCOLARITE/fichiers/EXTRACTIONS_SAP/DOSSIER_ADMIN"

#################################################################
####  Concaténation des fichiers du dossier administratif ###
#################################################################

# Fichier ID
cat $path_fichiers/*DA_ID*.txt>>$path_logs/id.txt

# Fichier AD
cat $path_fichiers/*DA_AD*.txt>>$path_logs/ad.txt

# Fichier INE
cat $path_fichiers/*DA_INE*>>$path_logs/ine.txt

# Fichier AP
cat $path_fichiers/*DA_AP*>>$path_logs/ap.txt

# Fichier DE
cat $path_fichiers/*DA_DE*>>$path_logs/de.txt

#################################################################
####          Vérifications sur le fichier ID          ###
#################################################################

# ID Vérification de doublons sur le numéro étudiant
cat $path_logs/id.txt|cut -f1| sort|uniq -d>$path_logs/doublons_num_etudiant_id.txt
echo  "Fichier contenant les numéros des doublons">>$path_logs/doublons_num_id.txt
while read ligne_id
        do
        set $(echo $ligne_id)
        num_id=$(eval echo $1)
        echo -e "Erreur : le numéro étudiant $num_id possède des
doublons">>$path_logs/doublons_num_id.txt
done < $path_logs/doublons_num_etudiant_id.txt


# ID Vérification de doublons ayant les mêmes nom, nom marital, prenom, date de naissance
cat $path_logs/id.txt|cut -f5,6,7,9| sort|uniq -d>$path_logs/doublons_etudiant_id.txt
echo "Fichier contenant des doublons">>$path_logs/doublons_etudiant2_id.txt
```

```
while read ligne_etudiant
        do
        set $(echo $ligne_etudiant)
        echo -e "Erreur : l'étudiant $ligne_etudiant possède des doublons">>$path_logs/doublons_etudiant2_id.txt
done < $path_logs/doublons_etudiant_id.txt

# ID Vérification que la date début est toujours égale à '19000101'
awk 'BEGIN { print "On cherche les lignes dont la date (champ 2) est différente de 19000101"; FS="
        "}
        $2 != 19000101 { print "Erreur : date incorrecte ligne n°"NR":  "$0}
END { print "Vérification terminée"} ' $path_logs/id.txt  > $path_logs/date_id.txt

# ID Vérification sur les tailles des noms et prénoms
awk -F " " 'length($5) > 40 {print "Erreur sur la taille du nom ligne n°"NR":  "$0}' $path_logs/id.txt  > $path_logs/taille_id.txt
awk -F " " 'length($6) > 40 {print "Erreur sur la taille du nom de naissance ligne n°"NR":  "$0}' $path_logs/id.txt  >> $path_logs/taille_id.txt
awk -F " " 'length($7) > 40 {print "Erreur sur le prénom ligne n°"NR":  "$0}' $path_logs/id.txt  >> $path_logs/taille_id.txt
awk -F " " 'length($8) > 40 {print "Erreur sur les prénoms ligne n°"NR":  "$0}' $path_logs/id.txt  >> $path_logs/taille_id.txt

# ID Vérification sur la présence des champs obligatoires
awk -F " " 'length($1) == 0 {print "Erreur sur la présence du numéro étudiant ligne n°"NR":  "$0}' $path_logs/id.txt  > $path_logs/obligatoire_id.txt
awk -F " " 'length($2) == 0 {print "Erreur sur la présence de la date début ligne n°"NR":  "$0}' $path_logs/id.txt  >> $path_logs/obligatoire_id.txt
awk -F " " 'length($5) == 0 {print "Erreur sur la présence du nom ligne n°"NR":  "$0}' $path_logs/id.txt >> $path_logs/obligatoire_id.txt
awk -F " " 'length($6) == 0 {print "Erreur sur la présence du nom de naissance début ligne n°"NR": "$0}' $path_logs/id.txt  >> $path_logs/obligatoire_id.txt
awk -F " " 'length($7) == 0 {print "Erreur sur la présence du prénom début ligne n°"NR":  "$0}' $path_logs/id.txt  >> $path_logs/obligatoire_id.txt

# ID Vérification que le code sexe est à 1 ou 2
awk 'BEGIN { print "On cherche les lignes dont le code sexe est différent de 1 ou 2"; FS="          "}
        $3!="1" && $3!="2" { print "Erreur : code sexe incorrect ligne n°"NR":  "$0}
END { print "Vérification terminée"} '  $path_logs/id.txt  > $path_logs/code_sexe_id.txt

# ID Vérification que le code civilité est à 1, 2 ou 3
awk 'BEGIN { print "On cherche les lignes dont le code civilité est différent de 1, 2 ou 3"; FS="     "}
        $4!="1" && $4!="2" && $4!="3" { print "Erreur : code civilité incorrecte ligne n°"NR":  "$0}
END { print "Vérification terminée"} '  $path_logs/id.txt  > $path_logs/code_civilite_id.txt

# ID Vérification que le code INSEE existe quand le pays est FR
awk 'BEGIN { print "On cherche les lignes dont le code insee (champ 10) existe pas quand le pays est FR"; FS="          "}
        $10=="" && $11=="FR"{ print "Erreur : code INSEE manquant alors que le pays est FR ligne n°"NR":  "$0}
END { print "Vérification terminée"} '  $path_logs/id.txt  > $path_logs/code_insee_id.txt

# ID Vérification que la date de naissance est comprise entre 18500101 et 19960101
awk 'BEGIN { print "On cherche les lignes dont la date (champ 9) est comprise entre 19000101 et 19960101"; FS=" "}
        $9 < 18500101 || $9 > 19960101 { print "Erreur : date incorrecte ligne n°"NR": "$0}
```

```
END { print "Vérification terminée"} '  $path_logs/id.txt  > $path_logs/date_naissance_id.txt

# ID Vérification que le code insee est compris entre 01000 et 99999
awk 'BEGIN { print "On cherche les lignes dont le code INSEE (champ 10) est supérieur à 99999"; FS="
        "}
        ($10 >99999 || $10 <1000) && $10 > 0 && $12!="2B" && $12!="2A"{ print "Erreur : code
INSEE incorrect ligne n°"NR": "$0}
END { print "Vérification terminée"} '  $path_logs/id.txt  > $path_logs/code_insee2_id.txt

# ID Vérification que le code département est bien inférieur à 100
awk 'BEGIN { print "On cherche les lignes dont le code département (champ 12) est inférieur à 100";
FS="    "}
        $12 > 100 && $12!="2A" && $12!="2B" && $12!=971 && $12!=972 && $12!=973 &&
$12!=974 && $12!=975 && $12!=976 && $12!=986 && $12!=987 && $12!=988{ print "Erreur : code
département incorrect ligne n°"NR":   "$0}
END { print "Vérification terminée"} '  $path_logs/id.txt  > $path_logs/code_departement_id.txt

# ID Vérification que le code département existe quand le pays est FR
awk 'BEGIN { print "On cherche les lignes dont le code département (champ 12) existe pas quand le
pays est FR"; FS="          "}
        $12=="" && $11=="FR" { print "Erreur : code département manquant alors que la pays est FR
ligne n°"NR":   "$0}
END { print "Vérification terminée"} '  $path_logs/id.txt  > $path_logs/code_departement2_id.txt

# génération du fichier de log 'log_id.txt'
cat $path_logs/doublons_num_id.txt|grep -i 'erreur'> $path_logs/log_id.txt
cat $path_logs/doublons_etudiant2_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/date_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/code_sexe_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/code_civilite_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/taille_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/obligatoire_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/date_naissance_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/code_insee_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/code_insee2_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/code_departement_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt
cat $path_logs/code_departement2_id.txt|grep -i 'erreur'>> $path_logs/log_id.txt

# suppression des fichiers temporaires
rm -f $path_logs/id.txt
rm -f $path_logs/code_departement_id.txt
rm -f $path_logs/code_departement2_id.txt
rm -f $path_logs/code_insee_id.txt
rm -f $path_logs/code_insee2_id.txt
rm -f $path_logs/date_naissance_id.txt
rm -f $path_logs/code_sexe_id.txt
rm -f $path_logs/code_civilite_id.txt
rm -f $path_logs/obligatoire_id.txt
rm -f $path_logs/taille_id.txt
rm -f $path_logs/date_id.txt
rm -f $path_logs/doublons_num_id.txt
rm -f $path_logs/doublons_num_etudiant_id.txt
rm -f $path_logs/doublons_etudiant_id.txt
rm -f $path_logs/doublons_etudiant2_id.txt

################################################################
```

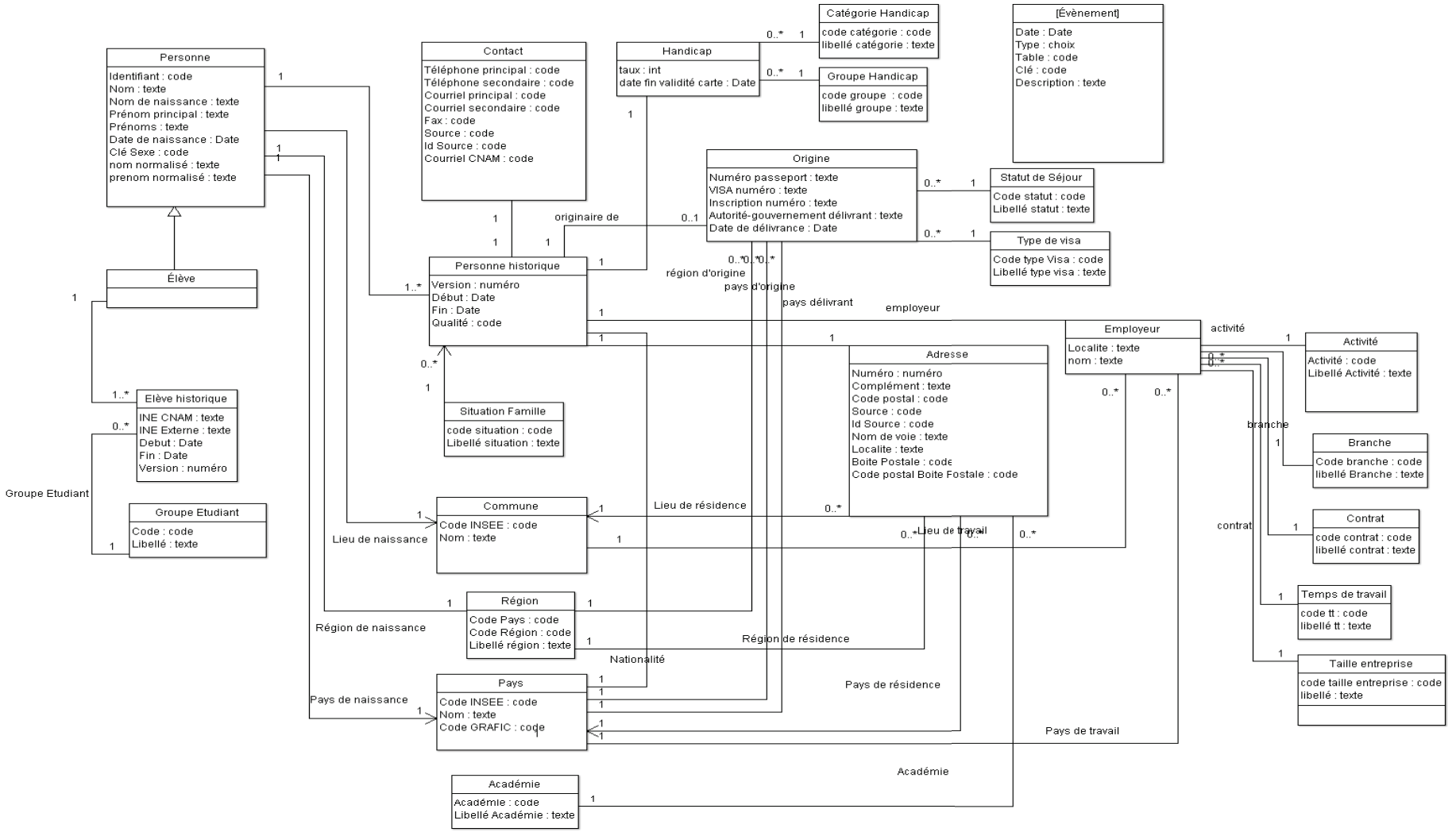
```
####          Vérifications sur le fichier AD          ###
#################################################################

# AD Vérification de doublons sur le numéro étudiant
cat $path_logs/ad.txt|cut -f1| sort|uniq -d>$path_logs/doublons_num_etudiant_ad.txt
echo "Fichier contenant les numéros doublons">>$path_logs/doublons_num_ad.txt
while read ligne_ad
        do
        set $(echo $ligne_ad)
        num_ad=$(eval echo $1)
        echo -e "Erreur : le numéro étudiant $num_ad possède des
doublons">>$path_logs/doublons_num_ad.txt
done < $path_logs/doublons_num_etudiant_ad.txt

# AD Vérification que la date début est toujours égale à '19000101'
awk 'BEGIN { print "On cherche les lignes dont la date (champ 2) est différente de 19000101"; FS="
        "}
        $2 != 19000101 { print "Erreur : date de début incorrecte ligne n°"NR":   "$0}
END { print "Vérification terminée"} ' $path_logs/ad.txt  > $path_logs/date_deb_ad.txt

# AD Vérification que la date fin est toujours égale à '99991231'
awk 'BEGIN { print "On cherche les lignes dont la date (champ 3) est différente de 99991231"; FS="
        "}
        $3 != 99991231 { print "Erreur : date de fin incorrecte ligne n°"NR":   "$0}
END { print "Vérification terminée"} ' $path_logs/ad.txt  > $path_logs/date_fin_ad.txt

# ID Vérification sur les tailles des différents champs
awk -F " " 'length($4) > 60 {print "Erreur sur la taille du champs rue ligne n°"NR":     "$0}'
$path_logs/ad.txt  > $path_logs/taille_ad.txt
awk -F " " 'length($7) > 40 {print "Erreur sur la taille de la localité ligne n°"NR":     "$0}'
$path_logs/ad.txt  >> $path_logs/taille_ad.txt
awk -F " " 'length($16) > 241 {print "Erreur sur le mail1 ligne n°"NR":   "$0}' $path_logs/ad.txt  >>
$path_logs/taille_ad.txt
awk -F " " 'length($19) > 241 {print "Erreur sur le mail2 ligne n°"NR":   "$0}' $path_logs/ad.txt  >>
$path_logs/taille_ad.txt

# génération du fichier de log 'log_ad.txt'
cat $path_logs/taille_ad.txt|grep -i 'erreur'> $path_logs/log_ad.txt
cat $path_logs/doublons_num_ad.txt|grep -i 'erreur'>> $path_logs/log_ad.txt
cat $path_logs/date_deb_ad.txt|grep -i 'erreur'>> $path_logs/log_ad.txt
cat $path_logs/date_fin_ad.txt|grep -i 'erreur'>> $path_logs/log_ad.txt

rm -f $path_logs/ad.txt
rm -f $path_logs/date_deb_ad.txt
rm -f $path_logs/date_fin_ad.txt
rm -f $path_logs/doublons_num_etudiant_ad.txt
rm -f $path_logs/doublons_num_ad.txt
rm -f $path_logs/taille_ad.txt

#################################################################
####          Vérifications sur le fichier INE          ###
#################################################################

# INE Vérification de doublons sur le numéro étudiant
cat $path_logs/ine.txt|cut -f1| sort|uniq -d>$path_logs/doublons_num_etudiant_ine.txt
echo "Fichier contenant les numéros doublons">>$path_logs/doublons_num_ine.txt
```

```
while read ligne_ine
        do
        set $(echo $ligne_ine)
        num_ine=$(eval echo $1)
        echo -e "Erreur : le numéro étudiant $num_ine possède des doublons">>$path_logs/doublons_num_ine.txt
done < $path_logs/doublons_num_etudiant_ine.txt

# INE Vérification de doublons sur le numéro INE
cat $path_logs/ine.txt|cut -f4| sort|uniq -d>$path_logs/doublons_num_ine_ine.txt
echo "Fichier contenant les numéros doublons">>$path_logs/doublons_ine_ine.txt
while read ligne_ine_ine
        do
        set $(echo $ligne_ine_ine)
        num_ine_ine=$(eval echo $1)
        echo -e "Erreur : le numéro ine $num_ine_ine possède des doublons">>$path_logs/doublons_ine_ine.txt
done < $path_logs/doublons_num_ine_ine.txt

# INE Vérification que la date début est toujours égale à '19000101'
awk 'BEGIN { print "On cherche les lignes dont la date (champ 2) est différente de 19000101"; FS="
        "}
        $2 != 19000101 { print "Erreur : date de début incorrecte ligne n°"NR":   "$0}
END { print "Vérification terminée"} ' $path_logs/ine.txt  > $path_logs/date_deb_ine.txt

# INE Vérification que la date fin est toujours égale à '99991231'
awk 'BEGIN { print "On cherche les lignes dont la date (champ 3) est différente de 99991231"; FS="
        "}
        $3 != 99991231 { print "Erreur : date de fin incorrecte ligne n°"NR":   "$0}
END { print "Vérification terminée"} ' $path_logs/ine.txt  > $path_logs/date_fin_ine.txt

# INE Vérification sur les tailles des différents champs
awk -F " " 'length($4) != 11 {print "Erreur sur la taille du code INE n°"NR":   "$0}' $path_logs/ine.txt  >
$path_logs/taille_ine.txt

# INE Vérification que le code INE est différent de '11111111111'
awk 'BEGIN { print "On cherche les lignes dont le code INE (champ 4) est égal à 11111111111"; FS="
        "}
        $4 == 11111111111 { print "Erreur : code INE incorrecte ligne n°"NR":   "$0}
END { print "Vérification terminée"} ' $path_logs/ine.txt  > $path_logs/num_ine.txt

# génération du fichier de log 'log_ine.txt'
cat $path_logs/doublons_num_ine.txt|grep -i 'erreur'> $path_logs/log_ine.txt
cat $path_logs/date_deb_ine.txt|grep -i 'erreur'>> $path_logs/log_ine.txt
cat $path_logs/date_fin_ine.txt|grep -i 'erreur'>> $path_logs/log_ine.txt
cat $path_logs/doublons_ine_ine.txt|grep -i 'erreur'>> $path_logs/log_ine.txt
cat $path_logs/num_ine.txt|grep -i 'erreur'>> $path_logs/log_ine.txt

rm -f $path_logs/doublons_num_etudiant_ine.txt
rm -f $path_logs/doublons_num_ine.txt
rm -f $path_logs/doublons_num_ine_ine.txt
rm -f $path_logs/doublons_ine_ine.txt
rm -f $path_logs/taille_ine.txt
rm -f $path_logs/date_deb_ine.txt
rm -f $path_logs/date_fin_ine.txt
rm -f $path_logs/ine.txt
```

```
rm -f $path_logs/num_ine.txt

#################################################################
####          Vérifications sur le fichier AP          ###
#################################################################

# AP Vérification de doublons sur le numéro étudiant
cat $path_logs/ap.txt|cut -f1| sort|uniq -d>$path_logs/doublons_num_etudiant_ap.txt
echo "Fichier contenant les numéros doublons">>$path_logs/doublons_num_ap.txt
while read ligne_ap
        do
        set $(echo $ligne_ap)
        num_ap=$(eval echo $1)
        echo -e "Erreur : le numéro étudiant $num_ap possède des doublons">>$path_logs/doublons_num_ap.txt
done < $path_logs/doublons_num_etudiant_ap.txt

# AP Vérification sur les tailles et présences des champs
awk -F " " 'length($4) == 0 {print "Erreur sur la présence du code activité ligne n°"NR":   "$0}' $path_logs/ap.txt  > $path_logs/taille_ap.txt
awk -F " " 'length($5) > 20 {print "Erreur sur la taille du champ employeur ligne n°"NR":   "$0}' $path_logs/ap.txt  >> $path_logs/taille_ap.txt
awk -F " " 'length($6) > 25 {print "Erreur sur la taille du champ entreprise ligne n°"NR":   "$0}' $path_logs/ap.txt  >> $path_logs/taille_ap.txt

# AP Vérification sur les valeurs du code activité
awk 'BEGIN { print "On cherche les lignes dont le code activité (champ 4) est hors référentiel"; FS="
        "}
        $4!=11 && $4!=21 && $4!=22 && $4!=23 && $4!=31 && $4!=33 && $4!=34 && $4!=35 && $4!=37 && $4!=38 && $4!=42 && $4!=43 && $4!=44 && $4!=45 && $4!=46 && $4!=47 && $4!=48 && $4!=49 && $4!=52 && $4!=53 && $4!=54 && $4!=55 && $4!=56 && $4!=62 && $4!=63 && $4!=64 && $4!=65 && $4!=67 && $4!=68 && $4!=69 && $4!=90 && $4!=91 && $4!=94 && $4!=95 && $4!=96 && $4!=97 && $4!=99 { print "Erreur : code activité hors référentiel ligne n°"NR":   "$0}
END { print "Vérification terminée"} '  $path_logs/ap.txt  > $path_logs/code_activite_ap.txt

# génération du fichier de log 'log_ap.txt'
cat $path_logs/doublons_num_ap.txt|grep -i 'erreur'> $path_logs/log_ap.txt
cat $path_logs/taille_ap.txt|grep -i 'erreur'>> $path_logs/log_ap.txt
cat $path_logs/code_activite_ap.txt|grep -i 'erreur'>> $path_logs/log_ap.txt

rm -f $path_logs/code_activite_ap.txt
rm -f $path_logs/doublons_num_etudiant_ap.txt
rm -f $path_logs/doublons_num_ap.txt
rm -f $path_logs/ap.txt
rm -f $path_logs/taille_ap.txt

#################################################################
####          Vérifications sur le fichier DE          ###
#################################################################

# DE Vérification de doublons sur le numéro étudiant
cat $path_logs/de.txt|cut -f1| sort|uniq -d>$path_logs/doublons_num_etudiant_de.txt
echo "Fichier contenant les numéros doublons">>$path_logs/doublons_num_de.txt
while read ligne_de
        do
        set $(echo $ligne_de)
```

```
        num_de=$(eval echo $1)
        echo -e "Erreur : le numéro étudiant $num_de possède des doublons">>$path_logs/doublons_num_de.txt
done < $path_logs/doublons_num_etudiant_de.txt

# DE Vérification que la date début est toujours égale à '19000101'
awk 'BEGIN { print "On cherche les lignes dont la date (champ 2) est différente de 19000101"; FS="
        "}
        $2 != 19000101 { print "Erreur : date de début incorrecte ligne n°"NR":   "$0}
END { print "Vérification terminée"} ' $path_logs/de.txt  > $path_logs/date_deb_de.txt

# DE Vérification que la date fin est toujours égale à '99991231'
awk 'BEGIN { print "On cherche les lignes dont la date (champ 3) est différente de 99991231"; FS="
        "}
        $3 != 99991231 { print "Erreur : date de fin incorrecte ligne n°"NR":   "$0}
END { print "Vérification terminée"} ' $path_logs/de.txt  > $path_logs/date_fin_de.txt

# DE Vérification sur le champ groupe
awk -F " " '$4 != "SALA" && $4 != "MERE" && $4 != "ETUD" && $4 != "RETR" && $4 != "RENE" && $4
!= "REPE" {print "Erreur le code du groupe existe pas n°"NR":     "$0}' $path_logs/de.txt  >
$path_logs/taille_de.txt

# génération du fichier de log 'log_de.txt'
cat $path_logs/doublons_num_de.txt|grep -i 'erreur'>$path_logs/log_de.txt
cat $path_logs/date_deb_de.txt|grep -i 'erreur'>> $path_logs/log_de.txt
cat $path_logs/date_fin_de.txt|grep -i 'erreur'>> $path_logs/log_de.txt
cat $path_logs/taille_de.txt|grep -i 'erreur'>> $path_logs/log_de.txt

rm -f $path_logs/doublons_num_etudiant_de.txt
rm -f $path_logs/doublons_num_de.txt
rm -f $path_logs/de.txt
rm -f $path_logs/date_deb_de.txt
rm -f $path_logs/date_fin_de.txt
rm -f $path_logs/taille_de.txt


exit 0
```

Figure 56 : Schéma des données administratives de l'auditeur dans l'entrepôt de données



Liste des figures

FIGURE 1 : ORGANISATION GENERALE DU CNAM DE PARIS 12
FIGURE 2 : ORGANIGRAMME DE LA DSI DU CNAM 13
FIGURE 3 : PERIMETRES DU S.I 17
FIGURE 4 : INTERACTIONS ENTRE LES DIFFERENTS LOGICIELS DU SI 17
FIGURE 5 : SOURCES DE DONNEES A INTEGRER DANS SISCOL 25
FIGURE 6 : INTERFACE ENTRE LES SOURCES D'INFORMATION ET LES DECIDEURS 27
FIGURE 7 : COMPOSANTS DE BASE D'UN ENTREPOT DE DONNEES 28
FIGURE 8 : LES FAITS 33
FIGURE 9 : LES DIMENSIONS 34
FIGURE 10 : EXEMPLE DE TABLE DES FAITS ET DIMENSIONS 35
FIGURE 11 : MODELISATION EN CONSTELLATION 36
FIGURE 12 : FENETRE DE COMPOSITION DE TALEND OPEN STUDIO 43
FIGURE 13 : LE REFERENTIEL TALEND 44
FIGURE 14 : « BUSINESS MODEL » 44
FIGURE 15 : JOB TALEND 45
FIGURE 16 : CREATION D'UNE ROUTINE 46
FIGURE 17 : PRINCIPAUX COMPOSANTS DE TALEND 47
FIGURE 18 : EXEMPLE DE COMPOSANT ET DE LEURS CONNEXIONS 48
FIGURE 19 : CORRESPONDANCE ATOMIQUE 49
FIGURE 20 : CORRESPONDANCE DE TYPE CALCUL 49
FIGURE 21 : CORRESPONDANCE DE TYPE VALEUR FIXE 50
FIGURE 22 : CORRESPONDANCE DE TYPE TRANSTYPAGE 50
FIGURE 23 : CORRESPONDANCES DE TYPE REFERENCE 51
FIGURE 24 : CORRESPONDANCE DE TYPE CONCATENATION 51
FIGURE 25 : LE CONSTRUCTEUR D'EXPRESSION 52
FIGURE 26 : DESCRIPTION D'UN COMPOSANT 53
FIGURE 27 : ARBORESCENCE DES FICHIERS D'UN COMPOSANT 53
FIGURE 28 : PROCESSUS DE GENERATION DE CODE JAVA 54
FIGURE 29 : BUSINESS MODEL REPRISE DE DONNEES 57
FIGURE 30 : PROCESSUS DE MISE NE QUALITE DES DONNEES 60
FIGURE 31 : MODELE DE DONNEE DU DOSSIER ADMINISTRATIF D'UN ELEVE 63
FIGURE 32 : CORRESPONDANCE DES DONNEES 75
FIGURE 33 : MISE EN QUALITE DES CRITERES D'UNICITE 76

FIGURE 34 : FILTRAGE SUR UNE EXPRESSION REGULIERE 77
FIGURE 35 : DEDOUBLAGE SIMPLE DE LA SOURCE 77
FIGURE 36 : DETECTION DES DOUBLONS PAR JOINTURE 78
FIGURE 37 : CORRECTION DE LA VILLE DE NAISSANCE 79
FIGURE 38 : CORRESPONDANCE CODE INSEE DU PAYS 81
FIGURE 39 : MISE EN QUALITE DES VILLES ET DES PAYS 82
FIGURE 40 : GESTION DES DOUBLONS 83
FIGURE 41 : MISE EN QUALITE DU DOSSIER PEDAGOGIQUE DE L'INTEC 84
FIGURE 42 : VALIDATION DES SM A PARTIR DU REFERENTIEL DANS L'ENTREPOT 85
FIGURE 43 : AJOUT DES CODES DIPLOME 88
FIGURE 44 : REPRISE DES DIPLOMES OBTENUS 89
FIGURE 45 : REPRISE DES DIPLOMES EN COURS D'OBTENTION 89
FIGURE 46 : REPRISE DES DIPLOMES INFORMATIQUE 90
FIGURE 47 : PROCESSUS D'AJOUT D'UN NOUVEL ELEVE DANS L'ENTREPOT 96
FIGURE 48 : PROCESSUS D'AJOUT D'UNE ADRESSE 96
FIGURE 49 : PROCESSUS DE MIGRATION DES DONNEES 98
FIGURE 50 : JOB REPRISE IDENTITE 103
FIGURE 51 : CORRESPONDANCE DES DONNEES SOURCE VERS LE SCHEMA CIBLE 104
FIGURE 52 : ORGANISATION DES JOBS 111
FIGURE 53 : IMPORTATION DES DONNEES D'ADRESSE 112
FIGURE 54 : ENCHAINEMENT DES JOBS DU DOSSIER ADMINISTRATIF 112
FIGURE 55 : ENCHAINEMENT DE JOBS DE TOUS LES DOMAINES 112
FIGURE 56 : SCHEMA DES DONNEES ADMINISTRATIVES DE L'AUDITEUR DANS L'ENTREPOT DE DONNEES 125

Liste des tableaux

TABLEAU 1 : COMPARAISON DE QUELQUES ETL DU MARCHE 40
TABLEAU 2 : COMPARAISON DES ETL DU MARCHE 41
TABLEAU 3 : EXEMPLE DE VIOLATION DE CONTRAINTES AU NIVEAU SCHEMA D'UNE SOURCE 67
TABLEAU 4 : EXEMPLES DE VIOLATION DE CONTRAINTES AU NIVEAU « INSTANCE » D'UNE SOURCE 68
TABLEAU 5: PROBLEMES MULTI-SOURCES AU NIVEAU INSTANCE 69
TABLEAU 6 : METHODE DE RECHERCHE DE DOUBLONS 71
TABLEAU 7 : IMPACT DE LA MISE EN QUALITE SUR LA REPRISE DE DONNEES ELEVES 97
TABLEAU 8 : SCHEMA DU FICHIER IDENTITE 102
TABLEAU 9 : LISTE LES 15 GROUPES DE DONNEES DU DOSSIER ELEVE 110

Résumé

Le but de ce mémoire est de présenter les concepts d'un entrepôt de données ainsi que les outils ETL permettant la gestion des flux de données. Après y avoir présenté l'environnement fonctionnel du projet de scolarité SISCOL du CNAM de Paris, j'y traite des problématiques de migration de données ainsi que des méthodes de mise en qualité des données. Puis j'y décris le déroulement du projet de l'analyse des besoins jusqu'à la réalisation.

Mots clés : Système d'information décisionnelle, ETL, entrepôt de données, reprise de données, migration de données, mise en qualité des données, SAP, CNAM.

Summary

The goal of this dissertation is to present the concepts of data warehouse and ETL tools for managing data flows. After having introduced the operating environment of the SISCOL project I deal with issues of data migration as well as methods for developing data quality. Then I describe the progress of the project from analysis to implementation.

Key words: Decision support system, ETL, data warehouse, data recovery, data migration, data quality, SAP, CNAM.