Amélioration de la qualité des données : correction sémantique des anomalies inter-colonnes

Houda Zaidi

le cnam

École doctorale Informatique, Télécommunications et Électronique (Paris)
Centre d'Études et de Recherche en Informatique et Communications

École doctorale Informatique, Université de Manouba (Tunisie)
Laboratoire de Recherche en génie logiciel, Applications distribuées, systèmes Décisionnels et Imagerie intelligente

THÈSE DE DOCTORAT

présentée par : **Houda ZAIDI**

soutenue le : **01 Février 2017**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Spécialité : **Informatique**

Amélioration de la qualité des données

Correction sémantique des anomalies inter-colonnes

THÈSE DIRIGÉE PAR

M. POLLET Yann	*Professeur des Universités, CNAM Paris*
M. BOUFARES Faouzi	*MCF HDR, Université Sorbonne Paris Cité*
M. KRAIEM Naoufel	*MCF HDR, Université de la Manouba, Tunisie*

RAPPORTEURS

Mme. SASSI HIDRI Minyar	*MA HDR, Université de Tunis, Tunisie*
M. CÉRIN Christophe	*Professeur des Universités, Université Sorbonne Paris Cité*

PRÉSIDENT

M. BARKAOUI Kamel	*Professeur des Universités, CNAM Paris*

EXAMINATEURS

Mme. DENECKERE Rebecca	*MCF, Université Paris-1-Panthéon-Sorbonne*
M. CARDON Alain	*Professeur des Universités, Institut National des Sciences Appliquées de Rouen*
M. CORREIA Sebastiao, Membre invité	*Chef de projet- Qualité de données- Société TALEND*

Remerciements

Le présent mémoire reflète le fruit des efforts conjugués de plusieurs personnes. Il m'est alors très agréable d'exprimer ma reconnaissance auprès de toutes ces personnes, dont l'intervention au cours de ces années de thèse, a favorisé son aboutissement.

Je remercie tout particulièrement mes encadrants Monsieur Faouzi BOUFARES, Maître de Conférences HdR à l'Université Sorbonne Paris Cité Paris 13, Monsieur Yann POLLET, Professeur au Cnam Paris, et Monsieur Naoufel KRAIEM, Maître de Conférences HdR à l'Université de la Manouba Tunisie, pour leur encadrement, leur soutien, ainsi que pour leurs conseils instructifs durant toute la période de la thèse réalisée en co-tutelle entre la France et la Tunisie.

Je remercie aussi Monsieur Kamel BARKAOUI, Professeur au Cnam Paris et directeur de mon équipe de Recherche (Vespa), pour son soutien incessant et pour sa participation au jury.

Mes vifs remerciements s'adressent aussi aux membres du jury Madame Minyar SASSI HIDRI, Maître de Conférences à Université de Tunis et Monsieur Christophe CERIN, Professeur à l'Université Sorbonne Paris Cité Paris 13 qui ont accepté de rapporter sur ce mémoire et aux autres membres du jury Monsieur Alain CARDON, Professeur à Institut National des Sciences Appliqués de Rouen, Madame Rebecca DENECKERE, Maître de Conférences à Université Paris-1-Panthéon-Sorbonne et Monsieur Sebastiao CORREIA de la société TALEND qui sont mes examinateurs.

Je n'oublierai jamais ma famille élargie qui a toujours été là quand j'avais besoin d'elle : Mes bien aimés père et mère, mes frères et soeurs ainsi que toutes leurs familles.

Ces quelques années passées au Cnam m'ont permis de me rendre compte de la vie dans des laboratoires de recherche à savoir le Cedric du Cnam Paris et le Lipn de l'Université Paris 13. Je remercie ceux qui m'ont permis de progresser et qui m'ont donné les moyens de cette évolution. Un grand merci aux membres des deux laboratoires, je n'oublierai évidemment pas tous nos partenaires de la société Talend. Je souhaite à tous une bonne continuation.

Je remercie tous mes ami(e)s.

Résumé

La qualité des données présente un grand enjeu au sein d'une organisation et influe énormément sur la qualité de ses services et sur sa rentabilité. La présence de données erronées engendre donc des préoccupations importantes autour de cette qualité. Ce rapport traite la problématique de l'amélioration de la qualité des données dans les grosses masses de données. Notre approche consiste à aider l'utilisateur afin de mieux comprendre les schémas des données manipulées, mais aussi définir les actions à réaliser sur celles-ci. Nous abordons plusieurs concepts tels que les anomalies des données au sein d'une même colonne, et les anomalies entre les colonnes relatives aux dépendances fonctionnelles. Nous proposons dans ce contexte plusieurs moyens de pallier ces défauts en nous intéressons à la performance des traitements ainsi opérés.

Mots clés : Qualité des données, Dépendances fonctionnelles, Dépendances sémantiques, traitement des valeurs nulles, nettoyage de données

Abstract

Data quality represents a major challenge because the cost of anomalies can be very high especially for large databases in enterprises that need to exchange information between systems and integrate large amounts of data. Decision making using erroneous data has a bad influence on the activities of organizations. Quantity of data continues to increase as well as the risks of anomalies. The automatic correction of these anomalies is a topic that is becoming more important both in business and in the academic world.
In this report, we propose an approach to better understand the semantics and the structure of the data. Our approach helps to correct automatically the intra-column anomalies and the inter-columns ones. We aim to improve the quality of data by processing the null values and the semantic dependencies between columns.

Keywords : Data Quality, Functional Dependencies, Semantic Dependencies, Null Values, Data Cleaning, Big Data.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

Introduction générale

Sommaire

1.1 Introduction

Les données ont une grande valeur économique pour les entreprises et les organismes qui savent en tirer profit. Elles sont le plus souvent insuffisamment exploitées à l'heure actuelle. D'une part, certaines données sont structurées, mais cependant mal gérées et assez mal automatisées. D'autres part, des données en quantités beaucoup plus grandes, sont non structurées, « dormantes », c'est à dire non prises en considération et non conformes aux données de référence correspondantes.

Les données sont de plus en plus abondantes, volumineuses, hétérogènes et de plus en plus distribuées. Elles sont, par ailleurs, de qualités variées. Elles constituent aussi pour les entreprises et les organismes de grands risques tels que risques financier, juridique ou risques afférant à l'image de marque, et ce pour des raisons diverses dont la confidentialité et la qualité des données.

Dans ce cadre, une exploitation de données erronées, si elle n'apporte aucun bénéfice, peut même nuire à l'organisme.

Plusieurs études ont montré que les données manipulées contiennent potentiellement plusieurs types d'anomalies. Il est fréquent que les données invalides représentent un pourcentage important du volume global traité (en l'occurence plus de 50%) [Toulemonde, 2008].

La qualité des données est un concept qui prend de plus en plus d'importance dans les grands organismes. Des projets de « Master Data Management MDM » sont mis en place afin de former à une meilleure qualité des données. Dans certains contextes tel que celui des données massives, les Bases de Données sont en perpétuelle évolution. Elles sont caractérisées par une structure variable dans le temps, de nouvelles données arrivant constamment. Il est donc utile de veiller à la qualité des données stockées en cherchant à développer des outils performants capables de détecter et de corriger automatiquement les anomalies.

Il existe plusieurs types d'anomalies sur les données. On peut citer en particulier le cas des doublons, des données similaires (doublons non stricts), des données aberrantes, des données obsolètes, des données incohérentes ou encore des valeurs nulles. Le coût lié aux difficultés induites par ces anomalies peut être très élevé, et celles-ci ont par conséquent une grande influence sur l'activité des organisations [Samitsch, 2015].

Force est de constater que les outils de gestion de la qualité des données et les outils ETL (Extract-Transform-Load) existants exigent une connaissance des structures et des contraintes sur les données manipulées. Pourtant, en pratique, ce n'est pas toujours le cas dans les organisations. Ceci est dû, d'une part, à l'absence de connaissances sémantiques sur les données et, d'autre part, à la pauvreté de méta-données voire à leur absence totale. Ceci a pour conséquence plusieurs types d'anomalies lors de l'intégration des données.

1.2 Contexte

Nous avons développé ce travail sur la base d'expérimentations menées en liaison avec la société Talend [Ref a] qui nous a aidé à appréhender des problématiques concrètes. La première partie du projet consistait à traiter essentiellement les anomalies inter-lignes (les doublons et les « similaires ») en se basant sur le concept de « profilage de données ». Cette étape a été développée dans le cadre de la thèse de [BenSalem 2015]. Les anomalies inter-colonnes font l'objet du présent manuscrit et font suite à ce projet.

Talend est un éditeur reconnu d'outils d'intégration de données, « Open Source ». La société Talend fournit des logiciels aux entreprises dont celles faisant un usage extensif de leurs données dans les domaines de l'intégration de données, de la gestion des données, de la qualité de données et du Big Data.

Elle a été Créée en 2006 et dipose des filiales en Californie, en Amérique du Nord, en Europe et en Asie.

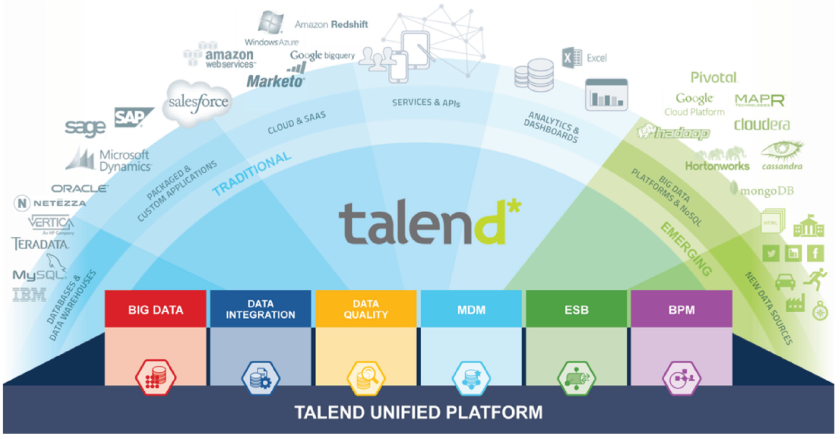


FIGURE 1.1 – Plate-forme de Talend

La figure ci-dessus illustre de manière imagée l'offre logicielle de la société Talend. Cette offre est constituée par une platforme logicielle intégrant le support aux différents aspects représentés, c'est à dire exploitation de données massives, intégration de données, gestion de la qualité des données.

1.3 Qualité des données

Données Source – DS : Patient.csv
175099943272264 ;- ;- ;M. ;M ;- ;Parisss ;France ;- ;-
180089987976564 ;CRI ;- ;Mme ;F ;- ;Paris ;Franc ;- ;-
165037895642322 ;AGRR ;- ;M. ;F ;15-mars-65 ;Loiret ;France ;Europe ;10/12/2014
180046378965464 ;CRP ;M. Martin DUPONT ;- ;M ;03-avr-80 ;Paris ;Fr ;Europe ;-
171038976542322 ;MGEN ;Mlle Anne MARTIN ; Mlle ;1 ;12/03/1971 ;Beijing ;Chine ;Asie ;-
278025125874563 ;- ;Mlle Karine LEBON ;Mlle ;1 ;- ;- ;China ;Afrique ;-
157054725912564 ;OTC ;M. Robert FORT ;M. ;0 ;- ;Frence ;Europe ;-
177125915879625 ;IPECA ;M. Simon GENEREUX ;M. ;0 ;- ;Bruxelle ;France ;- ;-
174046784763822 ;IPECA ;M. Simon GENEREUX ;M. ;- ;16/10/1996 ;Paris ;- ;Eurape ;01/02/2000
174046784763822 ;IPECA ;M. Simon GENEREUX ;M. ;- ;10-16-1996 ; Paris ;- ;Europe ;23/11/1015
283068794585464 ;IPECA ;Mlle Katia BON ;Mademoisele ;Femme ;24/06/1983 ;Calvados ;- ;- ;-
275478784581464 ;- ;Mlle Houda ZAIDI ;Mlle ;F ;30/02/2000 ;Vill ;Pai ;Conti ;-
285099935116964 ;- ;M. Adem LE BON ;M. ;0 ;- ;Pékin ;Chine ;Asia ;-
285099935116964 ;- ;M. Adem LE BON ;M. ;0 ;- ;Beijing ;Chin ;Asia ;-
285099935115564 ;- ;M. Robert LEBON ;M. ;0 ;- ;Bruxelle ;France ;- ;-
285099935115522 ;- ;M. Robert DUPONT ;M. ;0 ;- ;Bruxelle ;France ;- ;-

TABLE 1.1 – Extrait d'une source de données (DS) au format CSV

La problématique de la qualité des données prend de plus en plus de place dans le monde professionnel. La gestion et l'extraction de données sont fortement dépendantes de la qualité de celle-ci. Notre objectif est d'améliorer la qualité contextuelle des données (la qualité basée sur la sémantique de données) au sein d'une source de données (Data Source). Cette dernière peut être le résultat d'un processus d'intégration de plusieurs sources éventuellement héterogènes et de qualités variables. Nous considérons ainsi, dans la suite de ce manuscrit, que la source est « sans schéma » (sans méta-données). Elle peut être par exemple au fromat CSV [IBM] tel que presenté dans l'exemple ci-dessus.
On constate qu'une telle source de données n'a pas nécessairement de structure régulière (« elle est sans schéma de données »). La sémantique des données, leurs types ainsi que les contraintes sont alors inconnus. Plusieurs types d'anomalies peuvent alors exister. Ces anomalies peuvent être mieux perçues dans une représentation tabulaire, comme figuré ci-dessous (voir table 1.2, table 1.3).

175099943272264	-	-	M.	M	-	-	-	-	-
180089987976564	CRI	-	Mme	F	-	-	-	-	-
165037895642322	AGRR	-	M.	F	-	-	-	-	-
180046378965464	CRP	M. Martin DUPONT	-	M	-	-	-	-	-
171038976542322	MGEN	Mlle Anne MARTIN	Mlle	1	-	-	-	-	-
278025125874563	-	Mlle Karine LEBON	Mlle	1	-	-	-	-	-
157054725912564	OTC	M. Robert FORT	M.	0	-	-	-	-	-
177125915879625	IPECA	M. Simon GENEREUX	M.	0	-	-	-	-	-
174046784763822	IPECA	M. Simon GENEREUX	M.	-	-	-	-	-	-
174046784763822	IPECA	M. Simon GENEREUX	M.	-	-	-	-	-	-
283068794585464	IPECA	Mlle Katia BON	Mademoisele	Femme	-	-	-	-	-
275478784581464	-	Mlle Houda ZAIDI	Mlle	F	-	-	-	-	-
285099935116964	-	M. Adem LE BON	M.	0	-	-	-	-	-
285099935116964	-	M. Adem LE BON	M.	0	-	-	-	-	-
285099935115564	-	M. Robert LEBON	M.	0	-	Bruxelle	France	-	-
285099935115522	-	M. Robert DUPONT	M.	0	-	Bruxelle	France	-	-

TABLE 1.2 – Représentation tabulaire de la source de données DS

175099943272264	-	-	-	-	-	Parisss	France	-	-
180089987976564	-	-	-	-	-	Paris	Franc	-	
165037895642322	-	-	-	-	1996-16-10	Loiret	France	Europe	10/12/2014
180046378965464	-	-	-	-	03-avr-80	Paris	Fr	Europe	
171038976542322	-	-	-	-	12/03/1971	Beijing	Chine	Asie	
278025125874563	-	-	-	-	-	-	China	Afrique	
157054725912564	-	-	-	-	-	Pari	Frence	Europe	
177125915879625	-	-	-	-	-	Bruxelle	France	-	
174046784763822	-	-	-	-	16-10-1996	Paris	-	Eurape	01/02/2000
174046784763822	-	-	-	-	10-16-1996	Paris	-	Europe	23/11/2015
283068794585464	-	-	-	-	24/06/1983	Calvados	-	-	
275478784581464	-	-	-	-	30/02/2000	Vill	Pai	Conti	
285099935116964	-	-	-	-	-	Pékin	Chine	Asia	
285099935116964	-	-	-	-	-	Beijing	Chin	Asia	
285099935115564	-	-	-	-	-	Bruxelle	France	-	-
285099935115522	-	-	-	-	-	Bruxelle	France	-	-

TABLE 1.3 – Représentation tabulaire de la source de données DS (Suite)

Pour pouvoir qualifier une donnée d'incorrecte, il faudrait en effet l'évaluer dans son contexte et donc lui attribuer ainsi une sémantique. Plusieurs cas permettent d'illustrer nos propos dans l'exemple ci-dessus :

1. La chaîne de caractéres « Pari » (colonne 7) ne peut être considérée incorrecte **syntaxiquement** que si l'on sait qu'il s'agit du nom de **ville** « Paris ».

2. Les mots « Pékin » et « Beijing » désignent la même chose dans deux langues

diffŕentes s'il l'on sait qu'il s'agit de noms de **villes**. « Beijing » pourrait être considérée **sémantiquement** incorrect si la **langue** est le français.

3. Les trois chaînes de caractères « 16-10-1996 » , « 10-16-1996 » et « 1996-16-10 » pourraient représenter la même information de **type date**, définie par une **expression régulière**. Le format n'est pas le même. La chaîne de carcatères « 1996-10 » n'est pas une date.

   La colonne numérotée 5 ne contient que quelques valeurs distinctes à savoir M, F, Femme, 1 et 0. S'agit-il de données hétérogènes qui représentent le sexe d'une personne ?

   Les colonnes numérotées 8 et 9, comme d'autres colonnes, contiennent des valeurs nulles. Sont-elles vraiment inconnues et est-il possible de les remplacer par des valeurs sémantiquement correctes ?

4. Les deux derniers enregistrements de la source de données se ressemblent fortement si l'on considère les trois égalités suivantes : (i) « Pékin » est égal à « Beijing » , (ii) « Chin » est égal à « Chine » , et (iii) « Asie » est égal à « Asia » . Aucun algorithme de calcul de distance de similarité ne peut détecter que « Pékin » est égal à « Beijing » . La connaissance de la sémantique de la chaîne de caractères (Ville écrite dans deux langues différentes) est déterminante dans cette comparaison. L'èlimination des doubles et des similaires, si l'on possède cette connaissance, serait plus efficaces.

5. Les colonnes 4 et 5 peuvent être liées sémantiquement. Si la colonne 5 représente le sexe d'une personne alors elle est dépendante de la colonne 4 si cette dernière désigne la civilité. En conséquence, Les lignes 1 et 3 ne sont pas compatibles car la dépendance n'est pas vérifiée. Une reconnaissance contextuelle de la sémantique paraît ici indispensable pour diagnostiquer et corriger, si possible automatiquement les erreurs.

Le terme de la qualité des données désigne, d'une part, les caractéristiques essentielles des données (fiabilié, cohérence, complétude, pertinence, disponibilité, actualisation), et d'autre part, l'ensemble des processus qui permettent de garantir ces caractéristiques [Batini et al., 2009]. Nous rappelons brièvement, ci-dessous, les définitions des caractéristiques essentielles :

— Fiabilié : données correctes et crédibles.

— Cohérence : données toujours présentées dans le même format et non contradictoires.

— Complétude : les données nécessaires sont présentes (peu de valeurs absentes, inconnues).

— Pertinence : données intéressantes et utilisables.

— Disponibilité : données disponibles et accessibles.

— Actualisation : données non obsolètes, régulièrement mises à jour.

Le but est donc d'obtenir des données : (i) syntaxiquement correctes (c'est-à-dire sans fautes d'orthographes et conformes à la structure définie) et sémantiquement valides (c'est à dire compréhensibles et cohérentes) ; (ii) sans doublons et sans redondances ([Toulemonde 2008], [BenSalem 2015]).

La qualité des données peut être mesurée par des **indicateurs** (mesures) tels que le nombre de valeurs distinctes, ou encore le nombre de valeurs en doubles. Ces indicateurs permettent d'évaluer **des dimensions** de la qualité des données telles que l'intégrité, la standardisation ou la déduplication [Berti-Equille 2012] [Eaton et al. 2012]. Ces évaluations doivent être réactualisées regulièrement sur des données de plus en plus volumineuses, héterogènes, distribuées et de qualités variables. De nouvelles problématiques se posent concernant les masses des données [Zoghlami et al. 2016] et la qualité des données.

Le concept de Big Data (données massives) laisse appraître de nouveaux types de dimensions à savoir :

— Volume : le Volume concerne la taille des données. On parle de téraoctet ($To = 10^{12}$ octets), pétaoctet (Po=10^{15} octets), zettaoctet ($Zo = 10^{21}$ octets) et yottaoctet ($Yo = 10^{24}$ octets).

— Variété : la Variété signifie qu'il existe plusieurs types de données (structurées, semi-structurées, non structurées). Ces données sont issues de différentes sources hétérogènes (les réseaux sociaux, le web, les téléphones mobiles). Ceci les rend difficilement utilisables avec les outils traditionnels.

— Vitesse : La vitesse décrit la fréquence à laquelle les données sont générées. Les données massives doivent être traitées en quasi-temps réel.

— **Véracité** : La notion de véracité met en avant la dimension qualitative des données. Dans notre travail, nous nous intéressons à la véracité des données dans un contexte de Big Data (données massives, sans connaissances au préalable de leurs structures).

De nouveaux outils d'intégration de données hétérogènes [khalfallah 2006], [Hamdoun and Boufarès 2010] ont vu le jour. Ces outils n'ont pas suffisament intégré les concepts de qualité de données. Leurs évolutions font l'objet des travaux récents. La prise en compte de la sémantique devra être leur point fort quant au traitement de la qualité des données.

Dans la littérature, il existe plusieurs travaux de recherche qui visent l'identification et la classification des problèmes de qualité des données [Rahm and Do 2000] [Oliveira et al. 2005a] [Oliveira et al. 2005b] [Batini et al. 2009]. Ces derniers peuvent être classés en deux groupes :

— Les problèmes monosources : il existe plusieurs types d'anomalies dans les données [Boufarès et al. 2012] [Benjalloun et al. 2009] [Berti-Equille 2012] [Benjalloun et al. 2009] [BenSalem 2015]. On peut citer par exemple les valeurs nulles, les valeurs aberrantes, les valeurs obsolètes, les valeurs qui n'appartiennent pas au domaine de défintion ou qui violent les contraintes d'intégrité non définies ou non activées (les règles de gestion, les clés primaires, les clés étrangères, les dépendances fonctionnelles), les valeurs hétérogènes (les synonymes), les redondances (les doublons et les similaires).

— Les problèmes multi-sources : l'intégration des données peut, d'une part, accentuer les problèmes d'hétérogénéité (de représentation et de codification par exemple au niveau unités de mesure) en plus des problèmes monosources, et d'autre part, générer des doublons et/ou des similaires.

Il faut constater que ces problèmes peuvent être dûs à plusieurs facteurs qui n'ont pas été vraiment traités dans la littérature, à savoir l'ambiguité du schéma descriptif des données et surtout l'absence de sémantique.

En effet, la définition des données n'a pas toujours respecté rigoureusement les quatre étapes suivantes : (i) Nommer les colonnes (utiliser des noms standards des données qui

reflètent leur contenu, attention aux synonymes et aux homonymes), (ii) Définir les types syntaxiques des données (String, Number, Date) en précisant la longueur et le format, (iii) Déclarer les contraintes sur chaque colonne ou entre les colonnes, et enfin (iv) Commenter clairement les colonnes.

Les travaux relatifs à la qualité des données peuvent être classés selon quatre types d'approches [Berti-Equille 2012], qui sont 1) les approches préventives, 2) correctives, 3) diagnostiques, et 4) adaptatives. Les approches préventives regroupent les techniques permettant d'évaluer la qualité des modèles conceptuels, la qualité des développements logiciels et la qualité des processus employés pour le traitement des données. Les approches diagnostiques sont centrées sur des techniques statistiques, d'analyse et de fouille de données afin de détecter les anomalies sur les données. Les approches correctives sont basées sur des techniques de nettoyage de données. Enfin, les approches adaptatives consistent à adapter des traitements de vérification et de nettoyage lors de la médiation ou de l'intégration des données.

Nous proposons de résumer différemment les anomalies dans les données, sur la base de trois types d'anomalies :

- Les anomalies intra-colonne : hétérogénéité et standardisation des données, valeurs nulles.
- Les anomalies inter-lignes : les doublons et les similaires [Boufarès et al. 2012], [BenSalem 2015].
- Les anomalies inter-colonnes : les dépendances et les liens sémantiques qui peuvent exister entre les colonnes telles que les dépendances fonctionelles [Garnaud 2013] et les dépendances conditionnelles ainsi que les relations d'ordre [Diallo and Novelli 2010].

1.4 Problématique

Des études plus ou moins récentes ont montré le coût non négligeable de la non qualité des données [Samitsch, 2015]. Nous citons ci-dessous deux exemples réels présentés par un directeur Marketing Relationnel et Web de Conforama[1] et un responsable Data Warehouse Clients et Prospects Groupe Moniteur[2] : *« Pour mesurer l'impact de la non qualité des données, nous disposons de quelques indicateurs, comme le taux de PND (plis non distribués). Il est évident que plus ce taux augmente, plus le coût de la campagne marketing est lourd. Mais au delà de ce strict impact comptable, un client que l'on n'arrive pas à contacter, par exemple parce que son adresse a été mal mise à jour ou pas mise à jour après un déménagement, c'est un client que l'on risque de perdre. », « Un courrier qui n'arrive pas, c'est un client potentiel que l'on ne va pas relancer ou que l'on ne va pas démarcher, et cela a un impact direct sur le chiffre d'affaires ».*
Plusieurs études menées, par exemple, par le Datawarhousing Institute, pendant la première décennie des années 2000 ont montré que la non qualité des données peut engendrer un coût de plusieurs Milliards d'euros.

De nos jours, le volume des données explose. De surcroît, les données sont hétérogénes, issues éventuellement de plusieurs sources (structurées, semi-structurées et non structurées), et de niveaux de qualité différents. De ce fait, il est fréquent de manipuler des données sans avoir la connaissance de leurs structures ni leurs sémantiques.
En effet, les anomalies sur les données peuvent être dues à la pauverté sémantique de leurs descriptions ou même à l'absence de leurs descriptions. C'est l'hypothèse que nous adoptons. Les données sont ici sans schéma, telles que des données CSV par exemple.
Il est par conséquent intéressant de développer de nouveaux outils d'intégration et de manipulation de données permettant de mieux comprendre la sémantique et la structure des données manipulées.
Dans la littérature, les différentes études existantes ont porté sur les anomalies au sein d'une même colonne ou entre les lignes [Boufarès et al., 2013] [BenSalem, 2015]. Elles sont

1. http ://www.conforama.fr/
2. http ://www.lemoniteur.fr/

généralement basées sur une étape préalable d'homogénéisation et de standardisation des valeurs avant de traiter les anomalies. En revanche, peu de travaux ont abordé les liens sémantiques qui peuvent exister entre les colonnes.
En effet, la correction des anomalies au sein d'une même colonne pourrait donner de meilleur résultat dans la mesure où sa sémantique est connue. Par ailleurs, la découverte des liens sémantiques qui peuvent exister entre les colonnes peut permettre d'éviter la violation de différents types de contraintes de dépendances entre elles. La vérification des contraintes de dépendances à partir de grandes quantités de données permettra de corriger les anomalies telles que les valeurs nulles et certaines dépendances fonctionnelles.
Nous étudions dans le cadre de ce travail les contraintes de dépendances entre les différentes colonnes d'une même source de données. Nous nous intéressons à la fois à la détection de ces dépendances et aux anomalies causées par la violation de ces contraintes. Les contraintes de dépendances regroupent les différents types de dépendances fonctionnelles telles que les dépendances fonctionnelles exactes [Novelli and Cicchetti, 2001], les dépendances fonctionnelles approximatives [Simonenko and Novelli, 2012a], les dépendances fonctionnelles conditionnelles [Diallo and Novelli, 2010] et enfin les règles de gestion.

Les approches proposées dans la littérature pour la découverte des contraintes de dépendances entre les colonnes : (i) ne prennent pas en considération la sémantique des colonnes ; (ii) exigent la connaissance des structures (schémas) et des contraintes sur les données, alors que ces informations ne sont pas présentes dans la majorité des cas ; (iii) de surcroît, elles ne permettent pas de détecter toutes les dépendances valides surtout si les données contiennent des anomalies ou s'il s'agit des données issues de plusieurs sources hétérogènes qui peuvent contenir plusieurs valeurs nulles ; (iv) elles ne permettent pas la découverte de dépendances fonctionnelles pour les valeurs similaires telles que « $Fr \approx France$ » et « $Pari \approx Paris$ » .
Par ailleurs, nous avons constaté aussi que les temps d'exécution des algorithmes proposés sont linéaires vis à vis du nombre de tupples. En conséquent, ils ne permettent pas le traitement de dépendances fonctionnelles dans de grosses bases de données en un temps raisonnable.

Il faut constater que les outils d'intégration de données et de qualité des données qui existent de nos jours ne sont que trop « manuels ». Ils n'assistent que peu l'utlisateur pour diagnostiquer les erreurs. Par ailleurs, ils n'abordent pas vraiment les liens qui peuvent exister entre les colonnes. Seul l'outil de gestion de la qualité Talend permet d'aborder, depuis peu, l'aspect sémantique des données [BenSalem 2015].

Afin d'enrichir ces outils, nous nous posons en particulier les questions suivantes :

— **Comment reconnaître la sémantique et la structure de données. En d'autre terme, comment reconnaître le sens de chaque colonne et éventuellement les liens entre elles ?**

— **Comment vérifier les contraintes de dépendances à partir de grandes quantités de données ?**

— **Comment corriger les données invalides intra et inter-colonnes** ?

1.5 Objectifs

L'objectif de notre travail est donc d'assister l'utilisateur dans le processus de détection et de correction de certaines anomalies qui peuvent exister dans une source de données. Les données rassemblées peuvent être issues de l'intégration de différentes sources avec différents niveaux de discriptions des métadonnées. Ces dernières peuvent être totalement absentes et le plus souvent insuffisantes pour réfleter le contenu réel des données et donc de traiter les éventuelles anomalies.
Afin de mettre en oeuvre notre approche, nous avons développé un outil (figure 5.1) appelé iDQMS (intelligent Data Quality Management System), celui-ci permet :

— La reconnaissance sémantique des structures de données : cette étape consiste à déterminer le sens de chaque colonne d'une source de données. Nous utilisons pour

ce faire la sémantique existante dans les dictionnaires de données préétablis afin de reconnaître le type de données, la « catégorie » sémantique et la « sous-catégorie » (telle que la langue utilisée) de chaque colonne. Cette étape permet d'inférer aussi les dépendances qui peuvent exister entre les colonnes d'une source de données. Ces connaissances sont stockées dans le référentiel (les dictionnaires de données).

Le processus de la reconnaissance sémantique des structures des données prend en entrée la source de données (avec ou sans schéma) ainsi que les dictionnaires de données, et renvoie une nouvelle structure sémantique de cette source.

— L'établissement de plusieurs rapports qui portent sur les anomalies détectées.

— Le nettoyage de données : il s'agit d'exploiter les connaissances sémantiques déduites à partir de l'étape précédente pour corriger les anomalies au niveau intra et inter-colonnes et traiter certaines valeurs nulles. La première grande étape de nettoyage consiste à corriger les anomalies causées par la violation des contraintes de dépendances. La deuxième grande étape permet l'homogénéisation des données. Nous proposons des corrections de valeurs syntaxiquement et sémantiquement incorrectes (unification et standardisation des données en un seul format et une seule langue) au sein d'une même colonne.

Dans un contexte de volumétrie de bases de données (Big Data) [Eaton et al., 2012] et afin d'améliorer les performances des algorithmes très coûteux, nous utiliserons la technologie MapReduce [Dean and Ghemawat, 2004] afin de vérifier les dépendances découvertes et de corriger les anomalies causées par la violation de ces dépendances.

En résumant, notre objectif final est de contribuer au développement de nouveaux outils ETL qui n'imposent pas à l'utilisateur la connaissance des structures et des sémantiques des données manipulées. Ces outils devront permettre de guider les utlisateurs dans les processus de détection et de correction des anomalies. Les données rassemblées dans les bases et les entrepôts devraient avoir ainsi plus de crédibilité.

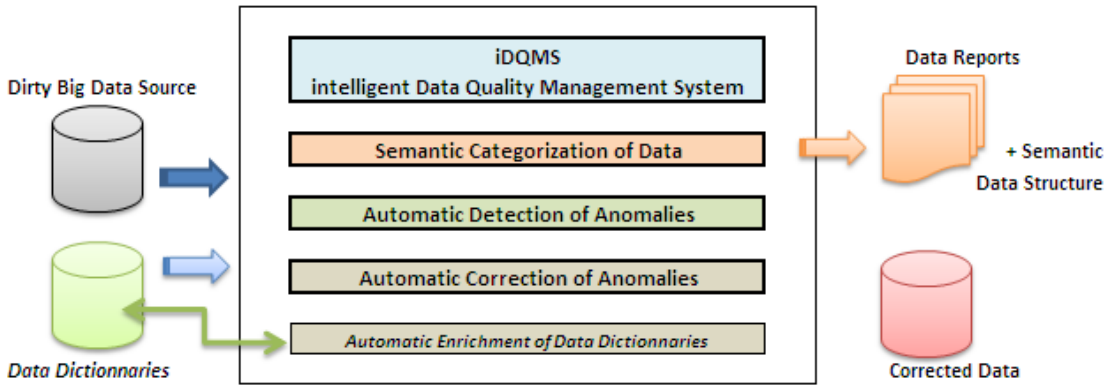


FIGURE 1.2 – L'outil iDQMS

1.6 Plan du document

Ce document est composé d'une introduction générale, d'un état de l'art, de plusieurs chapitres qui détaillent l'essentiel de notre contribution et d'une conclusion. À la suite de l'introduction au **chapitre 1**, le **chapitre 2** présente un état de l'art des travaux existants sur la qualité des données. Dans ce chapitre nous exposons les différentes anomalies pouvant exister dans une source de données. Celles-ci peuvent être présentes aussi bien dans les méta-données que dans les données elles-mêmes. Nous mettons l'accent sur le fait qu'un utlisateur n'est pas assisté dans sa double démarche tant pour détetecter et diagnostiquer les problèmes que pour les corriger. Un bilan récapitulatif des problématiques de la détection et de la correction des anomalies dans les données y est présenté.

Notre contribution est détaillée dans les chapitres suivants.

Le **chapitre 3** concerne la catégorisation sémantique des données. Elle permet, dans un premier temps, de reconnaître les sens des colonnes d'une source en exploitant la sémantique existante dans les données. Le but étant de reconstruire le schéma de description des données. Rappelons que la source de données est considérée sans schéma et que donc les méta-données sont indisponibles. Dans un second temps, la reconnaissance des sémantiques et des structures de données permet d'inférer les relations inter-colonnes.

Le **chapitre 4** concerne l'aide au diagnostic des anomalies dans les données et propose des

corrections automatiques de certaines d'entre elles. Le nettoyage des données consiste à corriger automatiquement les anomalies intra-colonne et inter-colonnes (homogénéisation et correction des anomalies causées par la violation de contraintes de dépendances).
Notre expérimentation est présentée dans le **chapitre 5**. L'accent est mis sur l'utlisation de les technologies du Big Data MapReduce afin d'améliorer les performances des algorithmes très coûteux de détection et de correction de dépendances fonctionnelles.
Un bilan récapitulatif des nouvelles fonctionnalités des outils ETL (Extact-Transform-Load) d'intégration de données et nos travaux futurs sont donnés en guise de **conclusion**.

Chapitre 2

Etat de l'art

Sommaire

2.1 Introduction

De nos jours, les données rassemblées dans les bases et les entrepôts de données sont très volumineuses, hétérogènes, distribuées et de différents niveaux de qualité. Ces données peuvent contenir différents types d'anomalies. On peut citer notamment les doublons, les données similaires (doublons non stricts), la violation des contraintes de dépendances, ou encore les valeurs nulles. En conséquence, l'extraction des connaissances et la prise de décisions à partir de ces données peuvent se révéler mauvaise. De surcroît, les coûts financiers qu'engendre la non-qualité des données, en ce qui concerne des prises de décision risquent d'être élevés.

Les outils de manipulation de données existants tels que les Système de Gestion de Bases de Données (SGBD) et les outils d'intégration (ETL) sont trop « manuels ». Ils exigent que l'utlisateur soit un expert ou une personne du domaine pour comprendre les structures et les contraintes sur les données afin de faire les conversions et les transformations nécessaires pour homogénéiser et corriger les données. Ces outils n'offrent ni assistance ni recommandation à l'utilisateur. Ils ne sont pas doté d'intelligence. Aucune cohérence globale des contraintes n'est étudiée. L'utilisateur est livré à lui même et doit détailler les actions correctives selon les interfaces offertes.

Dés lors, la manipulation des gros volumes de données dans les bases et les entrepôts de données nécessite le développement de nouveaux outils ETL. Ces derniers devront assister l'utilisateur dans (i) la démarche de reconnaissance des structures et des sémantiques des données manipulées en provenance des sources, (ii) le diagnostic des anomalies dans les données rassemblées, et enfin (iii) recommander des procédures de corrections.

Dans ce chapitre, nous commençons par rappeler un ensemble d'anomalies susceptibles d'être rencontrées dans les schémas et dans les données elles-mêmes. La deuxième partie fait le point sur les travaux présentés dans la littérature pour détecter et corriger ces anomalies [Deng et al. 2013] [Venetis et al. 2011] [Boufarès 2012].

Avant de commencer la partie état de l'art, nous rappelons quelques notions et définitions qui seront utilisées tout au long du manuscrit.

Définition 2.1 : domaine de données

Un domaine de données est l'ensemble de valeurs concrètes d'un type de données tels que : (i) les chaînes de caractères alphanumériques (String, char, varchar), (ii) les types numériques (Integer, number, real), (iii) les dates, (iv) les booléens ou encore (v) des listes et des intervalles de valeurs.
Un domaine de données est identifié par un nom. Soit D le nom attribué à un domaine.
Le nom de domaine est alors une chaîne de caractères. Un tel nom n'est pas toujours significatif du sens de la donnée hors des personnes ayant présidé à leur création. •

Exemple 2.1 : noms de domaine de données
- DomainName20 est le nom donné au type String de vingt caractères.
- CityName30 est le nom donné au type String de trente caractères.
- Age est le nom donné au type Integer de valeurs comprises entre 0 et 150.

Exemple 2.2 : définition d'un domaine de données
Le domaine de données de nom DomainName20 est composé de l'ensemble de toutes les chaînes de caractères de longueur maximale vingt.
Le domaine de données de nom CityName30 est composé de l'ensemble de toutes les chaînes de caractères de longueur maximale trente.
Il faut remarquer que dans ces deux ensembles de valeurs (ces deux domaines), certaines chaînes peuvent être non significatives et devraient être qualifiées sémantiquement incorrectes, alors qu'elles sont syntaxiquement correctes si leurs longueurs sont inférieurs ou égales à la taille maximale. Evidemment le nom du domaine ne donne aucune signification supplémentaire au contenu.

- DomainName20 ={Paris, Tunis, Cnam, B+, 31°C, Parisss, Hauspital HM}
- CityName30 ={Paris, Tunis, Cnam, B+, 31°C, Parisss, Hauspital HM}
- Age = {0, 1, 2, 3, ..., 150}

Remarques :

- Par abus de langage, on considèrera que le mot **colonne** signifie **attribut**. Elle est définie sur un domaine, celui-ci étant, le plus souvent, implicite. C'est ce qui prête à confusion. En effet :
- Les valeurs qui appartiennent à un domaine devraient être rattachées à un contexte afin d'être qualifiées de sémantiquement correctes ou pas.
- Les valeurs {Paris, Tunis} et {Cnam, B+, 31°C, Parisss, Hauspital HM} ne semblent pas appartenir à un même concept sémantique, alors qu'elles sont syntaxiquement correctes.

Soit $C = \{C_1, C_2, ..., C_n\}$ un ensemble de noms de colonnes. Chaque colonne est définie sur un domaine.

Définition 2.2 : source de données

Une source de données DS, définie sur les colonnes $\{C_1...C_n\}$, est un sous-ensemble du produit cartésien des domaines de définition de chacune des colonnes. Chaque sous-ensemble constitue une instance (c'est à dire une occurrence) de la source de données.
Il s'agit donc d'un ensemble de n-uplets $\{C_1...C_n\}$. Chaque n-uplet est un élément du sous-ensemble du produit cartésien des domaines.
On notera : t un n-uplet ; $t[C_i]$ une valeur v de la colonne C_i dans t. •

Exemple 2.3 : une source de données

Soit $C = \{Col_1, Col_2\}$ un ensemble de noms des colonnes définies respectivement sur les domaines DomainName20 et CityName30.

Données	
Parisss	France
Paris	Franc
Loiret	France
Paris	Fr
Beijing	Chine

TABLE 2.1 – Exemple d'une source de données, DS (données)

Remarque :

On notera que le contenu des colonnes ne reflète pas ici les sens attachés aux doamines !

Définition 2.3 : schéma d'une source de données

Un schéma est une définition de la structure complète d'une source de données [Menard 2008]. Il liste toutes les colonnes sur lesquelles porte la source de données.
Un schéma décrit toutes les propriétés d'une source à savoir : (i) les noms des colonnes, (ii) les aspects syntaxiques de chacune (c'est à dire les domaines syntaxiques ou les types de données), (iii) les aspects sémantiques des colonnes (les contraintes) et (iv) des commentaires éventuels attachés aux différentes colonnes.
Les contraintes (**Constraint**) définies sur les colonnes expriment des règles sémantiques. Elles peuvent être de plusieurs types : clé primaire (**Primary key**), unique (**Unique**), clé étrangère (**Foreign key**), appartenance à un intervalle ou à une liste (Check), vérification de certaines règles de gestion et des déclencheurs (**Triggers**). •

Notations :

Soit $C = \{C_1, C_2, ..., C_n\}$ un ensemble de noms de colonnes. C_i, i = 1 ; n. Les colonnes sont supposées ici toutes distinctes.
Soit $Dom = \{D_1, D_2, ..., D_n\}$ l'ensemble des domaines des colonnes.
D_i ne présente que le domaine syntaxique (type de données). Les domaines peuvent être ainsi les mêmes.

Soit $K = \{K_1, K_2, ..., K_n\}$ l'ensemble des ensembles de contraintes sur les colonnes colonnes C_i, i = 1 ; n.

K_i représente le(s) domaine(s) sémantique(s) d'une colonne.

Soit $Com = \{Com_1, Com_2, ..., Com_n\}$ l'ensemble des commentaires sur les colonnes.

On note un schéma :

$DS(C_1 : \{D_1, K_1, Com_1\}, C2 : \{D_2, K_2, Com_2\}, ..., C_n : \{D_n, K_n, Com_n\})$.

Remarques :

— Par abus de langage on utilise la notation $DS(C_1, C_2, ..., C_n)$ sachant que les domaines, les contraintes et les commentaires sont implicites. On note aussi DS(C). Une colonne C_i de DS peut être désignée par $DS.C_i$.

— Dans la pratique les schémas sont très mal renseignés (voir Table 2.3). Ainsi, les sources de données sont rarement accompagnées de descriptif complet de leurs contenus ainsi des anomalies peuvent exister dans les données.

Exemple 2.4 : exemples de schémas

— DS1(Fname, BDate, Country). DS1 est une source de données définie avec seulement les noms des colonnes. C'est le format le plus souvent utilisé lors de la définition d'une source. Aucune précision n'est donnée sur les colonnes.

— DS2(Name, BirthDate : {Date, « >01/01/1980 », //Date of birth}, Country). DS2 est définie avec des noms de colonnes significatifs et une contrainte et un commentaire pour la colonne BirthDate. Les colonnes Name et Country sont mal définies.

— DS3(COL1 : {String, clé primaire}, COL2 : {String}, COL3 :{String}, COL4 : {String}). DS3 est une source avec des noms de colonnes ambiguës (meaningless). Cependant, le type de données est présent pour chaque colonne et une contrainte est définie sur le premier attribut.

Nous adopterons la notation tabulaire suivante pour les schémas des sources de données.

Schéma de la source de données				
Nom de la Source	DS			
Noms des colonnes	C_1	C_2	...	C_n
Types de données	D_1	D_2	...	D_n
Contraintes	K_1	K_2	...	K_n
Commentaires	Com_1	Com_2	...	Com_n

TABLE 2.2 – Exemple de Schéma d'une source de données DS

Notons qu'en pratique, les sources ne sont pas toujours accompagnées de définitions complètes des colonnes. Le plus souvent, les domaines, les contraintes et les commentaires sont négligés.
Les noms de colonnes peuvent alors prêter à confusion. Ils ne reflètent en aucun cas les contenus de domaines de définition et à fortiori le sens de la donnée.

Exemple 2.5 : exemple d'une source mal définie

Schéma				
Nom de la Source	DS			
Noms des colonnes	COL1	COL2	COL3	COL4
Types de données	String	String	String	String
Contraintes	-	-	-	-
Commentaires	-	-	-	-

TABLE 2.3 – Exemple de Schéma ambigu

Données			
175099943272264	M	Parisss	France
180089987976564	Mme	Paris	Franc
165037895642322	Mlle	Loiret	France
180046378965464	M	Paris	Fr
171038976542322	Mlle	Beijing	Chine
278025125874563	Mlle	-	China
157054725912564	M	Pari	Frence
177125915879625	M	Bruxelle	France
174046784763822	M	Paris	Eurape
174046784763822	M	Paris	-
283068794585464	Mlle	Calvados	-

TABLE 2.4 – Exemple d'une source de données DS

Notre objectif est de rédecouvrir le schéma de données à partir des données elles-mêmes. On parle alors de *profilage de données* ou de *reconnaissance sémantique de données* ainsi que de leurs schémas, afin de corriger au mieux les anomalies.

Définition 2.4 : les métadonnées

Les métadonnées sont les données qui décrivent d'autre données (sémantiquement les données à propos des données)[Tanghe et al. 2016]. •

Elles permettent à un individu ou un ordinateur de comprendre le sens et l'organisation de données et facilitent leur intégration, leur collecte et leur partage [Tanghe et al. 2016].

Remarque :

Sur le plan conceptuel, il ne peut y avoir de données sans métadonnées ! En effet, les données n'ont de valeur que dans un contexte bien précis et l'ajout d'informations qui permettent de les identifier et de les localiser est essentiel [Tanghe et al. 2016]. Cependant, en pratique, les données sont rarement accompagnées des métadonnées ce qui rend difficile leurs exploitation et à fortiori l'amélioration de leurs qualité.

2.2 Anomalies dans les schémas de données et dans les données

Dans la littérature, il existe plusieurs travaux de recherche qui visent l'identification de différentes anomalies sur les données et sur leurs schémas (leurs définitions) [Rahm and Do 2000] [Oliveira et al. 2005a]

Ces anomalies peuvent être classées en deux groupes : les anomalies dans les métadonnées et celles dans les données. Nous classons ces dernières en trois grandes catégories à savoir les anomalies intra-colonne (Données malformatées, les valeurs nulles), celles inter-colonnes (les dépendances) et enfin celles entre lignes (les redondances) .
Nous allons en dresser ici un panorama synthétique.

2.2.1 Anomalies dans les schémas de données

Nous présentons ci-dessous des exemples d'anomalies dans les schémas de sources de données :

— Les noms des colonnes utilisés dans le schéma de données tels que Adr, Ad, Add, ZipC, ZC et les noms des sources de données sont insignifiants.
— Les domaines syntaxiques de données ne sont pas bien définis.
— Les contraintes sont le plus souvent absentes ou non activées.
— Les commentaires sont imprécis voire inexistants.

Les anomalies dans les schémas peuvent causer des anomalies lors de l'intégration de données. Les exemples ci-dessous montrent qu'en l'absence de contraintes clairement définies sur les données ou même on l'absence de commentaires, l'intégration de plusieurs sources peut construire des colonnes dont le contenu est totalement hétérogène et sémantiquement non signifiant.

Exemple 2.6 : intégration de données avec contraintes non définies
Soient DS1 et DS2 deux sources de données dont les schémas sont inconnus ou mal renseignés. La première contient la liste des clients d'une clinique (Client.csv) et la seconde les

patients d'une autre clinique (Patient.csv) :

Client(Id : {String(10)}, Coordonnées : {Sting(30)}, Profession : {String})

Patient(NP : {String}, Co {String}, Mutuelle : {String})

Schéma			
Nom de la Source	**DS1 : Client**		
Noms des colonnes	Id	Coordonnées	Profession
Types de données	String(10)	String(13)	String
Contraintes	-	-	-
Commentaires	-	-	-

TABLE 2.5 – Schéma de DS1 (Client.csv)

Données		
C1	06 89 56 13 25	Etudiant
C2	06 77 51 10 23	Enseignant
C3	06 77 51 10 26	Ingénieur

TABLE 2.6 – Extrait du fichier Client.csv

Schéma			
Nom de la source	**DS2 : Patient**		
Nom des colonnes	NP	Co	Mutuelle
Type de donnée	String	String	String
Contraintes	-	-	-
Commentaires	-	-	-

TABLE 2.7 – Schéma de DS2 (Patient.csv)

Données		
P1	houda@yahoo.fr	MGEN
P2	Bisson@cnam.fr	AGRR
P3	Dervaux@gmail.fr	OTC

TABLE 2.8 – Extrait du fichier Patient.csv

Les scripts SQL qui permettent de créer ces données au sein d'un Système de Gestion

de Base de Données Relationnel (SGBDR) sont :

```
CREATE TABLE DS1 (Id String(10), Coordonnées String(13) ,Profession String);
INSERT INTO DS1 VALUES ('C1', '06 89 56 13 25', 'Etudiant');
INSERT INTO DS1 VALUES ('C2', '06 77 51 10 23', 'Enseignant');
INSERT INTO DS1 VALUES ('C3', '06 77 51 10 26', 'Ingénieur');

CREATE TABLE DS2 (NP String, Co String ,Mutuelle String);
INSERT INTO DS1 VALUES ('P1', 'houda@yahoo.fr', 'MGEN');
INSERT INTO DS1 VALUES ('P2', 'Bisson@cnam.fr', 'AGRR');
INSERT INTO DS1 VALUES ('P3', 'Dervaux@gmail', 'OTC');
```

L'intégration des deux sources DS1 et DS2 s'obtient avec le script SQL ci-dessous :

```
CREATE TABLE DS (Id, Coordonnées, Profession) AS
(SELECT * FROM DS1 UNION ALL SELECT * FROM DS2);
```

Remarque :
Des choix arbitraires ont été faits sur les noms des colonnes résultats et leurs types respectifs. La seule vérification réalisée par les outils se place au niveau de la compatibilité syntaxique !
Le schéma du résultat DS de l'intégration des deux sources est :
DS(Id : {String}, Coordonnées : {String}, Profession : {String})

Schéma			
Nom de la source	**DS : Résultat de l'intégration de DS1 et DS2**		
Noms des colonnes	Id	Coordonnées	Profession
Types de données	String	String	String
Contraintes	-	-	-
Commentaires	-	-	-

TABLE 2.9 – Schéma de DS

Les schémas des deux sources étant mal renseignés, la définition du résultat est à la charge de l'utilisateur. Ainsi :
- La nouvelle colonne DS.Id est créée à partir de DS1.Id et DS2.NP. Le type String est retenu. On peut facilement constater que les données peuvent être tronquées si le type du résultat n'est pas pertinent.
- La nouvelle colonne DS.Coordonnées qui est la « fusion » des colonnes DS1.Coordonnées et DS2.Co, montre l'hétérogénéité sémantique du résultat. Il en est de même pour la colonne DS.Profession issue de l'union de DS1.Profession et DS2.Mutuelle.

Le résultat de l'intégration des colonnes DS1.Coordonnées et DS2.Co contient des anomalies. Des adresses mail et des numéros de téléphone sont présents dans la colonne DS.Coordonnées. Les outils ETL existants ne permettent pas de guider l'utilisateur dans le processus d'intégration. Il reste à la charge de l'utilisateur de reconnaître que la sémantique de la colonne DS1.Coordonnées contient des numéros de téléphone et que la sémantique de la colonne DS2.Co contient des adresses mail. Dès lors, on constate la nécessité de nouveaux outils qui permettront la reconnaissance automatique de la sémantique des colonnes afin d'assister le processus de l'intégration.
La table résultat de l'intégration est présentée ci-dessous. Tous les SGBD ainsi que tous les ETL existants autorisent ce type d'intégration sur deux sources sémantiquement incompatibles. La seule compatibilité considérée est celle des colonnes, prises deux à deux, doivent être syntaxiquement définies sur des domaines compatibles.

Données		
C1	06 89 56 13 25	Etudiant
C2	06 77 51 10 23	Enseignant
C3	06 77 51 10 26	Ingénieur
P1	houda@yahoo.fr	MGEN
P2	Bisson@cnam.fr	AGRR
P3	Dervaux@gmail.fr	OTC

TABLE 2.10 – Résultat de l'intégration de deux sources DS1 et DS2

Exemple 2.7 : intégration de données avec contraintes définies mais sans commentaires
Soient DS1 et DS2 deux sources de données dont les schémas sont mal renseignés. Elles contiennent les détails sur les clients d'une clinique (Client.csv) et les patients d'un centre médical (Patient.csv) :
DS1 (Taille :{[0..250]}, Sexe :{[F,M]})
DS2 (Taille :{[0..2,5]}, Sexe :{[0,1]})
Les deux colonnes DS1.Taille et DS2.Taille ne sont pas définies de la même manière. Il semblerait que DS1.Taille représente des valeurs numériques en centimètre alors que DS2.Taille représente des valeurs en mètre. Il en est de même pour la codification utilisée pour définir les colonnes DS1.Sexe et DS2.Sexe. Les données ne sont pas homogènes malgré les noms des colonnes qui sont les mêmes.
Par conséquent, pour intégrer les données des deux colonnes en une seule, l'utilisateur doit avoir des connaissances sur les domaines des sources et sur les transformations à faire pour standardiser et homogénéiser les données. L'utilisateur doit expliciter les transformations

nécessaires pour intégrer correctement les données.

Schéma		
Nom de la source	**DS1**	
Noms des colonnes	Taille	Sexe
Types de donnéeS	Number	String
Contraintes	[0..250]	{F,M}
Commentaires	-	-

TABLE 2.11 – Schéma de DS1

Schéma		
Nom de la source	**DS2**	
Noms des colonnes	Taille	Sexe
Types de donnée	Number	String
Contraintes	[0..2.5]	{0,1}
Commentaires	-	-

TABLE 2.12 – Schéma de DS2

Données	
160	M
190	F
180	M
170	F

TABLE 2.13 – Extrait de DS1

Données	
1.58	0
1.69	1
1.8	1
1.9	0

TABLE 2.14 – Extrait de DS2

Les scripts SQL qui permettent de créer ces données au sein d'un SGBD sont :

```
CREATE TABLE DS1
(Taille Number CHECK (Taille BETWEEN 0 AND 250),
Sexe CHAR(1) CHECK (SEXE IN ('F','M')));
INSERT INTO DS1 VALUES (160, 'M');
INSERT INTO DS1 VALUES (190, 'F');
INSERT INTO DS1 VALUES (180, 'M');
INSERT INTO DS1 VALUES (170, 'F');

CREATE TABLE DS2
(Taille Number CHECK (Taille BETWEEN 0 AND 2.5),
Sexe CHAR(1) CHECK (SEXE IN ('0','1')));
INSERT INTO DS2 VALUES (1.58, '0');
INSERT INTO DS2 VALUES (1.69, '1');
INSERT INTO DS2 VALUES (1.8, '1');
INSERT INTO DS2 VALUES (1.9, '0');
```

L'intégration des deux sources DS1 et DS2 s'obtient avec le script SQL ci-dessous :

```
CREATE TABLE DS (Taille,Sexe) AS
(SELECT * FROM DS1 UNION ALL SELECT * FROM DS2);
```

Le schéma du résultat DS de l'intégration des deux sources est :
DS(Taille : {Number}, Sexe : {CHAR})

Schéma		
Nom de la source	**DS= intégration(DS1,DS2)**	
NomS des colonnes	Taille	Sexe
Types de données	Number	String
Contraintes	-	-
Commentaires	-	-

TABLE 2.15 – Intégration des deux sources de données

Données	
160	M
190	F
180	M
170	F
1.58	0
1.69	1
1.8	1
1.9	0

TABLE 2.16 – Intégration sans transformation

Les schémas des deux sources étant mal renseignés, la définition du résultat est à la charge de l'utilisateur :
- La nouvelle colonne DS.Taille est créée à partir de DS1.Taille et DS2.Taille. Le type Number est retenu. N'ayant aucun commentaire sur les contraintes, celles-ci ne sont pas retenues dans les résultats. Il est même possible de garder celle où l'intervalle de définition est le plus large [0..250]. On peut facilement constater que le calcul de la moyenne de la Taille est erronée à cause de l'hétérogénéité des valeurs.
- La nouvelle colonne DS.Sexe qui est la « fusion » des colonnes DS1.Sexe et DS2.Sexe, montre l'hétérogénéité sémantique du résultat, encore faut-il trouver la correspondance entre les domaines de définitions des colonnes S1.Sexe et S2.Sexe.

Des transformations sont nécessaires afin d'unifier la représentation des données (codification, format).

La table résultat de l'intégration, qui devrait être générée, est présentée ci-dessous. Aucun ETL n'assiste l'utilisateur pendant l'étape de transformation pour standardiser et homogénéiser les résultats.

Données	
160	M
190	F
180	M
170	F
158	F
169	M
180	M
190	F

TABLE 2.17 – Intégration avec transformations

Force est de constater que les sources de données sont en général mal décrites, plusieurs sortes d'anomalies existant dans les données ce qui rend leurs exploitations très difficiles voire insignifiantes. Les algorithmes d'apprentissage lancés récemment sur de très gros volumes de données soulignent la nécessité absolue de nettoyer et corriger ces dernières.
Les paragraphes suivants citent plusieurs anomalies dans les données elles-mêmes : intra-colonne, inter-lignes et inter-colonnes.

2.2.2 Anomalies dans les données intra-colonne

Plusieurs anomalies, au sein d'une même colonne, dans les données ont été répertoriées dans la littérature [Berti-équille 2005] [Berti-Equille 2012]. Ramenées à leurs contextes, on peut les classer en deux grands types : les anomalies afférant à des données **syntaxiquement incorrectes** et celles afférant à des données **sémantiquement invalides**.
En effet, la chaîne de caractéres « Pari » ne peut être considérée incorrecte **syntaxiquement** que s'il s'agit du nom de la **ville** (**City**) « Paris ». La chaîne de caractéres « le bon@labas.fr » est syntaxiquement invalides si cette dernière répresente un mail car elle contient un espace. Les deux chaînes de caractères « 10-16-1996 » et « 1996-10-16 » représentent des dates mais dans deux formats différents. Les mots « Pékin » et « Beijing » désignent la même chose dans deux langues diffrentes. Le mot « Beijing » est qualifié incorrect sémantiquement si les chaînes de caractères doivent être en français.

Les données peuvent être alors **mal formatées**, **hétérogènes**, **codifiées de manières différentes**. D'autres types d'anomalies au sein d'une même colonne existent se sont les **valeurs manquantes (Null values)** [Pearson 2006], **les valeurs aberrantes (Outliers)** [Berti-Equille 2012] et les données obsolètes. Aucun référence bibliographique ne mentionne les anomalies intra-colonne dans la mesure ou la donnée est composée de deux ou plusieurs mots tels que « NOM Prénom » ou « Prénom NOM » ou encore « Code Postal Ville » ou « Ville Code Postal ».
Les données ci-dessous résument les anomalies les plus fréquentes dans les sources de données (anomalies intra-colonne).

31°C	M	2016-10-16	Parisss	hz@u.fr	Houda Dupont	75003 Paris
27 °C	F	16-10-2016	Paris	fb@u.fr	Yann Bon	75013 Paris
60°F	0	2016-10	Beijing	NULL	Adem Traifort	Villtaneuse 93430
NULL	1	30-02-2015	NULL	NULL	Adam Sympa	94 Orly
-10°C	Madam	-	Pari	le bon@u.fr	Sympa Adam	Paris 75008
20160°F	Homme	-	London	aujourd'hui@u.fr		
NULL	Femme	NULL	Londres	yp@u.fr		
-	NULL	-	Pékin	kb@u.fr	Dupont Houda	

TABLE 2.18 – Anomalies intra-colonne

La difficulté réside essentiellement dans le processus de détection automatique de ces anomalies afin de pouvoir proposer des actions correctives ou même de les réaliser automatiquement.
Peu de travaux ont abordé cette problématique. La sémantique de données a été plutôt négligée [BenSalem 2015]. Les valeurs manquantes ne sont pas toujours traitées, a fortiori

les données aberrantes et obsolètes.
Les problèmes sont encore plus compliqués si l'on considère que les sources de données sont rarement accompagnées de descriptifs complets de leurs contenus. Rappelons que dans notre travail, nous considérons que les sources sont au format CSV et donc sans schéma.

2.2.3 Anomalies dans les données inter-lignes

Les **redondances** inter-lignes regroupent les doublons et les similaires [Boufarès et al. 2012] [Boufarès et al. 2013] [Benjalloun et al. 2009]. La plus grande difficulté réside dans la reconnaissance des enregistrements qui sont « proches ». Les lignes 8 et 9, présentées dans l'exemple ci-dessous 2.19 issus de celui de l'introduction (Table 1.1), se ressemblent fortement. Il en est de même des deux dernières lignes.

Données source DS – DS.csv									
-	-	-	M	M	-	Parisss	France	-	-
-	-	-	Mme	F	-	Paris	Franc	-	-
-	-	-	M	F	1996-16-10	Loiret	France	Europe	-
-	-	Martin	-	M	03-avr-80	Paris	Fr	Europe	-
-	-	Anne	Mlle	1	12/03/1971	Beijing	Chine	Asie	-
-	-	Karine	Mlle	1	-	-	China	Afrique	-
-	-	Robert	M	0	-	Pari	Frence	Europe	-
-	-	Simon	M	0	-	Bruxelle	France	-	-
-	-	Simon	M	-	16-10-1996	Paris	-	Eurape	-
-	-	Simon	M	-	10-16-1996	Paris	-	Europe	-
-	-	Katia	Mlle	Femme	24/06/1983	Calvados	-	-	-
-	-	Houda	Mme	F	30/02/2000	Vill	Pai	Conti	-
-	-	Adem	M	0	-	Pékin	Chine	Asia	-
-	-	Adem	M	0	-	Beijing	Chin	Asia	-

TABLE 2.19 – Extrait tabulaire de la source de données DS

2.2.4 Anomalies dans les données inter-colonnes

Les données dans les colonnes peuvent être **dépendantes**. Plusieurs dépendances ont été mentionnées dans la littérature tels que les dépendances fonctionnelles, les dépendances fonctionnelles exactes, les dépendances fonctionnelles approximatives ou encore les dépendances fonctionnelles conditionnelles [Novelli and Cicchetti 2001] [Diallo and Novelli 2010] [Simonenko and Novelli 2012a] [Bohannon et al. 2007] [Liu et al. 2012].
La violation des dépendances fonctionnelles, est la cause de plusieurs anomalies dans les données. Ce type d'anomalies porte sur les liens sémantiques qui peuvent exister entre les

colonnes. Nous rappelons ci-dessous les contraintes de dépendances qui nous paraissent utiles dans le cadre de notre travail.

Soit $DS(C_1, C_2, ..., C_n)$ le schéma d'une source de données. Il est noté DS(C).
C est l'ensemble de toutes les colonnes. C= $(C_1, C_2, ..., C_n)$.
Une colonne C_i de DS peut être désignée par $DS.C_i$.
Soient $X \subseteq C$ et $Y \subseteq C$, deux sous ensembles de colonnes de C.
Soit t un tuple (une ligne, un enregistrement, un n-uplet) de DS. On note t[x] la valeur de la colonne (ou des colonnes) dans DS.

Définition 2.5 : dépendance fonctionnelle exacte

Une dépendance fonctionnelle (DF) exacte notée $(X \rightarrow Y)$ est valide dans une source de données DS dont le schéma est DS(C), pour une instance donnée, ssi pour toute paire de tuples t1 et t2 de DS : si t1[X]=t2[X] alors t1[Y]=t2[Y] [Huhtala et al., 1999]. •

Exemple 2.8 dépendance fonctionnelle exacte
Dans la mesure où les colonnes 8 et 9 sont dépendantes ($Col8 \rightarrow Col9$), les couples de valeurs (Paris, Franc) et (Paris, Fr) qui se trouvent dans les lignes 2 et 4 (Table 2.19) montrent que les données sont incohérentes et que donc la dépendance fonctionnelle exacte n'est pas vérifiée.

Définition 2.6 : dépendance fonctionnelle approximative

Une dépendance fonctionnelle entre X et Y est dite approximative suivant une mesure d'erreur, si la dépendance fonctionnelle $(X \rightarrow Y)$ est « presque exacte ». [Kivinen and Mannila, 1995] proposent trois mesures (g1, g2, g3) pour définir l'approximation d'une dépendance :

- La mesure g1 représente la proportion de paires qui ne vérifient pas la dépendance fonctionnelle exacte.
- La mesure g2 représente la proportion de tuples qui ne vérifient pas la dépendance fonctionnelle exacte (les tuples appartenant aux paires vérifient la condition de la mesure g1).
- La mesure g3 représente la proportion minimale des tuples à retirer pour que la DF exacte soit valide. •

Exemple 2.9 dépendance fonctionnelle approximative
$COL8 \rightarrow COL9$, g1= 0.35% :
{(France ;-),(France ;Europe),(France ;-),(Chine ;Asie),(Chine ;Asia)} (voir Table 2.19).

Définition 2.7 : dépendance fonctionnelle conditionnelle

Une dépendance fonctionnelle conditionnelle est une paire ($X \rightarrow Y$,Tp), Tp est un pattern tableau d'attributs de DS(C) [Bohannon et al. 2007]. •

Exemple 2.10 : dépendance fonctionnelle conditionnelle
($COL4 \rightarrow COL5$ (Mme||F)) (voir Table 2.19)

L'orginalité de notre travail consiste à traiter ce type d'anomalie inter-colonnes en mettant l'accent sur la sémantique des données. Ces redondances ont été peu ou pas du tout étudiées dans la littérature. Les traitements de la violation des dépendances fonctionnelles n'est pas non plus intégré dans les outils ETL. Seul Talend vérifie si une contrainte DF ($X \rightarrow Y$) exacte est respectée. Celle-ci doit être explicitée par l'utilisateur, ce qui suppose une connaissance sémantique des données et une expertise de ce dernier.

Les principales causes des anomalies dans les données celles se situant au niveau soit de la saisie manuelle des données, soit de certaines générations automatiques ou encore celles provenant de l'intégration et des transformations des données.
En effet, les formulaires de saisie en général et ceux du Web en particulier peuvent être mal conçus. Ainsi, on peut constater : (i) peu ou pas d'utilisation de référentiel afin d'éviter par exemples les erreurs typographiques ; (ii) la définition syntaxique des données est rarement accompagnée des types de données et à fortiori aucune contrainte sémantique n'est mentionnée ; (iii) la compatibilité sémantique entre les champs de saisie n'est pas étudiée. De nombreuses questions relatives à la qualité doivent être posées tout au long du cycle de vie de la donnée, telles que :

— la date de création de la donnée,
— les dates de mises à jour de la donnée,
— la compatibilité des types syntaxiques lors de l'intégration,
— la compatibilité des domaines sémantiques lors de l'intégration,
— les raisons des transformations des données lors de l'intégration.

Pour aider à résoudre ces problèmes, notre apport est de chercher à comprendre les données dans leur contexte. Nous basons notre réflexion sur le sens attribué à ces dernières.

Dès lors, la re-découverte de la sémantique des données à partir de la source manipulée a été notre objectif premier. Ce travail a été abordé dans [BenSalem 2015]. Les liens sémantiques éventuels inter-colonnes et d'autre part intra-colonne n'ont pas été pris en compte afin de traiter certaines valeurs nulles et mieux localiser les anomalies.

Les paragraphes suivants abordent les notions de traitement des anomalies à savoir leurs détections et leurs corrections. La compréhension des schémas de données s'est largement inspirée, au départ, de la notion d'alignement d'ontologie.

2.3 Reconnaissance sémantique des données

Le rapprochement de schémas ou l'alignement d'ontologies est l'un des problèmes majeurs rencontrés lors du processus d'intégration soit de données (par exemple les entrepôts de données), soit des applications (par exemple, le e-commerce, le Web sémantique). Tout ceci pose de grands problèmes aux utilisateurs qui cherchent à intégrer des informations provenant de sources différentes. Dans cette section, nous présentons un ensemble d'approches permettant d'aligner des schémas ou des ontologies, c'est-à-dire de trouver des similarités ou des correspondances entre des éléments de deux schémas ou ontologies qui sont sémantiquement liées afin d'être en mesure d'intégrer les données provenant d'une source dans une autre source.
Dans la littérature, il existe de nombreux travaux qui abordent le problème de l'automatisation de l'alignement de schémas, nous pouvons citer par exemple [P.Shvaiko and Euzenat. 2005], [Erhard 2011], [Erhard and Philip 2011], [Elbyed 2009] et [Saais 2007] [Diallo 2013] [Garnaud 2013],[Dallachiesay et al. 2013] [Garnaud et al. 2013] [Chu1and et al. 2015]. Plusieurs types des techniques d'alignement ont été présentés :

Des techniques basées sur des chaînes de caractères. Nous citons ci-dessous des exemples :

- Préfixe : cette méthode prend en entrée deux chaînes, qui correspondent aux noms des colonnes, et vérifie si la première chaîne est incluse au début de la deuxième chaîne (Par exemple : Pers et Person).
- Suffixe : cette méthode prend en entrée deux chaînes et vérifie si la première chaîne est incluse à la fin de la deuxième chaîne (Par exemple « phone » et « téléphone »).
- Edit distance : le nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre. (Par exemple d(Client, Clt) =3).
- N-grammes : est une sous-séquence de n éléments construite à partir d'une séquence donnée. En général, le n varie entre 1 et 5. (Par exemple les tri-grammes (3) pour

Adresse sont Adr, dre, res, ess, sse La distance entre Adresse et Adr est 0 (0/5).

Des Techniques linguistiques tells que les thésaurus (Par exemple WordNet) ou les thésaurus de domaine spécifique. Elles sont utilisées afin de faire correspondre des mots basés sur des relations linguistiques entre elles (par exemple synonymes, hyponymes, hyperonymes).

Des techniques qui s'appuient sur les différents types de structures de données notamment la structure en graphe et les taxonomies :

— Structure en graphe : les schémas de données en entrée du processus d'alignement (par exemple schémas de base de données, des taxonomies ou ontologies) sont considérés comme des graphes contenant des noeuds (sommets) et des relations (arcs). La comparaison de la similarité entre une paire de nuds de deux schémas de données (schémas ou ontologies) est généralement basée sur l'analyse de leur position dans le graphe.

— Les taxonomies : il s'agit d'une structure sous forme de graphe qui considère uniquement la relation de spécialisation. Les techniques taxonomiques se basent sur l'hypothèse que si des nuds possèdent des relations de type « is-a » alors ils sont souvent similaires et leurs voisins peuvent aussi être similaires [Limayeand et al. 2010] [Hamdoun and Boufarès 2010].

[Chu1and et al. 2015] proposent l'approche KATARA qui permet de reconnaitre certaines méta-données d'une table. Elle est basée sur des bases de connaissances (Yago [J. Hoffart and Weikumt 2013], DBPedia [Lehmann et al. 2015]). Cependant, cette approche ne permet pas de reconnaître la sémantique de certains types de données tels que les valeurs numériques (numéro de téléphone, la température) ou les valeurs qui doivent vérifier des expressions régulières. Il existe peut être un lien entre Madrid et Espagne ou encore entre Rome et Italie (VILLE et PAYS). Ce lien n'aurait pas de sens entre Madrid et Italie (l'une ou l'autre de deux valeurs pourrait être sémantiquement erronnée). Aucun lien sémantique n'existe entre adem@gmail.com et Madrid. Ce type de raisonnement pourrait être intérprété par un raisonnement ligne par ligne dans la source de données. Il serait judicieux de compléter par un raisonnement colonne.
Dans le domaine du web sémantique, des travaux sont en cours afin de reconnaître les sémantiques des colonnes dans une table et éventuellement découvrir des liens entre elles. Ces approches utilisent des bases de connaissances afin de faciliter la recherche des applications sur le web [Deng et al. 2013] [Venetis et al. 2011] [Limayeand et al. 2010].

2.4 Détection des anomalies dans les données

Le profilage des données, très utilisé dans le monde industriel, représente une collection de statistiques sur les données. Il s'agit d'une étape primordiale pour détecter les anomalies dans les données. Il s'agit d'une analyse exploratoire des données sur plusieurs niveaux à savoir au sein d'une même colonne, entre les lignes et entre les colonnes.
Il faut souligner que les outils ETL n'offrent aucune aide ni assistance. Les utilisateurs sont censés avoir une idée sur le contenu des sources manipulées tels que par exemple (i) le nombre de colonnes, (ii) la sémantique de chacune d'entre-elles et les liens entre-elles, (iii) la présence de valeurs manquantes, et (iv) les lignes en double : une tâche très difficile sinon impossible dans le cadre de gros volumes de données.

2.4.1 Détection des anomalies intra-colonne

Les outils de profilage des données, existants sur le marché, fournissent des résumés statistiques sur les données colonne par colonne. Ces analyses portent exclusivement sur les aspects syntaxiques des données, aucun indicateur n'existe pour déterminer la nature (la sémantique) de la colonne traitée. Elles ne permettent pas la détection des anomalies. Seuls nos travaux avec l'entreprise Talend, entamés depuis peu ([Ben-Salem et al., 2015], [BenSalem, 2015]) ont mis l'accent sur la qualité contextuelle des données. Des dictionnaires pé-établis aident à découvrir le sens des colonnes et par conséquent assistent la détection de certaines anomalies. Nous avons alors commencé la catégorisation des données.
Trois dictionnaires de données ont été présentés. Dans la première version des dictionnaires présentés, les liens sémantiques entre les catégories n'étaient pas pris en compte (les dépendances entre les colonnes n'étaient pas traitées). Ce référentiel contient : (i) la liste des chaînes de caractères supposées valides, (ii) un ensemble de mots clés qui aident à reconnaître le contenu d'une donnée, et (iii) une liste d'expressions régulières afin de vérifier la syntaxe d'un type de données (chaîne de caractères, numérique ou date) par rapport à sa catégorie.
La nouvelle version originale du dictionnaire de données sera détaillée dans le chapitre suivant.
L'apport sémantique constitue l'originalité de notre approche dans la détection des anomalies intra-colonne. En effet, nos dictionnaires de données apportent une aide quant à la découverte du sens de la colonne à partir de son contenu. Par exemple, la présence de deux caractères @@ dans une chaîne de caractères qui désigne un mail traduit une anomalie. Celle-ci n'est détectée que grâce à l'expression régulière, stockée dans le dictionnaire, qui définit un mail.

2.4.2 Détection des doublons et des similaires (anomalies inter-lignes)

L'élimination des doubles et des similaires, appelée dé-duplication des données (Record Linkage, Duplicates Data, Entity Resolution, Entity Matching), consiste à détecter les lignes doubles ou similaires afin d'éliminer les redondances dans une source de données [Hernandez and Stolfo 1998] [Sarawagi and Bhamidipaty 2002] [Koudas et al. 2006] [Benjalloun et al. 2009] [Boufarès et al. 2012] [F. Boufarès and S.Correia 2012] [Boufarès et al. 2013] [F. Boufarès and S.Correia 2012]. Les lignes similaires ou en double pourraient être fusionnées ou encore éliminées [Koudas et al. 2006]. Cette problématique continue à faire l'objet de plusieurs travaux de recherche, d'une part, afin d'améliorer les performances des algorithmes pour de gros volumes de données, et d'autre part, afin d'obtenir de meilleures précisions quant au calcul de similarité entre les lignes à cause des différentes anomalies qui existent dans les données et en particulier les valeurs nulles [Elmagarmid et al. 2007] .

En effet, la similarité entre les données est basée sur les différentes méthodes de mesures de distances de similarités qui existent dans la littérature telles que : (i) les mesures lexicographiques (Levenshtein, Jaro-Winkler, Q-Gram), ou (ii) les mesures phonétiques (Soundex, Double Metaphone) [Levenshtein 2007] [Ukkonen 1992] [Winkler 2006] [Winkler 2000]. Nous signalons que toutes ces méthodes ne prennent pas en considération la sémantique contextuelle des données à comparer.
Le choix des attributs clés pour la comparaison constitue aussi un grand problème. Le résultat de l'élimination des doubles est fortement dépendant des étapes de nettoyage et de profilage des données qui précèdent la dé-duplication effective [Ben-Salem et al. 2015]. L'ordre des priorités des actions à mener influence grandement la qualité du résultat. L'homogénéisation, le traitement des valeurs nulles, les liens inter-colonnes et les liens inter-lignes sont des étapes dépendantes.
Différentes approches déterministes existent pour la comparaison des lignes dans une source de données. Elles sont basées sur deux fonctions : Match (pour la comparaison) et Merge (pour la fusion). Là encore, peu ou pas de sémantique est prise en compte dans le processus de dédoublonnage. L'originalité de notre approche, en faisant appel à la catégorisation sémantique, consiste à recommander des colonnes à prendre en considération pour l'élimination des doubles ce qui est fondamental pour la validité du résultat.

Afin de décider de la similarité entre valeurs et ensuite entre tuples, nous avons défini trois similarités pour la fonction Match : similarité entre valeurs, règles de similarité et similarité entre tuples.

Définition 2.8 : similarité entre valeurs (notée ≈)

Deux valeurs v et v' sont similaires, on note ($v \approx v'$), ssi la distance de similarité d, calculée entre ces deux valeurs, vérifie une condition k. Pour deux tuples t1 et t2, pour une colonne Ci avec (i=1 ;n avec n est le nombre de colonnes), $t1.C_i \approx t2.C_i$ ssi la condition k_j est vérifiée (j=1 ;f avec f est le nombre de conditions). La condition k_j se base sur le calcul de la distance d_i de similarité selon le type des données. Plusieurs cas de figures peuvent être envisagés. L'utilisateur peut ainsi fixer (donner) un ou plusieurs seuils.
k_j : d_i est inférieur à un seuil maximal ; $(d_i < \epsilon_i)$ avec $\epsilon_i \in [0,1]$ et $d_i \in [0,1]$.
k_j : d_i appartient à un ensemble de seuils ; $(d_i \in \{\epsilon_i, ..., \epsilon_p\})$;
k_j : d_i appartient à un intervalle de seuils ; $(d_i \in [\epsilon_1..\epsilon_2])$

La similarité (≈) vérifie évidemment les propriétés de réflexivité, commutativité et d'associativité. Cest-à-dire [v≈v'], [(v≈v') ⇔ (v'≈v)] et [v≈(v'≈v") ⇔ (v≈v')≈v"]. •

Exemple 2.11 : similarité entre valeurs
Soit :
$Address1 \leftarrow t1.Address$; $Address2 \leftarrow t2.Address$;
$Email1 \leftarrow t.Email$; $Email2 \leftarrow t2.Email$;
$Name1 \leftarrow t1.Name$; $Name2 \leftarrow t2.Name$;
$Phone1 \leftarrow t1.Phone$; $Phone2 \leftarrow t2.Phone$

Deux adresses sont similaires (Address1 ≈ Address2) si la distance d de similarité, selon une mesure choisie, est inférieure à un seuil donné.
Par exemple, les distances de similarité des deux adresses suivantes
Address1 = « 133 BOULEVARD FOCH EPINAY/SEINE » et
Address2= « 133 BOULEVARD FOCH 93800 EPINAY-SUR-SEINE »
selon les méthodes Edit-distance, Jaro-Winkler et Soundex sont données avec les scripts SQL ci-dessous.

```
SELECT
UTL_MATCH.edit_distance_similarity(
'133 BOULVARD FOCH EPINAYSEINE',
'133 BOULVARD FOCH 93800 EPINAY-SUR-SEINE') AS EDS,
UTL_MATCH.jaro_winkler_similarity(
'133 BOULVARD FOCH EPINAYSEINE',
'133 BOULVARD FOCH 93800 EPINAY-SUR-SEINE') AS JWS,
UTL_MATCH.edit_distance_similarity(
soundex('133 BOULVARD FOCH EPINAYSEINE'),
soundex('133 BOULVARD FOCH 93800 EPINAY-SUR-SEINE')) AS SDX
FROM dual;
```

Les résultats obtenus, sous oracle, sont : EDS : 73 JWS : 92 SDX : 100

```
SELECT
UTL_MATCH.edit_distance(
'133BOULVARDFOCHEPINAYSEINE',
'133BOULVARDFOCH93800EPINAY-SUR-SEINE')ASED,
UTL_MATCH.jaro_winkler(
'133BOULVARDFOCHEPINAY/SEINE',
'133BOULVARDFOCH93800EPINAY-SUR-SEINE')ASJW,
UTL_MATCH.edit_distance(
soundex('133BOULVARDFOCHEPINAY/SEINE'),
soundex('133BOULVARDFOCH93800EPINAY-SUR-SEINE'))ASSDX
FROMdual;
```

Les résultats obtenus, sous oracle, sont : ED : 11 JW : 9,245E-001 SDX : 0

Cet exemple montre bien la difficulté, d'une part, de choisir la méthode de calcul de distance de similarité, et d'autre part, le seuil à partir du quel on considère que les deux valeurs comparées sont similaires ou égales.

Définition 2.9 : règle de similarité

Une règle r de similarité est une conjonction de similarités qui portent sur des colonnes C_i avec (i=1 ;n) de l'ensemble C de la source (formule 1). j=1 ;g avec g est le nombre de règles de similarité.

$r_j = \mathrm{d}_1 \leq \epsilon_1 \wedge \mathrm{d}_2 \leq \epsilon_2 ... \wedge \mathrm{d}_n \leq \epsilon_n (1)$

avec d_a est la distance de similarité entre deux colonnes clés et ϵ_α un seuil pour chaque paire de colonnes.
$r = (t1.C_1 \approx t2.C_1) \wedge (t1.C_2 \approx t2.C_2) \wedge (t1.C_i \approx t2.C_i) ... \wedge (t1.C_n \approx t2.C_n)$. •

Exemple 2.12 : Règles de similarité
Voici un exemple de deux règles de similarités r1 et r2 :
$r1 = (Name1 \approx Name2) \wedge (Email1 \approx Email2) \wedge (Address1 \approx Address2)$,
$r2 = (Email1 \approx Email2) \wedge (Phone1 \approx Phone2)$.

Définition 2.10 : similarité entre tuples
Deux tuples t1 et t2 sont similaires ssi la disjonction des règles de similarité définies sur la source S est vraie. On note $t1 \approx t2$ ssi $r1 \vee r2 \vee ... \vee rk ... \vee$ rq est vraie avec q étant le nombre de règles de similarité (formule 2).

$$\text{Rule} = \vee_{x=1}^{q} r_x \qquad (2) \bullet$$

Exemple 2.13 : similarité entre tuples
t1 et t2 sont similaires ssi $r_1 \vee r_2$:

$$t1 \approx t2$$

ssi ((Name1 $\approx Name2$) $\vee$ ($Email1 \approx Email2$) $\vee$ ($Address1 \approx Address2$)) $\vee$ (($Email1 \approx Email2$) $\vee$ ($Phone1 \approx Phone2$))

Remarque :
Ces règles peuvent présenter toutefois des incohérences ou des contradictions qui ne sont pas étudiées dans ce manuscrit.
Le choix des seuils dans les travaux précédents nécessite des connaissances métiers [Hernandez and Stolfo, 1998]. Notre apport consiste à recommander les meilleurs seuils en fonction de la catégorie sémantique des données.

2.4.3 Détection des contraintes de dépendances (inter-colonnes)

Novelli et al. ont traité le problème de la découverte des dépendances fonctionnelles exactes dans [Novelli and Cicchetti, 2001]. Ils ont proposé un algorithme basé sur les notions de fermeture et quasi-Fermeture. L'algorithme FUN parcourt la table en commençant par les sous-ensembles de taille1 et il construit le niveau supérieur tout en se basant sur les informations du niveau en cours. Il permet de calculer la fermeture et la quasi-fermeture de chaque niveau.

Définition 2.11 : fermeture d'un ensemble d'attributs dans une source DS

Soit $X \subset DS(C)$ un ensemble d'attributs dans DS. Sa fermeture est définie comme suit :
X^{+}_{DS} = X $\cup \{A \subset DS(C) - X / |\mathrm{X}|_{DS} = |X \subset A|_{DS}\}$•

Définition 2.12 : quasi-fermeture d'un ensemble d'attributs dans une source

La quasi-fermeture d'un ensemble d'attributs notée X° est définie par :
X°_{DS} = X $\cup \bigcup_{A \in X} (X - A)^{+}_{DS}$. •

Niveau 1			
X	**|X|**	X^{+}	X°
C2	6	C2	C2
C3	11	C3	C3
C4	3	C4	C4
C5	5	C5	C5
C6	7	C6	C6
C7	9	C7	C7
C8	8	C8	C8
C9	6	C9	C9

TABLE 2.20 – FUN (Niveau 1) : exemple d'une source de données

Niveau 2			
X	**|X|**	X^{+}	X°
C2C3	13	C2C3	C2C3
C2C4	12	C2C4	C2C4
C2C5	12	C2C5	C2C5
C2C6	14	C2C6	C2C6
C2C7	13	C2C7	C2C7
C2C8	14	C2C8	C2C8
C2C9	14	C2C9	C2C9
C3C4	...		
C3C5	...		

TABLE 2.21 – FUN (Niveau 2)

En appliquant l'algorithme FUN sur l'exemple (2.18). Le résultat montre que l'algorithme FUN ne permet pas de détecter les dépendances fonctionnelles telles que :
- $C5 \rightarrow C4$ ($Civilit \rightarrow Gender$),
- $C7 \rightarrow C8$ ($Ville \rightarrow Pays$), et
- $C8 \rightarrow C9$ ($Pays \rightarrow Continent$),

car le fichier Patient.csv contient des données rassemblées de plusieurs sources hétérogènes. Il contient des anomalies (Fr, Eurape, Parisss, Pari) et des données de différents formats par exemple 0, 1, M, F, Femme.
Il est donc nécessaire donc de standardiser les données et corriger les anomalies pour que l'algorithme FUN permette de déterminer toutes les dépendances fonctionnelles valides d'une source de données.
En plus, le principe de l'algorithme FUN est d'utiliser la connaissance extraite d'un niveau pour accélérer les calculs du niveau suivant. Il s'agit d'une occupation mémoire non négligable pour traiter de grandes quantités de données car il faut stocker deux niveaux simultanément.

Diallo et al dans [Diallo and Novelli, 2010] ont montré que la technique utilisée pour la découverte de dépendances fonctionnelles exactes peut être étendue aux dépendances fonctionnelles conditionnelles. Ils ont proposé l'algorithme CFun basé sur l'approche FUN. Cet algorithme permet la découverte des dépendances fonctionnelles conditionnelles valides à partir d'une source de données. En appliquant l'algorithme CFun sur notre exemple, le résultat montre que l'algorithme CFun ne permet pas de détecter les dépendances fonctionnelles citées ci-dessus. Il détecte des dépendances fonctionnelles non valides telles que : $C7Bruxelle \rightarrow C8France$.
L'algorithme CFun ne permet pas la découverte de l'ensemble de toutes les dépendances fonctionnelles valides.

Simonenko et al ont proposé dans [Simonenko and Novelli, 2012b] l'algorithme AFD-DYNAMICUPDATE pour la découverte de dépendances fonctionnelles. C'est une approche incrémentale qui maintient à jour l'ensemble des dépendances fonctionnelles valides, exactes ou approximatives selon une erreur donnée, quand des tuples sont insérés et supprimés. Cette erreur représente un ensemble minimal de tuples à retirer de la source de données pour que la dépendance fonctionnelle exacte ($X \rightarrow Y$) soit valide.
En appliquant cet algorithme sur l'exemple (Patien.csv, Table1.1), la dépendance fonctionnelle $Ville \rightarrow Pays$, n'est valide que si l'on retire 6 lignes qui représentent 42% de l'ensemble des tuples.

2.5 Correction des anomalies dans les données

Afin de remédier à certaines anomalies, différents travaux ont été proposés dans la littérature. Ils ont porté essentiellement sur les actions de nettoyage au sein d'une même colonne (homogénéisation) et entre les lignes (dé-duplication).
Peu de travaux ont abordé vraiment la vérification des dépendances qui peuvent exister entre les colonnes et surtout la correction des anomalies dues à la violation de ce type de contraintes.

2.5.1 Correction des anomalies intra-colonne

L'homogénéisation des valeurs dans une même colonne consiste à assurer la conformité de la donnée par rapport aux standards utilisés. La codification et le format utilisés devront être clairs et commentés. Par exemple, (i) le sexe d'une personne devrait être 'M' ou 'F' et non '1' ou '0' ni 'Homme' ou 'Femme' ; (ii) la température doit être représentée en °C et non en F° ; (iii) ; le numéro de téléphone devrait contenir l'indicatif du pays ; (iv) l'adresse doit respecter les normes de la poste. Des conversions de types de données sont parfois nécessaires pour homogénéiser les colonnes (tels que le passage du type String vers les autres types ou du type date vers le type string ou encore du type numérique vers le type string). Nous pensons qu'une donnée du type Date devrait avoir le format « année(4)-mois(2)-jour(2) ». celui-ci permet de réaliser le plus grand nombre possible d'opérations sans ambiguïté telles que les tris implicites ou l'extraction des années ou des mois.

Les outils ETL permettent de faire plusieurs type d'unifications du contenu d'une colonne. Le problème réside dans la détection automatique de l'hétérogénéité. L'utilisateur n'est donc pas assisté dans le processus de nettoyage des colonnes dans une source donnée. La question est « Qui décide de l'homogénéisation du contenu d'une colonne, laquelle et à quel moment ? »

L'approche sémantique abordée dans ([Ben-Salem et al. 2015]) permet d'assurer un meilleur suivi des modifications.

2.5.2 Correction des anomalies inter-lignes

Différents algorithmes de dé-duplication ont été présentés dans la littérature [Cohen 2002]. Nous présentons brièvement ceux implémentés récemment dans l'outil Talend. Il s'agit des algorithmes MFBi (i=1,4) ([Boufarès et al. 2012], [Boufarès et al. 2013]) dont

les performances étaient très satisfaisantes tant au niveau de la précision qu'au niveau du temps de réponse. Ces algorithmes sont basés sur les fonctions Match et Merge présentées ci-dessus. La fonction Match compare les lignes de la source deux à deux. Elle est basée sur les méthodes de calcul de distances de similarités (lexicographiques combinées aux phonétiques) Elle bénéficie largement de la sémantique de la colonne quant au choix des attributs de dé-duplication. Des règles de similarités sont explicitées dans les comparaisons. La fonction Merge, qui est à améliorer, se base sur des règles simples telles que : (i) privilégier les valeurs non nulles ; (ii) la valeur la plus récente est gardée pour le type date ; (iii) la moyenne des deux valeurs est retenue (ou encore la valeur minimale ou maximale selon le choix de l'utilisateur) ; (iv) pour les chaînes de caractères le choix est plus large à savoir la concaténation des deux ou la plus longue ou la plus fréquente des deux.

Les algorithmes appliquent le principe de tri-fusion. La complexité de l'algorithme MFB3 est de l'ordre de (n Log n). L'algorithme MFB4, en cours de tests, est basé sur la technologie MapReduce. Les premiers résultats sont prometteurs en terme de temps de réponse.

Les étapes de nettoyage étant dépendantes, nous pensons aujourd'hui que le traitement des dépendances fonctionnelles entre les colonnes devrait précéder l'étape d'élimination des doubles. En effet, la présence des valeurs nulles influence la précision des résultats du dédoublonnage. Il faut noter que certaines valeurs nulles peuvent être substituées par la valeur exacte en traitant les dépendances fonctionnelles.

2.5.3 Correction des anomalies inter-colonnes

Peu de travaux dans la littérature ont traité le problème de la violation de contraintes de dépendances fonctionnelles. Ce dernier peut engender plusieurs anomalies inter-colonnes. On peut citer par exemple l'approche proposée par [Bohannon et al. 2007]. L'idée consiste à extraire toutes les dépendances vérifiée sur une table de référence correcte. Ensuite, il faut vérifier leur validité sur les tables potentiellement impropres pour les corriger.
Une première solution proposée pour corriger les valeurs incohérentes est de supprimer les tuples qui ne vérifient pas les dépendances fonctionnelles mais il est facile de constater que ce n'est pas satisfaisant d'éliminer une ligne sous prétexte qu'une colonne est mal renseignée.
[Chiang and Millert 2008] proposent de nettoyer la table en recherchant les dépendances conditionnelles qui sont entièrement vérifiées pour retourner à un expert les lignes incorrectes. Ces dernières sont soumises à vérification.

[Chu1and et al. 2015] proposent une méthode pour corriger les valeurs incohérentes. Il s'agit de chercher pour chaque ligne d'une source de données une ligne équivalente dans la base de connaissance. Si elle n'existe pas il faut proposer un ensemble de corrections possibles. Ensuite, cette méthode permet de classifier ces corrections selon leurs coûts (nombre de valeurs à remplacer dans la ligne à corriger).

2.6 Outils de gestion de la qualité des données

Nous avons étudié les fonctionnalités des outils de gestion de la qualité des données et des outils ETL (voir tableau 2.4). Nous avons comparé 5 outils à savoir Talend Data Quality [Ref a] (A), Pentaho Data Integration [Ref b] (B), Informatica [Ref c] (C), NADEEF [Dallachiesay et al. 2013] (D) et KATARA [Chu1and et al. 2015] (E).

Numéro	**Nom**
A	Talend Data Quality
B	Pentaho Data Integration
C	Informatica
D	NADEEF
E	KATARA

TABLE 2.22 – FUN (Niveau 2)

La comparaison porte sur plusieurs critères. Ces derniers représentent des fonctionnalités liées à la qualité des données telles que :

— Les analyses statistiques sur les données : les fonctions qui fournissent des statistiques simples, par exemple, le nombre de lignes, le nombre de valeurs nulles, le nombre de valeurs distinctes et uniques. Des statistiques sur des données de type texte, de type numérique et de type date permettent d'analyser les caractéristiques des colonnes, telles que les longueurs minimale, maximale et moyenne.

— Les transformations nécessaires sur les valeurs des données lors de l'intégration : regroupe les fonctions de transformation des dates et des nombres.

— Des doublons et des similaires (doublons non strictes).

Le tableau 2.23 permet de comparer les différents outils étudiés. Il met l'accent sur les fonctionnalités qui n'existent pas, c'est ce qui justifie l'importance et l'originalité de notre approche. La légende utilisée est : (X) pour les fonctionnalités couvertes et (-) pour les

fonctionnalités non couvertes.

Ces outils offrent la possibilité de faire des statistiques et des transformations des données. Ils permettent l'élimination des doubles. Mais l'utilisateur doit avoir des connaissances sur les structures et la sémantique des données pour arriver à corriger les anomalies et les valeurs incohérentes. Ces actions correctives doivent être déclenchées par l'utilisateur. Aucune aide ne lui est apportée.
Pentaho Data Integration et data cleaner ne permettent pas de vérifier les dépendances fonctionnelles. TalendDataQuality permet de vérifier les dépendances fonctionnelles, l'utilisateur doit avoir des connaissances sur le schéma de données et sur les dépendances à vérifier. Les deux outils ne corrigent pas les erreurs causées par la violation de dépendances fonctionnelles. Après cette étude sur ces ETL open source. Nous avons constaté les points faibles à améliorer et les fonctionnalités à développer pour contribuer au développement de nouveaux outils qui n'imposent pas à l'utilisateur la connaissance des structures et des sémantiques des données manipulées en provenance des sources et permettent d'assister la correction des anomalies causées par la violation de dépendances fonctionnelles.

Fonctionnalités	**A**	**B**	**C**	**D**	**E**
Mesures globales					
Traitement de fichier CSV (avec schéma)	X	X	X	X	-
Traitement de fichier CSV (sans schéma)(intervention utlisateur)	X	X	X	X	-
Nombre total de valeurs nulles dans DS	X	X	X	-	
Nombre total de valeurs syntaxiquement invalides dans DS	X	-	-	-	-
Nombre total de lignes en double dans DS (tous les attributs)	X	-	-	-	-
Nombre total de lignes en double dans DS (attributs clés)	X	-	-	-	-
Nombre total de lignes similaires dans DS (attributs clés)	X	-	-	-	-
Nombre de colonnes en double dans DS	X	-	-	-	-
Assistance	-	-	-	-	-
Recommandation	-	-	-	-	-
Mesures intra-colonne					
Nombre de valeurs syntaxiquement invalides dans la colonne	X	-	-	-	-
Nombre de valeurs distinctes parmi les valeurs syntaxiquement invalides	-	-	-	-	-
Le nombre de "mots" à considérer au sein d'une même colonne	-	-	-	-	-
Le nombre de valeurs par catégorie sémantique	X	-	-	-	-
Le nombre de valeurs par sous-catégorie sémantique	X	-	-	-	-
Mesures inter-colonnes					
Relation sémantique (=) entre deux colonnes	X	-	-	-	-
Relation sémantique (<) entre deux colonnes	-	-	-	-	-
Relation sémantique (>) entre deux colonnes	-	-	-	-	-
Vérification de la DF (intervention utlisateur)	X	-	-	X	-
Nettoyage de données					
Transformations sur les valeurs des données (manuel)	X	X	X	X	-
Transformations sur les valeurs des données (automatique)	-	-	-	-	-
Définir le schéma cible (manuel)	X	X	X	X	X
Reconnaissance de schéma (automatique)	X	-	-	X	-
Restructuration de la source (automatique)	-	-	-	-	-
Dédoublonnage (manuel)	X	X	X	X	-
Dédoublonnage (automatique)	-	-	-	-	-
Découverte de la DF (manuel)	X	-	-	-	X
Découverte de la DF (automatique)	-	-	-	-	X
Correction des anomalies causées par la violation de DF	-	-	-	-	X

TABLE 2.23 – Tableau comparatif des outils de gestion de la qualité des données et des outils ETL

2.7 Bilan

Les techniques de rapprochement de schémas présentées dans le pragraphe précédent sont basées sur la comparaison lexicographique des noms de colonnes. Ces techniques ne permettent pas de trouver la correspondance entre les colonnes s'il s'agit des noms insignifiants tels que Cl ou Co. Ce rapprochement ne prend pas en considération la langue de la colonne, il ne permet pas de déduire l'équivalence entre les noms de colonnes qui sont synonymes dans des langues différentes.

L'orginalité de notre approche est que nous utilisons les données pour construire un schéma sémantique. Ce schéma contient le domaine sémantique (la catégorie et la sous-catégorie telle que la langue) et syntaxique (le type de données) de chaque colonne d'une manière explicite. Ce schéma sémantique facilite la détection des colonnes similiaires et l'intégration de données dans un ETL.
Dans ce chapitre, nous avons étudié différents algorithmes de découverte de dépendances fonctionnelle. La détection des contraintes de dépendances est une étape préalable pour corriger les anomalies causées par la violation de ces contraintes. Ces algorithmes consistent à chercher les dépendances à travers toutes les combinaisons possibles pour les différentes colonnes d'une source de données. Ceci augmente la taille de l'espace de recherche. Ces algorithmes ne prennent pas en considération la sémantique des colonnes.
La nouveauté de notre approche est de guider l'utilisateur dans le processus de vérification des contraintes de dépendance en lui suggérant toutes les dépendances signifiantes tout en se basant sur la sémantique des colonnes.
Les outils de la qualité de données et les outils ETL existants sont trop manuels et ils imposent à l'utilisateur la connaissance des structures et des sémantiques des données manipulées en provenance des sources.
Notre objectif dans ce travail est de développer un nouvel outil qui n'exige pas la connaissance des structures et des sémantiques des données manipulées en provenance des sources. Cet outil permet de reconnaitre automatiquement les métadonnées d'une source de données (Catégorie, sous catégorie, type de données). Il permet aussi d'assister la correction

des anomalies dans une colonne et les anomalies causées par la violation des contrtaintes de dépendance.

2.8 Conclusion

Nous avons présenté dans ce chapitre les différentes anomalies dans les schémas des données et dans les données elles-mêmes. Les travaux existants dans la littérature sur les traitements des anomalies ne permettent pas de résoudre les incohérences entre les colonnes ni même les anomalies sémantiques dans une même colonne. Cette dernière peut, elle aussi, reflèter la dépendance sémantique entre les données qui a toujours été négligée. Nous nous sommes concentrés sur les travaux liés au rapprochement de schémas d'une part et ceux liés aux découvertes de dépendances fonctionnelles d'une autre part. À la fin de ce chapitre nous avons présenté le résultat de notre étude sur les outils ETL.
Nous avons déduit que nous avons besoin de sémantique sur les données pour guider l'utilisateur dans les processus de l'intégration, de détection des anomalies et de nettoyage des données.

Chapitre 3

Catégorisation sémantique des données et liens inter-colonnes

Sommaire

3.1 Introduction

L'étape de catégorisation sémantique des données consiste à déterminer le sens de chaque colonne d'une source de données à savoir :

- Reconnaître le nom sémantique d'une colonne : en se basant sur son contenu, on peut par exemple découvrir s'il s'agit de chaines de caractères qui désignent des noms de Villes (**City**) ou des noms de personnes (**Firstname**), ou encore des **dates** au **format jj-mm-aaaa**.
- Attribuer un **type** à une colonne (le type syntaxique) : La découverte du type des données (**String**, **Number**, **Date**) dans la colonne se basant sur son contenu.
- Déterminer les **contraintes syntaxiques** et **sémantiques** entre les colonnes.
- Enrichir la description de chaque colonne par d'éventuels commentaires tels que la langue utlisée ou des contraintes d'intégrité.
- Restructurer une colonne : éventuellement créer plusieurs colonnes à partir d'une seule.
- Déduire les liens sémantiques que peut avoir la colonne en question avec d'autres colonnes.

Le but final est de découvrir la structure de données la plus complète possible de la source. Cette dernière est composée de données massives et éventuellement non structurées. La description détaillée des données devra permettre de guider la détection et la correction des anomalies.

En effet, les sources de données étant considérées sans schéma (ou encore avec un schéma insignifiant, incomplet et ambiguë), l'étape de catégorisarion des données (appelée aussi l'étape de profilage des données) a pour objectif de permettre une meilleure compréhension de la définition des données afin d'améliorer la détection et la correction des anomalies. Cette étape prend en entrée une source de données sans schéma (voir table 4.16, table 3.2) ainsi qu'un ensemble de dictionnaires de données. Elle renvoie des rapports d'anomalies sur les données et propose un nouveau schéma sémantique (voir figure 3.11, figure 3.13).

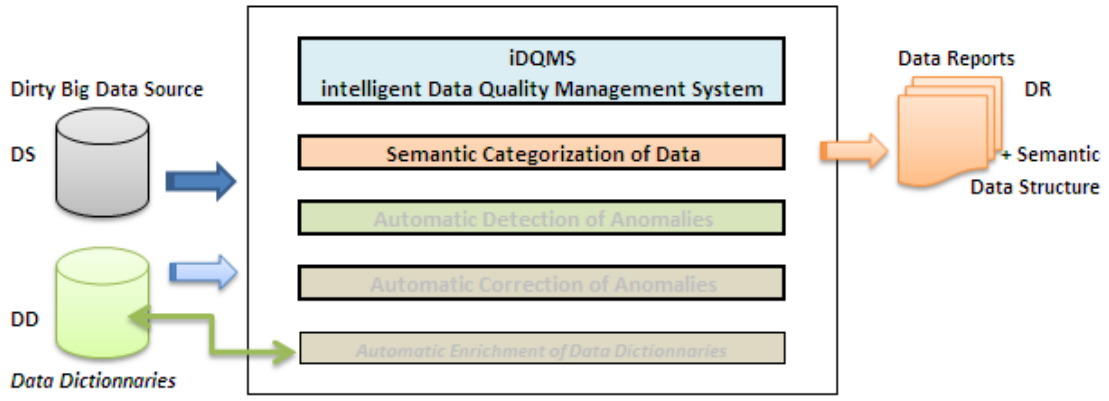


FIGURE 3.1 – L'étape de catégorisation des données

La catégorisation sémantique consiste donc à attribuer une catégorie et éventuellement une sous-catégorie à partir d'un ensemble prédéfini de catégories [Zaidi et al. 2015] [Zaidi et al. 2016b]. L'orginalité de notre approche réside dans le fait d'établir des liens sémantiques entre les colonnes qui correspondent aux catégories. Cette étape permet de mieux comprendre les données et recommander des actions de corrections ou de mises à jour. Notre outil iDQMS propose des actions correctives intra et inter-colonnes

L'utilisateur de ce type d'outil est ainsi assisté pendant les actions de traitement des anomalies. Le processus de détection des anomalies va se baser sur un ensemble de dictionnaires de données dont les méta-modèles sont présentés ci-dessous. Un ensemble de mesures et de ratios sera calculé afin d'aider à la prise de décision à propos des actions correctives. Dans ce chapitre nous détaillons l'étape préalable de la catégorisation. Cette dernière consiste à rapprocher les données d'une source de données au dictionnaire DDVS pour déduire une première proposition de la catégorisation sémantique. Il s'agit de reconnaitre la catégorie, la sous-catégorie et les contraintes pour chaque colonnes, et les liens sémantiques qui peuvent exister entre les colonnes.

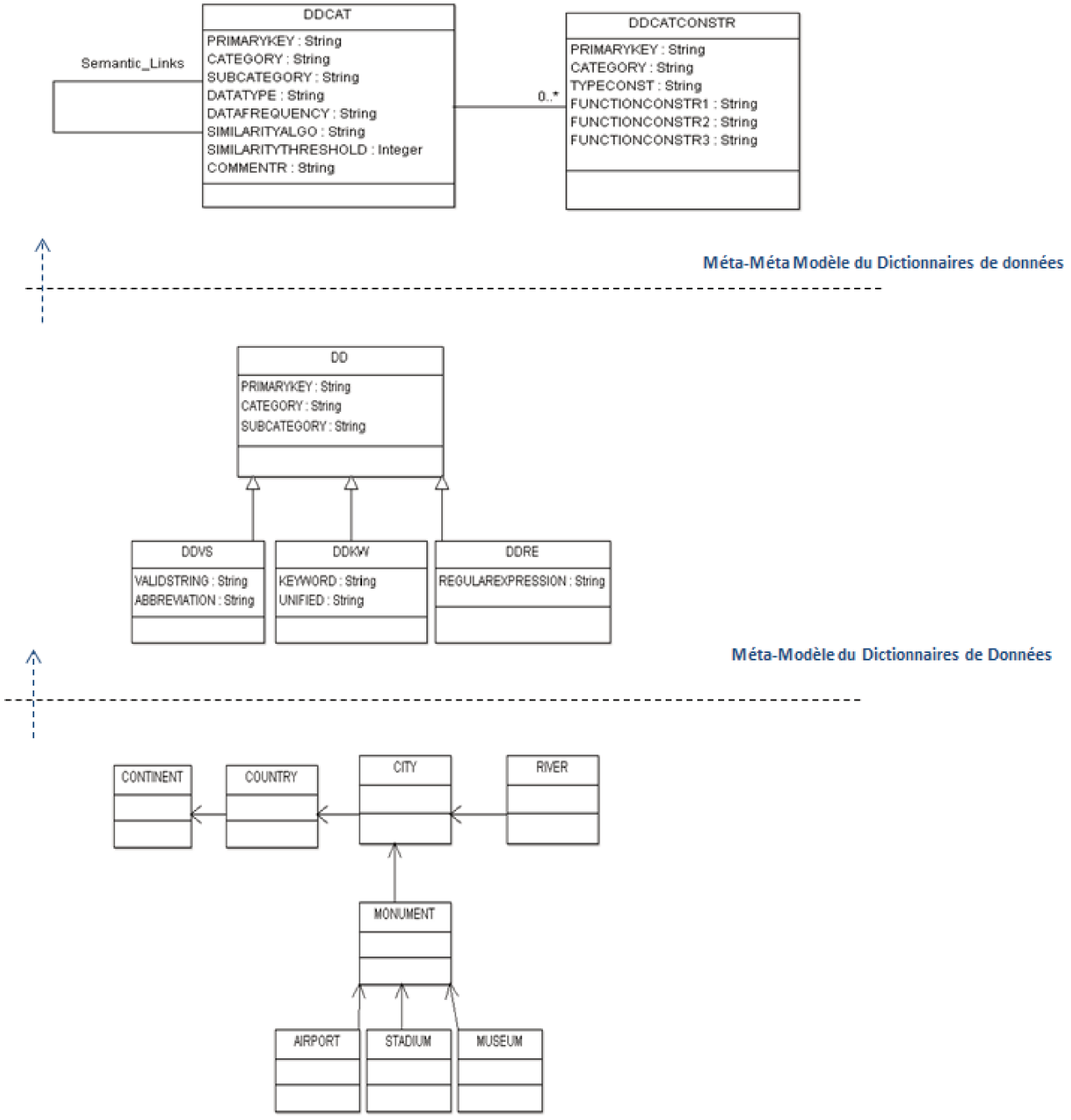


FIGURE 3.2 – Méta-modèles des dictionnaires de données

Avant d'entamer les étapes du processus de la catégorisation sémantique des données et des liens entre elles, nous annonçons quelques notions qui seront utilisées tout au long

de ce chapitre.

Définition 3.1 : Source de Données (DS)

Soit $DS(C_1, C_2, ..., C_n)$ est une source de données.

C est l'ensemble des colonnes, n est le nombre de colonnes et N est le nombre de lignes.

DS peut être vu comme un ensemble de valeurs v_{ij} qui peuvent être regroupées en colonnes selon leurs sens.

$DS = \{v_{ij}; i = 1, N; j = 1, n\}$

Chaque colonne C_j contient l'ensemble des valeurs $\{v_{1j}, v_{2j}, ...v_{Nj}\}$, tel que $v_{ij} \in Ligne_i$ *et* $Colonne_j$, i=1 ;N, j=1 ;n.

La figure ci-dessous donne un exemple de source de données. Chaque colonne C_j devrait contenir des valeurs homogènes qui appartiennent au domaine de définition de la colonne (le domaine sémantique). Une instance d'une source de données est alors un sous-ensemble de produit cartésien des domaines de chacune des colonnes (voir figure 3.3). •

Data Source : DS					
C1	C2	...	Cj	...	Cn
v11	v12	...	v1j	...	v1n
v21	v22	...	v2j	...	v2n
...		...		...	...
vi1	vi2	...	vij	...	vin
...			...		
...			...		
vN1	vN2		vNj		vNn

FIGURE 3.3 – Une instance d'une source de données (DS)

Exemple 3.1 : source de données sans schéma

L'exemple ci-dessous (Table 4.16 et Table 3.2) constitue un extrait de la source de données (DS) au format CSV, sans schéma, donné en introduction (Table 1.1), nous adoptons la

reprÃ©sentation tabulaire de cette source.

175099943272264	-	-	M.	M	-	-	-	-	-
180089987976564	CRI	-	Mme	F	-	-	-	-	-
165037895642322	AGRR	-	M.	F	-	-	-	-	-
180046378965464	CRP	M. Martin DUPONT	-	M	-	-	-	-	-
171038976542322	MGEN	Mlle Anne MARTIN	Mlle	1	-	-	-	-	-
278025125874563	-	Mlle Karine LEBON	Mlle	1	-	-	-	-	-
157054725912564	OTC	M. Robert FORT	M.	0	-	-	-	-	-
177125915879625	IPECA	M. Simon GENEREUX	M.	0	-	-	-	-	-
174046784763822	IPECA	M. Simon GENEREUX	M.	-	-	-	-	-	-
174046784763822	IPECA	M. Simon GENEREUX	M.	-	-	-	-	-	-
283068794585464	IPECA	Mlle Katia BON	Mademoisele	Femme	-	-	-	-	-
275478784581464	-	Mlle Houda ZAIDI	Mlle	F	-	-	-	-	-
285099935116964	-	M. Adem LE BON	M.	0	-	-	-	-	-
285099935116964	-	M. Adem LE BON	M.	0	-	-	-	-	-
285099935115564	-	M. Robert LEBON	M.	0	-	-	-	-	-
285099935115522	-	M. Robert DUPONT	M.	0	-	-	-	-	-

TABLE 3.1 – Représentation tabulaire de la source de données DS

175099943272264	-	-	-	-	-	Parisss	France	-	-
180089987976564	-	-	-	-	-	Paris	Franc	-	
165037895642322	-	-	-	-	1996-16-10	Loiret	France	Europe	10/12/2014
180046378965464	-	-	-	-	03-avr-80	Paris	Fr	Europe	
171038976542322	-	-	-	-	12/03/1971	Beijing	Chine	Asie	
278025125874563	-	-	-	-	-	-	China	Afrique	
157054725912564	-	-	-	-	-	Pari	Frence	Europe	
177125915879625	-	-	-	-	-	Bruxelle	France	-	
174046784763822	-	-	-	-	16-10-1996	Paris	-	Eurape	01/02/2000
174046784763822	-	-	-	-	10-16-1996	Paris	-	Europe	23/11/2015
283068794585464	-	-	-	-	24/06/1983	Calvados	-	-	
275478784581464	-	-	-	-	30/02/2000	Vill	Pai	Conti	
285099935116964	-	-	-	-	-	Pékin	Chine	Asia	
285099935116964	-	-	-	-	-	Beijing	Chin	Asia	
285099935115564	-	-	-	-	-	Bruxelle	France	-	-
285099935115522	-	-	-	-	-	Bruxelle	France	-	-

TABLE 3.2 – Représentation tabulaire de la source de données DS (Suite)

3.2 Les Dictionnaires de Données (DD)

Les dictionnaires de données contiennent un ensemble prédéfini de catégories. Ces catgories sont des connaissances identifiées suivant trois modes :

— **les donnés définies par extension**. Ce sont des listes données à priori telles que

celles des Noms de villes, des pays, des sociétés et d'organismes, les listes des civilités et des sexes *(DDVS)*.

— **les données définies par intention**. Ces connaissances doivent être conformes à un modèle et/ou vérifier des propriétés telles que les expressions régulières ou l'appartenance à un intervalle de valeurs. Les adresses Email ou celles des sites Web ne peuvent être définies que par intention *(DDRE)*.

— **les données définies par mots clés**. La signification des données peut être découvertes grâce à des mots clés. Une partie de la donnée (un ou plusieurs mots) peut servir pour en découvrir le sens. Ces mots clés peuvent être préstockés dans un dictionnaire de nom *(DDKW)*.

Nous avons défini de nouveaux dictionnaires qui permettent, d'une part, d'enrichir la description de chacune des catégories, et d'autre part, de permettre d'établir des liens sémantiques entre les catégories (DDCAT).

3.2.1 Dictionnaire de Données des chaînes de caractères valides (DDVS)

Soit l'ensemble **CategoriesVS** qui regroupe toutes les catégories de données définies par extension. Chaque catégorie est référencée par un nom unique (Cat).
$CategoriesVS = \{Cat_k; k = 1, \alpha\}$, α est le nombre de catégories.
Soit l'ensemble **SubCategoriesVS** qui regroupe toutes les sous-catégories de données définies par extension. Chaque sous-catégorie est référéncée par un nom unique (SubCat).
$SubCategoriesVS = \{SubCat_l; 1 = 1, \beta\}$ β est le nombre de Sous-catégories.
La figure 3.4 ci-dessous réprésente une instance du Dictionnaire de Données des chaînes de caractères valides (DDVS).

Data Dictionnary of Valid Strings : DDVS				
CategoriesVS	SubCatategoriesVS			
CatVS	SubCat1	SubCat2	...	SubCatβ
Cat1	w111, w112, w113...	w121, w122...	...	w1β1, w1β2...
Cat2	w211, w212, w213...	w221, ...	...	w2β1, w2β2...
...	...		...	
	...			...
	...			...
Catα	wα11, wα12...	wα21, ...		wαβ1, ...

FIGURE 3.4 – Un ensemble de catégories et de sous-catégories

Exemple 3.2 : catégories et Sous-catégories

CategoriesVS={CITY, COUNTRY, CONTINENT}

SubCategoriesVS={ENGLISH, FRENCH, ABREVIATION }

La liste des chaînes de caractères valides qui réprésentent des villes est référencée par le nom sémantique de la catégorie CITY. Cette liste peut être regroupée en plusieurs sous-catégories telles que ENGLISH pour répresenter les chaînes des caractères valides en anglais et FRENCH pour répresenter les chaînes des caractères valides en français.

Définition 3.2 : Dictionnaire de Données des 'Valid Strings' (DDVS)

DDVS est un ensemble de valeurs w_{ab} qui peuvent être regroupées en catégories selon leurs sens et éventuellement en sous-catégories.

Soit DDVS= $\{w_{ab}$; a=1,α; $b = 1, \beta$, w$_{ab}$ $\in$Category$_a$ et $SubCategory_b$ $\}$

est un ensemble de données à priori.

Chaque ligne Cat_k contient, pour chaque sous-catégorie $SubCat_{k'}$, un ensemble de valeurs $\{w^1_{kk'}, w^2_{kk'}, ...w^P_{kk'}\}$, P est le nombre de valeurs pour une catégorie donnée et une sous catégorie donnée. •

Exemple 3.3 : dictionnaire de données des chaînes valides (DDVS)

Le dictionnaire de données DDVS contient des données définies par extension, c'est une liste de données à priori telles que les noms de villes, des pays, des sociétés et des organismes.

La table 3.3 est une instance du dictionnaire DDVS.

Le dictionnaire de données DDVS est composé par l'ensemble des catégories *(CATEGORY)*, des valeurs correctes en anglais *(ENGLISH)* et en français *(FRENCH)* et des abréviations de certaines valeurs *(ABREVIATION)*.

DDVS= $\{(CATEGORY, ENGLISH, FRENCH, ABREVIATION, PRIMARYKEY, FOREIGNKEY)_i$, i=1,$n_2\}$, n_2 est la cardinalité de l'ensemble DDVS.

L'orginalité de notre approche réside dans le fait de stocker des données et des liens sémantiques entre elles telles que les dépendances qui réprésentent le concept sémantique « appartenir à » .

Data Dictionary of Valid Strings : DDVS					
CATEGORY	**ENGLISH**	**FRENCH**	**ABBR**	**PKEY**	**FKEY**
CONTINENT	Africa	Afrique		Cont1	
CONTINENT	America	Amérique		Cont2	
CONTINENT	Asia	Asie		Cont3	
CONTINENT	Australia	Australie		Cont4	
CONTINENT	Europe	Europe		Cont5	
COUNTRY	Algeria	Algérie		Ctry001	Cont1
COUNTRY	Egypt	Egypte		Ctry002	Cont1
COUNTRY	Tunisia	Tunisie		Ctry007	Cont1
COUNTRY	China	Chine		Ctry008	Cont3
COUNTRY	France	France		Ctry017	Cont5
CITY	Paris	Paris		Cit001	Ctry017
CITY	Lyon	Lyon		Cit003	Ctry0017
CITY	Nice	Nice		Cit004	Ctry0017
CITY	Beijing	Pékin		City007	Ctry008
CITY	Alger	Alger		Cit008	Ctry001
AIRPORT	Airport Orly	Aéroport d'Orly	ORY	Airport001	Cit001

TABLE 3.3 – Une instance du dictionnaire DDVS

Le script SQL permettant la création du dictionnaire DDVS est :

```
CREATE TABLE DDVS
(CATEGORY VARCHAR(50),
ENGLISH VARCHAR(100),
FRENCH VARCHAR(100),
ABREVIATION VARCHAR(10),
PRIMARYKEY VARCHAR(20),
FOREIGNKEY VARCHAR(20),
CONSTRAINT PKDDVS PRIMARY KEY (PRIMARYKEY));

INSERT INTO DDVS VALUES ('CITY', 'Paris', 'Paris', '', 'CITYFRA000001', 'COUNTRYEUR001');
INSERT INTO DDVS VALUES ('CITY', 'Tunis', 'Tunis', '', 'CITYTUN000001', 'COUNTRYAFR014');
INSERT INTO DDVS VALUES ('CONTINENT', 'Europe', 'Europe', '', 'CONTINENT05', '');
INSERT INTO DDVS VALUES ('COUNTRY', 'France', 'France', 'FRA', 'COUNTRYEUR001', 'CONTINENT05');
```

Le dictionnaire DDVS peut être vu sous une forme « allongée » DDVSTOT.
DDVSTOT= $\{(CATEGORY, SUBCATEGORY, VALIDSTRING)_i$, i=1,$n_3\}$, n_3 est la cardinalité de l'ensemble DDVSTOT.

Le tableau 3.4 donne un exemple de chaînes de caractères considérées valides (Valid Strings), vues sous forme d'une liste. L'intérêt est d'avoir pour toute chaine de caractères sa catégorie et sa sous-catégorie.
Le script SQL qui permet de créer DDVSTOT est :

```
CREATE OR REPLACE VIEW DDVSTOT(CATEGORY, SUBCATEGORY, VALIDSTRING) AS
(SELECT CATEGORY, 'ENGLISH', ENGLISH
FROM DDVS
WHERE ENGLISH IS NOT NULL
UNION
SELECT CATEGORY, 'FRENCH', FRENCH
FROM DDVS
WHERE FRENCH IS NOT NULL
UNION
SELECT CATEGORY, 'ABREVIATION', ABREVIATION
FROM DDVS
WHERE ABREVIATION IS NOT NULL);
```

Data Dictionary of Valid Strings : DDVSTOT		
CATEGORY	**SUBCATEGORY**	**VALIDSTRING**
AIRPORT	ABREVIATION	CDG
AIRPORT	ABREVIATION	ORY
AIRPORT	ENGLISH	Airport charles de Gaulle
AIRPORT	ENGLISH	Airport Orly
AIRPORT	FRENCH	Aéroport Charles de Gaulle
AIRPORT	FRENCH	Aéroport d'Orly
CITY	ENGLISH	Alger
CITY	ENGLISH	Lyon
CITY	ENGLISH	Nice
CITY	ENGLISH	Orly
CITY	ENGLISH	Paris
COUNTRY	ENGLISH	Egypt
COUNTRY	ENGLISH	China
COUNTRY	ENGLISH	France
COUNTRY	ENGLISH	Tunisia
COUNTRY	FRENCH	Egypte
COUNTRY	FRENCH	Chine
COUNTRY	FRENCH	France
COUNTRY	FRENCH	Tunisie

TABLE 3.4 – Une instance du dictionnaire DDVSTOT

Le dictionnaire DDVS permet d'inférer différents liens sémantiques inter-colonnes. En effet, en se basant sur la règle de transitivité des dépendances fonctionnelles par exemple, il est nécessaire de transformer le graphe des dépendances selon la figure ci-dessous (voir figure 3.5).

Les dépendances qui peuvent exister entre les différentes valeurs des catégories sont calculées à partir de plusieurs jointures de DDVS. Le nombre de jointures est calculé selon la longueur maximale dans le graphe de dépendances fonctionnelles du chemin d'un sommet vers la feuille.
Les liens sémantiques déduits à partir du dictionnaire DDVS sont calculés et stockés dans le dictionnaire DDVSLINKS selon le script SQL suivant :

```
CREATE OR REPLACE VIEW DDVSLINKS1
(CATEGORYL, CATEGORYR, ENGLISHL, ENGLISHR, FRENCHL, FRENCHR) AS
(SELECT DISTINCT T1.CATEGORY, T2.CATEGORY, T1.ENGLISH, T2.ENGLISH, T1.FRENCH,
T2.FRENCH
FROM DDVS T1, DDVS T2
WHERE T1.FOREIGNKEY = T2.PRIMARYKEY) ;

CREATE OR REPLACE VIEW DDVSLINKS2
(CATEGORYL, CATEGORYR, ENGLISHL, ENGLISHR, FRENCHL, FRENCHR) AS
(SELECT DISTINCT T2.CATEGORYL, T1.CATEGORYR, T2.ENGLISHL, T1.ENGLISHR,
T2.FRENCHL, T1.FRENCHR
FROM DDVSLINKS1 T1, DDVSLINKS1 T2
WHERE T1.CATEGORYL = T2.CATEGORYR AND T1.ENGLISHL = T2.ENGLISHR) ;

CREATE OR REPLACE VIEW DDVSLINKS3
(CATEGORYL, CATEGORYR, ENGLISHL, ENGLISHR, FRENCHL, FRENCHR) AS
(SELECT * FROM DDVSLINKS1
UNION
SELECT * FROM DDVSLINKS2) ;

CREATE OR REPLACE VIEW DDVSLINKS4
(CATEGORYL, CATEGORYR, ENGLISHL, ENGLISHR, FRENCHL, FRENCHR) AS
(SELECT DISTINCT T2.CATEGORYL, T1.CATEGORYR, T2.ENGLISHL, T1.ENGLISHR,
T2.FRENCHL, T1.FRENCHR
FROM DDVSLINKS3 T1, DDVSLINKS3 T2
WHERE T1.CATEGORYL = T2.CATEGORYR AND T1.ENGLISHL = T2.ENGLISHR) ;

CREATE OR REPLACE VIEW DDVSLINKS
(CATEGORYL, CATEGORYR, ENGLISHL, ENGLISHR, FRENCHL, FRENCHR) AS
(SELECT DISTINCT CATEGORYL, CATEGORYR, ENGLISHL, ENGLISHR, FRENCHL, FREN-
CHR FROM DDVSLINKS3
UNION
SELECT DISTINCT CATEGORYL, CATEGORYR, ENGLISHL, ENGLISHR, FRENCHL, FREN-
CHR FROM DDVSLINKS4) ;
```

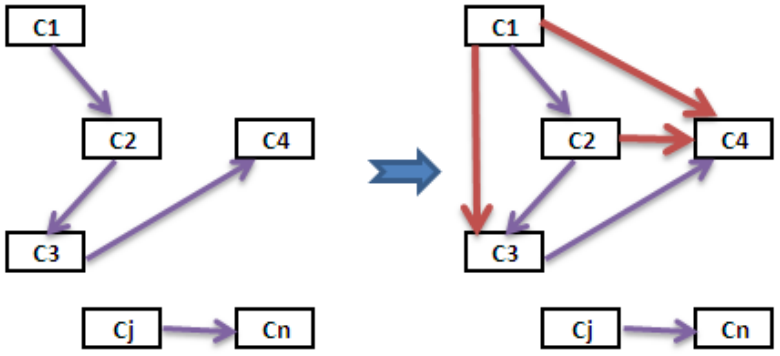


FIGURE 3.5 – Un ensemble de liens sémantiques inter-colonnes

Le tableau 3.5 donne des exemples de dépendances fonctionnelles pré-stockées dans DDVSLINKS. Ainsi, pour chaque paire de dépendances fonctionnelles, on a le couple de

valeurs associées pour chaque sous-catégorie.

DDVSLINKS					
DEPENDENCIES $L \rightarrow R$					
CATEGORYL	**CATEGORYR**	**ENGLISHL**	**ENGLISHR**	**FRENCHL**	**FRENCHR**
COUNTRY	CONTINENT	Tunisia	Africa	Tunisie	Afrique
COUNTRY	CONTINENT	France	Europe	France	Europe
CITY	COUNTRY	Beijing	China	Pékin	Chine
CITY	COUNTRY	Paris	France	Paris	France
CITY	COUNTRY	Rome	Italy	Rome	Italie
AIRPORT	CITY	Airport Orly	Orly	...	...

TABLE 3.5 – Une instance du dictionnaire DDVSLINKS

Le schéma conceptuel des méta-données peut être reprsenté par le diagramme de classes UML de la figure ci-dessous 3.6.

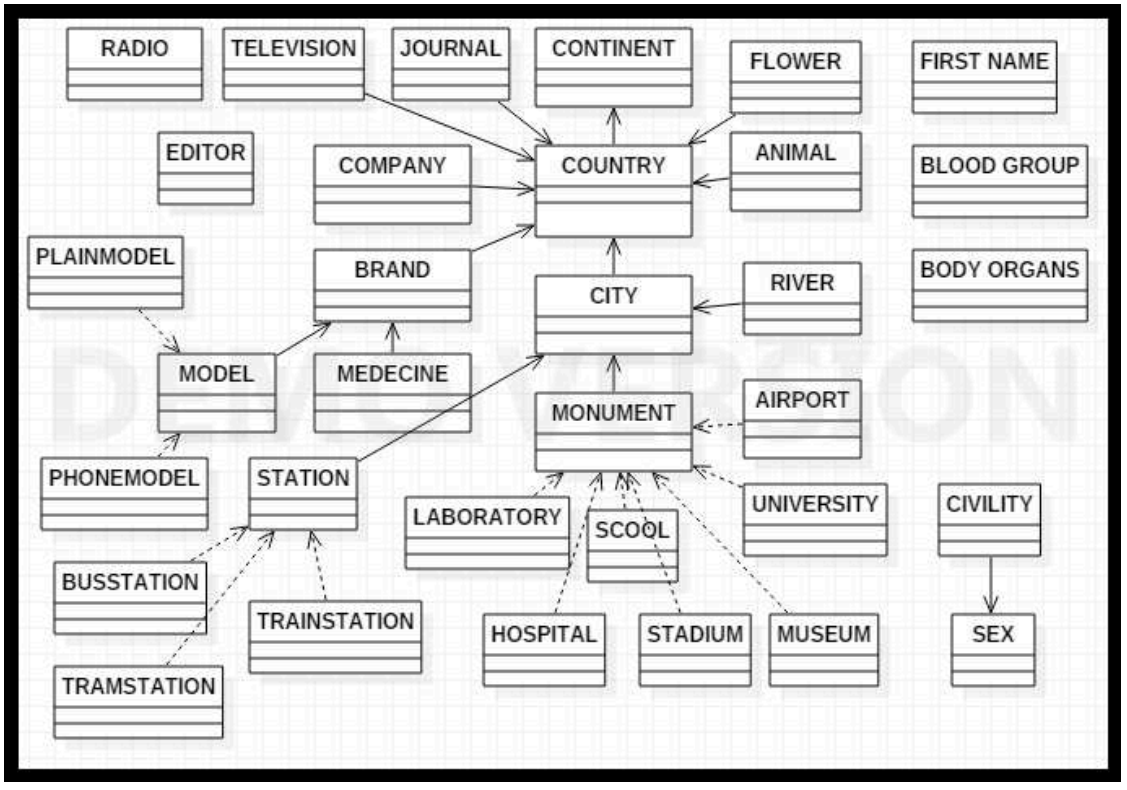


FIGURE 3.6 – Schéma conceptuel des méta-données et des liens sémantiques

3.2.2 Dictionnaire de Données des Expressions Régulières (DDRE)

Soit l'ensemble **CategoriesRE** qui regroupe toutes les catégories de données définies par intention. Chaque catégorie est référencée par un nom unique (Cat).

$CategoriesRE = \{Cat_k; k = \alpha + 1, \alpha + \alpha'\}$, α' est le nombre d'expressions régulières qui définissent les catégories par intention.

Soit $DDRE = \{e_a; a = 1, \alpha'\}$
e_a est une expression régulière donnée à priori.
DDRE est un ensemble d'expressions régulières répresentant chacune une catégorie. La figure ci-dessous réprésente une instance du Dictionnaire de Données (DDRE).

Data Dictionnary of Regular Expressions : DDRE	
CategoriesRE	Regular Expressions
Catα+1	e1
Catα+2	e2
...	...
Catα+α'	eα'

FIGURE 3.7 – Ensemble de catégories définies par des expressions régulières

Exemple 3.4 : exemple d'une expression régulière
Pour la catégorie Email, toutes les chaînes de caractères valides sont répresentées (doivent être vérifiées) par l'expression régulière correspondante :
ˆ[a-zA-Z0-9._%]+@[a-zA-Z0-9.-]+ \.[a-zA-Z]2,4$

Définition 3.3 : Dictionnaire de Données des Expressions régulières (DDRE)
Le dictionnaire des expressions régulières DDRE est composé de l'ensemble des catégories *(CATEGORY)* définies par intention. L'ensemble des expressions régulières permet de définir des données de plusieurs types (chaîne de caratères, date et numérique).
DDRE= $\{(CATEGORY, REGULAREXPRESSION, PRIMARYKEY, FOREIGNKEY)_f$, f=1,$n_6\}$, n_6 est la cardinalité de l'ensemble DDRE. •
Le tableau 3.6 donne un exemple d'expressions régulières.

CATEGORY-SUBCATEGORY	REGULAR EXPRESSION
EMAIL	^[a-zA-Z0-9._%]+@[a-zA-Z0-9.-]+ \.[a-zA-Z]2,4$
DATE-DDMMYYYY	^(0 ?[1-9]| [12][0 − 9]|3[01])[/−](0?[1 − 9]|1[012])[/−]4$
DATE-YYYYMMDD	^([0-9]4)-([0-9]1,2)-([0-9]1,2)$
DATE-MMDDYY	^((0[1-9])|(1[0-2]))(|-)((0[1-9])|(1[0-9])|(2[0-9])|(3[0-1]))(|-)([0-9]2)$
DATETIME-MMDDYYYY-hh :mm :ss	^([0-9]1,2)()([0-9]1,2)()([0-9]4) (([01][0-9]|[2][0-3]) :([0-5][0-9]) :[0-5][0-9])$
DATETIME-MMDDYYYY-hh :mm	^([0-9]1,2)()([0-9]1,2)()([0-9]4) (([01][0-9]|[2][0-3]) :([0-5][0-9]))$
24HOURTIME	^([0-1] ?[0-9]|2[0-4]) :([0-5][0-9])(:[0-5][0-9]) ?$
TEMPERATURE	^(- ?[0-9]\d*(.\d+) ?) ?(°C|°F)$
URL	^(http|https) :) ?(www[.]) ?([a-zA-Z0-9]|-)+([.][a-zA-Z0-9(-|=|?) ?]+)+$
MASTER CARD NUMBER	^5[1-5][0-9]0-914–$
INTEGER	^[[: digit :]]*$

TABLE 3.6 – Exemples d'expressions régulières (DDRE)

Le script SQL qui permet de créer le dictionnaire DDRE est :

```
CREATE TABLE DDRE
(CATEGORY VARCHAR(50),
REGEXPR VARCHAR(1000),
PRIMARYKEY VARCHAR(20),
FOREIGNKEY VARCHAR(20),
CONSTRAINT PKDDRE PRIMARY KEY (PRIMARYKEY));

INSERT INTO DDRE VALUES
('EMAIL','^[A − Za − z0 − 9._%−] + @[A − Za − z0 − 9.−] + .[A − Za − z]2, 4$','EMAIL001', '');
INSERT INTO DDRE VALUES
('DATE-YYYYMMDD', '^([0 − 9]4) − ([0 − 9]1, 2) − ([0 − 9]1, 2)$','DATE001', '');
```

3.2.3 Dictionnaire de données des mots clés (DDKW)

Certaines catégories sont découvertes grâce à des mots clés et non pas en utilisant toute la chaîne de caractères.

Définition 3.4 : dictionnaire de données des mots clés (DDKW)

DDKW est un ensemble de valeurs w_{ab} qui peuvent être regroupées en catégories selon leurs sens et éventuellement en sous-catégories.
Soit DDKW= $\{w_{ab}$; a=1,α; $b = 1, \beta$, w$_{ab}$ $\in$Category$_a$ et $SubCategory_b$ $\}$
est un ensemble de données à priori.
Chaque ligne Cat_k contient, pour chaque sous-catégorie $SubCat_{k'}$, un ensemble de valeurs $\{w^1_{kk'}, w^2_{kk'}, ...w^P_{kk'}\}$, P est le nombre de valeurs pour une catégorie donnée et une sous catégorie donnée. •

Data Dictionary of Key Words : DDKW		
CATEGORY	**ENGLISH**	**FRENCH**
ADRESS	Street	Rue
ADRESS	St.	R.
ADRESS	Avenue	Avenue
ADRESS	Av.	Av.
ADRESS	Place	Place
ADRESS	Pl.	Pl.
ADRESS		Boulevard
ADRESS		Bld
ADRESS		Quai
ADRESS		Pont
ADRESS	Square	Square
STUDYORGANIZATION	University	Université
STUDYORGANIZATION	School	École
HEALTHORGANIZATION	Hospital	Hôpital
CULTUREORGANIZATION	Theatre	Théatre
CULTUREORGANIZATION	Cinema	Cinéma

TABLE 3.7 – Une instance du dictionnaire DDKW

3.2.4 Dictionnaire de Donnés des CATégories (DDCAT)

L'ensemble des catégories qui correspondent aux données définies par extension contenues dans DDVS peuvent être sémantiquement liées, alors que l'ensemble des catégories qui correspondent aux données définies par intention contenues dans DDRE ne le sont pas. Il existe cependant des liens sémantiques entre ces deux ensembles disjoints. Le dictionnaire de toutes les catégories est alors créé pour expliciter les liens évetuels entre les différents types de catégories, et pour stocker certains détails supplémentaires tels que le type de données, les contraintes et les commentaires qui portent sur ces catégories.

Définition 3.5 : dictionnaire des catégories (DDCAT)

Le dictionnaire des catégories contient la liste des catégories définies précédemment par extension et par intention telles que CITY, COUNTRY, CONTINENT, AIRPORT, EMAIL, URL et ZIPCODE.
Rappelons que, par exemple, CITY (respectivement COUNTRY) est le nom attribué à la liste des chaînes de caractères valides qui réprensentent sémantiquement des villes (respectivement des pays). Il en est de même pour le mot URL qui désigne la catégorie représentée par l'expression régulière correspondante à un site Web.
DATAFREQUENCY renseigne sur la répartition de données dans la colonne.
DDCAT= $\{(CATEGORY, ENGLISH, FRENCH, PRIMARYKEY, FOREIGNKEY, DATATYPE, DATAFREQUENCY, COMMENTR, CONSTRAINTS)_i$, i=1,$n_1\}$,

n_1 est la cardinalité de l'ensemble DDCAT. •

Les tableaux 3.8 et 3.9 réprésentent une instance du dictionnaire de données DDCAT. Le script SQL qui permet de créer le dictionnaire DDCAT est :

```
CREATE TABLE DDCAT
(CATEGORY VARCHAR(30),
ENGLISH VARCHAR2(30),
FRENCH VARCHAR2(30),
PRIMARYKEY VARCHAR(30),
FOREIGNKEY VARCHAR(30),
DATATYPE VARCHAR(10),
DATAFREQUENCY VARCHAR(10),
SIMILARITYALGO VARCHAR2(30),
SIMILARITYTHRESHOLD NUMBER,
COMMENTR VARCHAR(200));

INSERT INTO DDCAT VALUES ('DDVS', 'AIRPORT', 'AÉROPORT', 'VS0001',
'VS0005','String','LOW','Soundex+Jaro-Winkler','3','List of airports in a city');
INSERT INTO DDCAT VALUES ('DDVS', 'BLOODGROUP', 'GROUPESANGUIN', 'VS0004',
'','String','HEIGHT','Edit-distance','0','List of blood groups');
INSERT INTO DDCAT VALUES ('DDVS', 'FIRSTNAME', 'PRENOM', 'VS0010',
'','String','HEIGHT','Edit-distance','2','List of first First names');
```

Data Ditionary of all CATegories : DDCAT				
CATEGORY	**CAT-ENGLISH**	**CAT-FRENCH**	**PKEY**	**FKEY**
DDVS	AIRPORT	AÈROPORT	VS0001	VS0003
DDVS	BLOODGROUP	GROUPESANGUIN		
DDVS	CITY	VILLE	VS0003	VS0002
DDVS	COMPANY	ENTREPRISE	VS0004	VS0005
DDVS	COUNTRY	PAYS	VS0005	VS0006
DDVS	CONTINENT	CONTINENT	VS0006	
DDVS	FIRSTNAME	PRÉNOM	VS0007	
DDVS	FRMUTUAL	FRMUTUELLE	VS0008	
DDVS	GENDER	GENRE	VS0009	
DDVS	MEDCINE	MÈDICAMENT	VS0010	
DDVS	MEDICALSPECIALIZATION	SPÉCIALITÉMÉDICALE	VS0011	
DDVS	MUSEUM	MUSÉE	VS0012	VS0003
DDRE	DATE-DDMMYYYY	DATE-JJMMAAAA	RE0001	
DDRE	DATE-MMDDYYYY	DATE-MMJJAAAA	RE0002	
DDRE	DATE-YYYYMMDD	DATE-AAAAMMJJ	RE0003	
DDRE	URL	URL	RE0004	
DDRE	ZIPCODE	CODEPOSTALE	RE0005	VS0003

TABLE 3.8 – Une instance du dictionnaire de données DDCAT (1)

Data Ditionary of all CATegories : DDCAT				
CAT-ENGLISH	**DATATYPE**	**DATAFREQUENCY**	**SIMILARITYALGO**	**SIMTHRESHOLD**
AIRPORT	String	LOW	Soundex+Jaro-Winkler	3
BLOODGROUP	String	HEIGHT	Edit-distance	0
CITY	String	HEIGHT		
COMPANY	String	HEIGHT	Edit-distance	
COUNTRY	String	LOW	Edit-distance	
CONTINENT	String	HEIGHT	Edit-distance	
FIRSTNAME	String	HEIGHT	Edit-distance	2
FRMUTUAL	String	MEDUIM	Edit-distance	
GENDER	String	MEDUIM	Edit-distance	0
MEDCINE	String	MEDUIM	Edit-distance	2
MEDICALSPECIALIZATION	String	MEDUIM	Edit-distance	2
MUSEUM	String	MEDUIM	Edit-distance	
DATE-DDMMYYYY	Date	HEIGHT	Edit-distance	0
DATE-MMDDYYYY	Date	HEIGHT	Edit-distance	0
DATE-YYYYMMDD	Date	HEIGHT	Edit-distance	0
URL	String	MEDUIM	Edit-distance	0
ZIPCODE	Number	MEDUIM	Edit-distance	

TABLE 3.9 – Une instance du dictionnaire de données DDCAT (2)

le dictionnaire DDCAT permet d'inférer les différents liens sémantiques tels que les dépendances fonctionnelles qui peuvent exister entre les catégories. La figure 3.8 réprésente les liens de base entre certaines catégories.

Remarque :
Il est nécessaire de déduire tous les autres liens en se basant sur la transivité des dépendances fonctionnelles : Si $X \rightarrow Y$ et $Y \rightarrow Z$ Alors $X \rightarrow Z$.
En effet, dans une source de données, toutes les catégories ne sont pas forcément présentes. Ainsi les dépendances fonctionelles de base ne suffisent pas pour déduire tous les liens possibles dans la source de données.

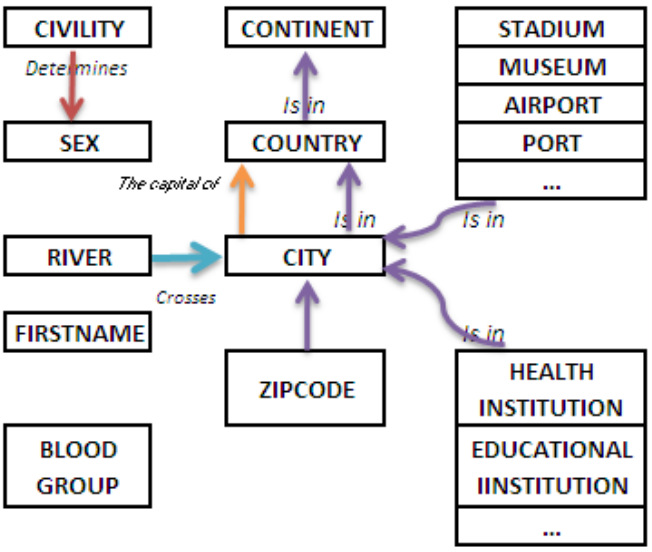


FIGURE 3.8 – Liens sémantiques entre catégories (DDVS+DDRE)

Le DDCAT peut être vue sous une forme liens sémantiques calulés et stockés dans DDCATLINKS en réalisant des auto-jointures.
Il existe plusieurs niveaux de liens sémantiques. C'est la longueur maximale du chemin entre une source de dépendance fonctionnelle et la feuille.

Exemple 3.5 :
Le nombre de jointures est égal à 3 car le nombre de dépendances fonctionnelles depuis une racine telle que AIRPORT vers une feuille telle que CONTINENT est égal à 3.

Le script SQL qui permet de créer DDCATLINKS est le suivant :

```
CREATE OR REPLACE VIEW DDCATLINKS1(CATEGORYL, CATEGORYR) AS
(SELECT T1.ENGLISH, T2.ENGLISH FROM DDCAT T1, DDCAT T2
WHERE T1.FOREIGNKEY = T2.PRIMARYKEY);

CREATE OR REPLACE VIEW DDCATLINKS2(CATEGORYL, CATEGORYR) AS
(SELECT distinct T1.CATEGORYL, T2.CATEGORYR
FROM DDCATLINKS1 T1, DDCATLINKS1 T2
WHERE T1.CATEGORYR = T2.CATEGORYL);

CREATE OR REPLACE VIEW DDCATLINKS3(CATEGORYL, CATEGORYR) AS
(SELECT distinct T1.CATEGORYL, T2.CATEGORYR
FROM DDCATLINKS2 T1, DDCATLINKS2 T2
WHERE T1.CATEGORYR = T2.CATEGORYL);

CREATE OR REPLACE VIEW DDCATLINKS(CATEGORYL, CATEGORYR) AS
(SELECT * FROM DDCATLINKS1
UNION
SELECT * FROM DDCATLINKS2
UNION
SELECT * FROM DDCATLINKS3);
```

Nous présentons dans le tableau 3.10 des exemples de dépendances fonctionnelles pré-stockées dans DDCATLINKS. Elles sont inférées depuis DDCAT ;

DDCATLINKS	
DEPENDENCIES : $L \rightarrow R$	
CATEGORYL	**CATEGORYR**
AIRPORT	CITY
AIRPORT	COUNTRY
AIRPORT	CONTINENT
CITY	COUNTRY
CITY	CONTINENT
COUNTRY	CONTINENT
ZIPCODE	CITY
ZIPCODE	COUNTRY
CIVILITY	GENDER

TABLE 3.10 – Une instance du dictionnaire de données DDCATLINKS

3.2.5 Dictionnaire des contraintes sur les catégories (DDCATCONSTR)

Des contraintes sont créés pour aider, d'une part à l'homogéneisation et la standardisation de données, et d'autre part, pour détecter et corriger les anomalies.

Définition 3.6 : dictionnaire des contraintes sur les catégories (DDCATCONSTR)

Le dictionnaire des contraintes contient la liste des contraintes définies sur les catégories et donc la liste des contraintes sur les données.

Nous proposons trois types de contraintes à savoir les contraintes de transformation, les contraintes conditionnelles et les contraintes d'appartenance :

- Les contraintes de transformation : permettent de préciser des règles de présentation sur les données afin d'homogénéiser et standardiser les données.
- Les contraintes conditionnelles : déterminent des transformations sur les données selon des conditions.
- Les contraintes d'appartenance : permettent de définir l'intervalle ou la liste de données.

DDCATCONSTR= $\{(PRIMARYKEY, CATEGORY, TYPECONST, FUNCTIONCONSTR1, FUNCTIONCONSTR2, FUNCTIONCONSTR3)_i$, i=1,$n_7\}$, n_7 est la cardinalité de l'ensemble DDCATCONSTR. •

Le script SQL qui permet de créer le dictionnaire DDCATCONSTR est :

```
CREATE TABLE DDCATCONSTR
(PRIMARYKEY VARCHAR(30),
CATEGORY VARCHAR(30),
TYPECONST VARCHAR(5),
FUNCTIONCONSTR1 VARCHAR(10), – Si condition
FUNCTIONCONSTR2 VARCHAR(10), – Alors action
FUNCTIONCONSTR3 VARCHAR(10)) ; – Nombre maximal de valeurs à fusionner

INSERT INTO DDCATCONSTR VALUES ('0001', 'CITY', 'TR', 'INITCAP', '', '') ;
INSERT INTO DDCATCONSTR VALUES ('0002', 'CONTINENT', 'TR', 'UPPER', '', ''); INSERT
INTO DDCATCONSTR VALUES ('0003', 'COUNTRY', 'TR', 'UPPER', '', '') ;
INSERT INTO DDCATCONSTR VALUES ('0004', 'FIRSTNAME', 'TR', 'INITCAP', '', '') ;
INSERT INTO DDCATCONSTR VALUES ('0010', 'GENDER', 'IF', 'Femme', 'F', '2') ;
INSERT INTO DDCATCONSTR VALUES ('0017', 'CIVILITY', 'IF', 'Madame', 'Mme', '1') ;
INSERT INTO DDCATCONSTR VALUES ('0027', 'DATE', 'TR', 'YYYY-MM-DD', '', '') ;
INSERT INTO DDCATCONSTR VALUES ('0028', 'BLOODGROUP', 'IN', 'A-,A+,B-,B+,O-
,O+,AB-,AB+', '', '') ;
INSERT INTO DDCATCONSTR VALUES ('0029', 'DISTANCECM', 'TR', 'MULTIPLICATION',
'100', '') ;
INSERT INTO DDCATCONSTR VALUES ('0030', 'DISTANCEKM', 'TR', 'DIVISION', '1000', '') ;
```

3.3 La découverte du schéma sémantique des données

La catégorisation sémantique des données intra-colonne consiste à établir toutes les relations qui peuvent exister entre l'ensemble DS, d'une part, et les ensembles CategoriesDS et CategoriesRE, d'autre part.
La découverte des liens sémantiques inter-colonnes porte sur la recherche des relations entre les élements de DS.

La figure 5.14 réprésente schématiquement les relations qui peuvent exister entre la source de données et les dictionnaires de données.

Remarques :
- La catégorisation intra-colonne consiste à chercher si chaque valeur de DS appartient à DDVS ou si elle vérifie une expression régulière de DDRE.
- Une valeur de DS peut exister plusieurs fois dans DDVS.
- Une valeur de DS peut vérifier plusieurs expressions régulières.
- La recherche des liens sémantiques inter-colonnes consiste à vérifier s'il existe une relation d'ordre (<,=,>) entre celles-ci ou encore si une règle de dépendance existe telle que « la seule connaissance d'une valeur de DS appartenant à une colonne permet de trouver la valeur correspondante dans une autre colonne ».

La figure 3.9 donne une idée sur les différents types de relations qui existent entre la source de données DS et les dictionnaires de données ainsi que celles entre les données elles-mêmes

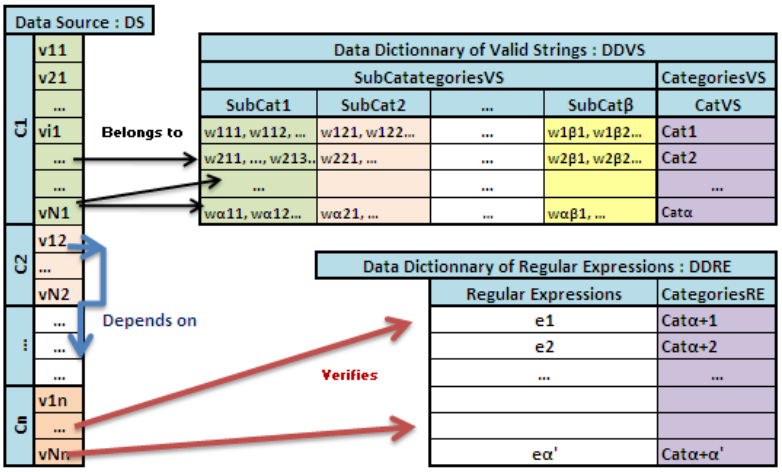


FIGURE 3.9 – Catégorisation sémantique des données et liens inter-colonnes

Nous présentons ci-dessous les définitions des concepts fondamentaux pour notre processus de catégorisation des données.

Définition 3.7 : valeur valide syntaxiquement

Une valeur v_{ik} de DS est syntaxiquement correcte ssi $v_{ik} \in DDVS$ *ou* v_{ik} vérifie au

moins une expression régulière qui appartient à DDRE. •

Remarques :
- On traite les valeurs exactes $v_{ik} \in DDVS$. Les valeurs similaires en utilisant une mesure de distance de similarité (Levenshtein, Jaro-Winkler) ne sont pas considérés, dans une première étape, comme syntaxiquement valides.
- v_{ik} peut appartenir à plusieurs catégories.

Exemple 3.6 : valeur valide syntaxiquement
- $Paris \in DDVS$; $Beijing \in DDVS$;
- houda@cnam.fr vérifie une expression de DDRE ;
- 16/12/2016 vérifie une expression de DDRE ;
- $Pariss \notin DDVS$
- fb@@Lipn.fr ne vérifie pas au moins une expression de DDRE.

Définition 3.8 : catégorie dominante

Soit A l'ensemble des valeurs d'une colonne C_j (une instance de C_j) est :
A = {v$_{1j}$, v_{1j},..., v_{pj}, v_{p+1j}, v_{Nj}}
A ={v_{1j}, v_{1j},..., v_{pj}} $\cup$ {v$_{p+1j}$, v_{Nj}}
A =A1 $\cup$ $A2$. L'ensemble A1 (respectivement A2) contient des valeurs qui appartiennent à la catégorie Cat_1 (respectivement Cat_2). Soit $Card_1$ (respectivement $Card_2$) la cardinalité de l'ensemble A1 (respectivement A2).
Cat_1 réprésente la catégorie dominante (notée Cat_d) ssi $Card_1 > Card_2$. •

Définition 3.9 : valeur valide sémantiquement

Une valeur v_{ik} de DS est sémantiquement correcte ssi $v_{ik} \in$ Cat$_d$ et $v_{ik} \in$ SubCat$_d$ •

Exemple 3.7 : valeur valide sémantiquement

— Paris appartient à la catégorie CITY en français.
— Pékin appartient à la catégorie CITY en français.
— Beijing n'appartient pas à la catégorie CITY en français, elle appartient à la catégorie CITY en anglais .
— www.lipn.fr appartient à la catégorie URL (vérifie l'expression régulière correspon-

dante) et n'appartient pas à la catégorie EMAIL (ne vérifie pas l'expression régulière correspondante).

3.3.1 Les mesures et les règles de catégorisation des données

Nous nous basons sur un ensemble de mesures pour valider le choix de la catégorie dominante et éventuellement la sous-catégorie dominante.
Les mesures peuvent être regroupées en trois catégories à savoir celles qui portent sur toute la source de données, celles qui ne concernent qu'une colonne à la fois (mesures intra-colonne) et enfin celles qui portent sur deux colonnes (mesures inter-colonnes)

3.3.1.1 Les mesures globales

Plusieurs mesures globales peuvent être calculées afin de donner une idée sur le contenu de la source de données :

— M000 : nombre total de lignes dans la source de données.
— M001 : nombre total de colonnes dans la source de données.
— M002 : nombre total de valeurs possibles dans la source de données.
— M003 : nombre total de valeurs nulles dans la source de données.
— M004 : nombre total de valeurs non nulles dans la source de données.
— M005 : nombre total de valeurs syntaxiquement invalides dans la source de données.
— M006 : nombre total de valeurs syntaxiquement valides dans la source de données.
— M007 : nombre total de lignes en double dans la source de données. Toutes les colonnes sont prises en considération.
— M008 : nombre total de lignes en double dans la source de données. Seules les colonnes (Attributs clés) sont prises en considération.
— M009 : nombre total de lignes similaires dans la source de données. Seules les colonnes (Attributs clés) sont prises en considération.
— M010 : nombre de colonnes en double.
— M011 : sous-catégorie dominante dans la source de données.

3.3.1.2 Les mesures intra-colonne

Les mesures qui portent sur une colonne devraient permettre d'avoir une idée sur le taux de remplissage d'une colonne, sur ses validités syntaxique et sémantique. Le nombre de valeurs distinctes ainsi que les types de valeurs peuvent aider à la catégorisation de données.

Le nombre de mots à considérer dans une même colonne peut permettre d'étudier une dépendance sémantique intra-colonne. Les mesures considérées sont les suivantes :

- M101 : nombre de valeurs nulles dans la colonne
- M102 : nombre de valeurs non nulles dans la colonne
- M103 : nombre de valeurs distinctes dans la colonne
- M104 : nombre de valeurs syntaxiquement invalides dans la colonne
- M105 : nombre de valeurs syntaxiquement valides dans la colonne
- M106 : nombre de valeurs distinctes parmi les valeurs syntaxiquement valides dans la colonne
- M107 : nombre de valeurs distinctes parmi les valeurs syntaxiquement invalides dans la colonne
- M108 : nombre de catégories par colonne
- M109 : nombre de sous-catégories par colonne
- M110 : la longueur minimale des chaînes de caractères par colonne
- M111 : la longueur maximale des chaînes de caractères par colonne
- M112 : la longueur moyenne des chaînes de caractères par colonne
- M113 : la valeur minimale des numériques par colonne
- M114 : la valeur maximale des numériques par colonne
- M115 : la valeur moyenne des numériques par colonne
- M116 : la valeur médiane des numériques par colonne
- M117 : la valeur de l'écart type des numériques par colonne
- M118 : la valeur minimale des dates (la plus ancienne) par colonne
- M119 : la valeur maximale des dates (la plus récente) par colonne
- M120 : le nombre de « mots » à considérer au sein d'une même colonne
- M121 : le nombre de valeurs par catégorie
- M122 : le nombre de valeurs par sous-catégorie

3.3.1.3 Les mesures inter-colonnes

Les mesures liées à deux colonnes de la source réprésentent certains liens sémantiques.

- M201 : relation sémantique (=) entre deux colonnes
- M202 : relation sémantique (<) entre deux colonnes
- M203 : relation sémantique (>) entre deux colonnes
- M204 : dépendances fonctionnelles entre deux colonnes

3.3.1.4 Les règles de catégorisation des données

Nous proposons un ensemble de règles déduites à partir des différentes mesures afin d'aider à la prise de décision. Nous utilisons un ensemble de ratios :

- R001 : M003/M002
- R002 : M005/M004
- R003 : M006/M004
- R101 : M101/M000
- R102 : M102/M000
- R103 : M103/M102
- R104 : M104/M102
- R105 : M105/M102
- R106 : M106/M102
- R201 : M201/M000
- R202 : M202/M000
- R203 : M203/M000
- R204 : M204/M000

Les règles déduites sont :

- Règle1 : si $R101 \geq \epsilon_1$ alors la colonne contient plusieurs valeurs nulles, alors la colonne n'est pas recommendée pour la partie droite d'une dépendance fonctionnelle.
- Règle2 : si $R104 \geq \epsilon_2$ alors la colonne contient plusieurs valeurs syntaxiquement invalides, alors la colonne n'est pas recommendée pour la partie droite d'une dépendance fonctionnelle.
- Règle3 : si $R106 \geq \epsilon_5$ alors la colonne peut être considérée comme une clé primaire.
- Règle4 : si $R201 \geq \epsilon_6$ alors il s'agit de deux colonnes identiques, il est recommandé de supprimer une des deux.
- Règle5 : si le domaine de définition est numérique alors les mesures M113 et M114 permettent de définir un domaine de définition de la colonne.
- Règle6 : si $M120 \geq 2$ alors la colonne peut être divisée en plusieurs colonnes.
- Règle7 : si $R105 \geq \epsilon_7$, M108=1 et $M109 \geq 2$ alors la colonne nécessite une homogénéisation de la sous-catégorie.
- Règle8 : si M108=1 et $R104 \neq 0$ alors la colonne nécessite des corrections syntaxiques.
- Règle9 : si $R202 \geq \epsilon_7$ alors il existe une relation de type '$<$' entre les deux colonnes.
- Règle10 : si $R203 \geq \epsilon_8$ alors il existe une relation de type '$>$' entre les deux colonnes.
- Règle11 : si $R204 \leq \epsilon_9$ alors il faut corriger la violation de dépendance fonctionnelle.

Dans une première étape, la catégorisation des colonnes consiste à trouver la catégorie sémantique de chaque colonne de la source tout en se basant sur DDCAT, DDVS et DDRE,

le type de donnés, les contraintes et les commentaires.
Les liens inter-colonnes (inter-catégories) sont étudiés dans une deuxième étape.

3.3.2 Algorithmes de diagnostics d'une source de données

Nous présentons dans cette section, les différentes étapes de la découverte de schéma sémantique sous forme d'un algorithme de diagnostics d'une source de données :

- Calculer les mesures et les ratios
- Déterminer les catégories et les sous-catégories, les types de données pour chaque colonne de la source de données et remplir les rapports DRDIAGNOGLOBAL et DRDIAGNOINTRACOL.
- Découvrir les dépendances fonctionnelles et les liens sémantiques qui peuvent exister entre les colonnes et remplir le rapport DRDIAGNOINTERCOL.

Algorithm 1 Semantic categorization

Require:

Data Source : DS

Data Dictionnaries : DDCAT, DDCATLINKS, DDCATCONSTR, DDVS, DDVSTOT, DDVSLINKS, DDRE

Ensure:

Data Report (a diagnostic of all the data source columns) : DRDIAGNOINTRACOL

Data Report (functional dependencies links) and (relational [<, =, >] links) : DRDIAGNOINTERCOL

Data Report : DRDIAGNOGLOBAL

Begin Algorithm Semantic categorization

// Calculate the different intra-measures and ratios using DDVS and DDRE

// Discover Categories and SubCategories intra-each-column

// Fill the data report DRDIAGNOINTRACOL

// Diagnostic intra-column

DIAGNOSTICDDVS('DS') ;

DIAGNOSTICDDRE('DS') ;

DIAGNOSTICDSINTRACOL('DS') ;

// Calculate the different inter-measures and ratios

//Discover semantic dependencies and relationships inter-columns

// Fill the data report DRDIAGNOINTERCOL

// Diagnostic inter-columns

DiscoverSEMANTICDFLinks('DS') ;

DiscoverSEMANTICLSLinks('DS') ;

DIAGNOSTICDSINTERCOL('DS') ;

// Calculate the different global-measures and ratios for the whole source

// Fill the data report DRDIAGNOGLOBAL

DIAGNOSTICDS('DS') ;

End Algorithm Semantic categorization

L'algorithme de catégorisation des colonnes (Algorithme 1) est composé de plusieurs procédures. Il se base sur le dictionnaire DDVS et ses dérivés. Les procédures sont détaillées ci-dessous.

La procédure DIAGNOSTICDDVS permet, dans un premier temps, de calculer les différentes mesures et ratios liés à une source de données. Dans un second temps, elle permet de reconnaître la catégorie, la sous-catégorie et le type de données de chaque colonne.

Elle est composée de deux procédures : Compute-IntraColMeasuresRatiosVS et Discover-CategoriesVS.

Algorithm 2 Procedure DIAGNOSTICDDVS

Require: DS

Ensure: DRDIAGNOINTRACOL

// Calculate : Category, SubCategory, Data Type, Comments, Constraints and Intra Column Measures

Begin

Compute-IntraColMeasuresRatiosVS

Discover-CategoriesVS

End Procedure DIAGNOSTICDDVS

Algorithm 3 Procedure Compute-IntraColMeasuresRatiosVS

// Calculate the IntraColMeasuresRatios (M1..) (R1..)

Begin

for each i in 1..NBRCOL **do**

for each j in 01...nm1 **do**

$M1j \leftarrow ComputeDDVS(M1j)$

Insert(M1j, DRDIAGNOINTRACOL)

end for

for each l in 001...nr **do**

$Rl \leftarrow ComputeR(R1l)$

Insert(Rl, DRDIAGNOINTRACOL)

end for

end for

End Algorithm Compute-IntraColMeasuresVS

Algorithm 4 Procedure Discover-CategoriesVS

```
// Discover Category, SubCategory, Datatype, Comments, Constraints
Begin
for each i in 1..NBRROWS do
  for each j in 1..NBRCOL do
    if v_ij ∈ DDVSTOT then
      Insert(Category, SubCategory, DataType, Comments, Constraints, DRDIA-
      GNOINTRACOL)
    end if
  end for
end for
End Algorithm Discover-CategoriesVS
```

L'algorithme de catégorisation des colonnes est composé de plusieurs procédures. Il se base sur le dictionnaire DDRE. Les procédures sont détaillées ci-dessous.
La procédure DIAGNOSTICDDRE permet, dans un premier temps, de calculer les différentes mesures et ratios liés à une source de données. Dans un second temps, elle pemet de reconnaître la catégorie et le type de données de chaque colonne. Elle est composée de deux procédures : Compute-IntraColMeasuresRatiosRE et Discover-CategoriesRE.

Algorithm 5 Procedure DIAGNOSTICDDRE

```
Require: DS
Ensure: DRDIAGNOINTRACOL
// Calculate : Category, Data Type, Comments, Constraints and Intra Column Measures

Begin
Compute-IntraColMeasuresRatiosRE
Discover-CategoriesRE
End Procedure DIAGNOSTICDDRE
```

Algorithm 6 Procedure Compute-IntraColMeasuresRatiosRE

```
// Calculate the IntraColMeasuresRatios (M1..) (R1..)
Begin
for each i in 1..NBRCOL do
  for each j in 01...nm1 do
    M1j ← ComputeDDRE(M1j)
    Insert(M1j, DRDIAGNOINTRACOL)
  end for
  for each l in 001...nr do
    Rl ← ComputeR(R1l)
    Insert(Rl, DRDIAGNOINTRACOL)
  end for
end for
End Algorithm Compute-IntraColMeasuresRE
```

Algorithm 7 Procedure Discover-CategoriesRE

```
// Discover Category, Datatype, Comments, Constraints
Begin
for each i in 1..NBRROWS do
  for each j in 1..NBRCOL do
    if v_ij  Verfies one of the regular expressions in DDRE then
      Insert(Category, DataType, Comments, Constraints, DRDIAGNOINTRACOL)
    end if
  end for
end for
End Algorithm Discover-CategoriesRE
```

Les étapes précédentes ont permis de calculer pour chacune des colonnes de la source de données les différentes mesures et certains ratios en utilisant les dictionnaires DDVS et DDRE. Les catégories et les sous-catégories de chacune des colonnes reconnues complètent le diagnostic intra-colonne. Le résultat est donné dans le rapport final DRDIAGNOINTRACOL grâce à la procédure DIAGNOSTICDSINTRACOL présentée ci-dessous.

Algorithm 8 Procedure DRDIAGNOINTRACOL

// Established the intra-column data report
Begin
Join the two reports DRDIAGNOINTRACOL using DDVS and DRDIAGNOINTRACOL using DDRE
End Algorithm DRDIAGNOINTRACOL

La découverte des liens sémantiques (dépendances fonctionnelles et relations d'ordre) entre les colonnes est notre ultime étape dans l'établissement du nouveau schéma sémantique de la source de données. Cette étape est composée de plusieurs procédures, en se basant sur le rapport DRDIAGNOINTRACOL. Les procédures sont détaillées ci-dessous. La procédure DiscoverSEMANTICDFLinks permet de découvrir les premiers types de liens sémantiques appélés dépendances fonctionnelles. La procédure DiscoverSEMANTICLS-Links permet de découvrir les autres types de liens sémantiques appélés relations d'ordres ($<$, $=$, $>$).

Algorithm 9 Procedure DiscoverSEMANTICDFLinks

Require: DDCATLINKS, DRDIAGNOINTRACOL
Ensure: DRDIAGNOINTERCOLDF
Begin

for each Cat in DRDIAGNOINTRACOL.CATEGORYENG **do**

if $Cat \in DDCATLINKS.CATEGORYL$ **then**
Insert (CATEGORYL, CATEGORYR, DRDIAGNOINTERCOLDF)
end if
end for
//Compute-InterColMeasures
for each i in 1..NBRCOLDF **do**
for each k in 01...nm2 **do**
$M2k \leftarrow ComputeDDVS(M2k)$; Insert(M2k, DRDIAGNOINTERCOLDF)
end for
end for
End Procedure DiscoverSEMANTICDFLinks

Algorithm 10 Procedure DiscoverSEMANTICLSLinks

Require: DS

Ensure: DRDIAGNOINTERCOLLS

Begin

for each pair of columns having the same category **do**

//Compute-InterColMeasures

for each k in 01...nm2 **do**

$M2k \leftarrow ComputeDDVS(M2k)$; Insert(M2k, DRDIAGNOINTERCOLLS)

end for

end for

End Procedure DiscoverSEMANTICLSLinks

La procédure DIAGNOSTICDSINTERCOL ci-dessous, permet de faire l'union des deux étapes intermédiaires. Le deux rapports DRDIAGNOINTERCOLDF et DRDIAGNOINTERCOLLS permettent de construire le diagnostic de liens inter-colonnes.

Algorithm 11 Procedure DIAGNOSTICDSINTERCOL

// Established the inter-column data report

Begin

Union of the two intermediate reports DRDIAGNOINTERCOLDF and DRDIAGNOINTERCOLLS to create DRDIAGNOINTERCOL

End Algorithm DIAGNOSTICDSINTERCOL

La procédure DIAGNOSTICDS permet de calculer les mesures globales d'une source de données telles que le nombre total de valeurs dans la source de données, le nombre total de valeurs syntaxiquement erronnées, le nombre total de valeurs nulles.

Algorithm 12 Procedure DIAGNOSTICDS

```
// Calculate the different global-measures and ratios for the whole source
// Fill the data report DRDIAGNOGLOBAL
Begin
for each i in 1..NBRCOL do
   for each j in 01...nm0 do
      M0j ← Compute(M0j)
      Insert(M0j, DRDIAGNOGLOBAL)
   end for
end for
// Calculate the Ratios
for each l in 001...nr do
   Rl ← ComputeR(R1l)
   Insert(Rl, DRDIAGNOGLOBAL)
end for
End Algorithm DIAGNOSTICDS
```

L'application de notre processus de catégorisation sémantique sur une source de données sans schéma permet de reconnaitre le sens de certaines colonnes selon le tableau ci-dessous. Ainsi une redécouverte partielle du schéma des données est établie.

Transformation de schéma					
Ancien schéma		**Nouveau schéma sémantique (Sch1)**			
Colonne	**Type de don.**	**Catégorie**	**Sous-Catégorie**	**Type de données**	**DD**
COL1	String	INTEGER	UNKNOWN	Number	DDRE
COL2	String	FRMUTUAL	ABREVIATION	String	DDVS
COL3	String	UNKNOWN	UNKNOWN	String	UNKNOWN
COL4	String	CIVILITY	FRENCH	String	DDVS
COL5	String	GENDER	FRENCH	String	DDVS
COL6	String	DATEDDMMYYYY	UNKNOWN	Date	DDRE
COL7	String	CITY	FRENCH	String	DDVS
COL8	String	COUNTRY	FRENCH	String	DDVS
COL9	String	CONTINENT	FRENCH	String	DDVS
COL10	String	DATEDDMMYYYY	UNKNOWN	Date	DDRE

TABLE 3.11 – Nouveau schéma sémantique de DS (1)

Le processus de catégorisation sémantique sur une source de données sans schéma permet d'enrichir la description des colonnes avec des contraintes et des commentaires comme présenté dans le tableau ci-dessous.

Nouveau schéma sémantique		
Colonne	**Types de Contraintes**	**Commentaires**
COL1		
COL2		List of french mutual
COL3		
COL4	IF	List of civilities such as Mrs., Miss, Mr.
COL5	IF	List of gender (or sex) as F, M
COL6	TR	A date in the format dd-mm-yyyy
COL7	TR	City List
COL8	TR	List of countries
COL9	TR	List of continents
COL10	TR	A date in the format YYYY-MM-DD

TABLE 3.12 – Nouveau schéma sémantique de DS (2)

Nouveau schéma sémantique				
Colonne	**Contraintes**			
	ID contrainte	**Type contraintes**	**Condition**	**Action**
COL1				
COL2				
COL3				
COL4	0018	IF	Madame	Mme
COL4	0023	IF	Mlle	Mme
COL5	0006	IF	0	F
COL5	0010	IF	Femme	F
COL5	0014	IF	Homme	M
COL5	0014	IF	1	M
COL6	0027	TR		YYYY-MM-DD
COL7	0001	TR		INITCAP
COL8	0002	TR		UPPER
COL9	0003	TR		UPPER
COL10	0027	TR		YYYY-MM-DD

TABLE 3.13 – Nouveau schéma sémantique de DS (3)

Le processus de catégorisation sémantique sur une source de données sans schéma permet d'enrichir la description en établissant les liens sémantiques qui peuvent exister entre les colonnes à savoir les dépendances fonctionnelles et les relations d'ordre ($<$, $=$, $>$). Les tableaux ci-dessous en sont des exemples.

Nouveau schéma sémantique				
Liens sémantiques inter-colonnes : DF				
Colonne	COLLEFT	Functional dependency DF	Colonne	COLRIGHT
COL2	FRMUTUAL	DF	COL8	COUNTRY
COL7	CITY	DF	COL8	COUNTRY
COL7	CITY	DF	COL9	CONTINENT
COL8	COUNTRY	DF	COL9	CONTINENT
COL4	CIVILITY	DF	COL5	GENDER

TABLE 3.14 – Nouveau schéma sémantique de DS (3)

Nouveau schéma sémantique				
Liens sémantiques inter-colonnes : LS				
Colonne	COLLEFT	Semantic Link LS	Colonne	COLRIGHT
COL6	DATE	$\leq$	COL10	DATE

TABLE 3.15 – Nouveau schéma sémantique de DS (3)

3.4 Conclusion

L'étape de la catégorisation permet de mieux comprendre la structure et la sémantique d'une source de données. Elle cherche à construire des connaissances intra-colonne et inter-colonnes. D'une part et afin de mieux cibler les actions correctives intra-colonne, elle consiste à reconnaître la catégorie sémantique, la sous-catégorie, le type de données et les contraintes de chaque colonne. D'autre part, l'étape de la catégorisation permet d'étudier les dépendances fonctionnelles et les liens sémantiques qui peuvent exister entre les colonnes. Cette étape est basée sur les connaissances pré-stockées dans les dictionnaires de données.
La catégorisation sémantique des données est une étape primordiale pour détecter et corriger les anomalies dans les données.

Chapitre 4

Nettoyage de données guidé par les sémantiques intra et inter-colonnes

Sommaire

4.1 Introduction

Dans le chapitre 3, nous avons présenté notre méthode qui permet de reconnaître la sémantique de chaque colonne d'une source de données et d'inférer aussi les liens sémantiques qui peuvent exister entre ces colonnes. Nous utilisons des connaissances pré-stockées dans notre réferentiel (les différents dictionnaires).

L'étape de catégorisation sémantique des données a permis donc de déterminer, dans la mesure du possible, le sens de chaque colonne d'une source de données à savoir :

— La reconnaîssance du sens d'une colonne
— L'attribution du type syntaxique
— La détermination des contraintes syntaxiques et sémantiques
— L'enrichissement de la description par d'éventuels commentaires
— La restructuration éventuelle de plusieurs colonnes à partir d'une seule
— La déduction des liens sémantiques inter-colonnes.

Dans ce chapitre, nous présentons notre approche qui permet la restructuration éventuelle de la source de données ainsi que la détection et la correction de certaines anomalies. Concrètement, il s'agit de créer plusieurs colonnes à partir d'une seule ou de supprimer une colonne si elle est identique à une autre.
Notre but final est d'assister la détection et la correction des anomalies au niveau intra-colonne telles que les valeurs syntaxiquement incorrectes et les valeurs hétérogènes. Notre approche consiste aussi à aborder la violation des contraintes de dépendances et les traitements des certaines valeurs nulles [Zaidi et al. 2016a] [Zaidi et al. 2016b].
Dans un contexte de grosse volumétrie de bases de données (Big Data), nous utilisons la technologie MapReduce afin de vérifier les dépendances découvertes.

4.2 Processus de nettoyage de données

Le processus de nettoyage de données est composé par les étapes suivantes :

1. Diagnostiquer la source de données : la catégorisation sémantique des données.
2. Restructurer éventuellement des colonnes à partir d'une seule : modification de la structure d'une source de données par l'ajout ou la suppression de colonnes.
3. Refaire le diagnostic de la source de données : il s'agit de répéter l'étape de la catégorisation sémantique sur la nouvelle structure de la source.
4. Modifier à nouveau la structure de données.

5. Corriger des anomalies grâce aux dépendances fonctionnelles reconnues : traiter certaines valeurs nulles et les anomalies causées par la violation de dépendances fonctionnelles en se basant sur le dictionnaire de données (DDVSLINKS). Grâce à une source de dépendances fonctionnelles, il est possible de corriger les valeurs erronnées et les valeurs nulles dans une colonne dépendante. Il s'agit donc de corrections à la fois intra et inter-colonnes.
6. Détecter et corriger les anomalies intra-colonne : la correction syntaxique et la standardisation de données.
7. Détecter et corriger les anomalies inter-colonnes : vérification des dépendances fonctionnelles et correction des anomalies causées par la violation de ces dépendances.
8. Refaire le diagnostic de la source de données.
9. Élimination de doublons et similaires.

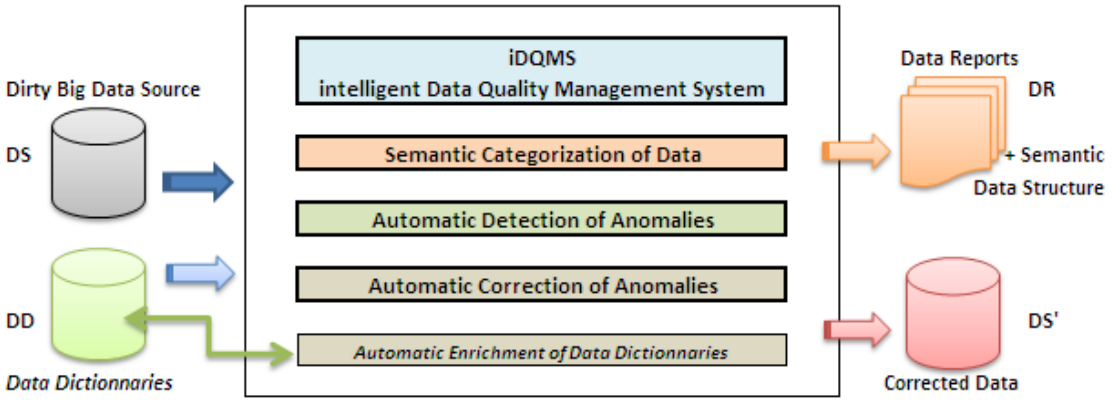


FIGURE 4.1 – Les étapes de processus de nettoyage de données

4.3 Modification de la structure d'une source de données

L'étape de la restructuration d'une source de données consiste à exploiter les connaissances inférées de la catégorisation sémantique. Il s'agit de proposer une nouvelle description de la source tout en se basant sur les mesures intra et inter-colonnes. Ces mesures permettent de mieux comprendre la structure de la source.

L'étape de la catégorisation des données peut renseigner le fait qu'une colonne puisse appartenir à plusieurs catégorie. Il est alors possible de transformer le schéma intial de données en plusieurs schémas selon le nombre de catégories par colonne.
Par exemple, pour une source DS(Col1, Col2, ..., Colj), si Col2 correspond à deux caté-

gories différentes soit Cat21 et Cat22, il est alors possible de définir de nouveaux schémas pour la source DS comme suit :

— DS(Col1, Col2, ..., Colj)
— DS(Col1, Cat21, ..., Colj)
— DS(Col1, Cat22, ..., Colj)
— DS(Col1, Cat21, Cat22 ..., Colj)

La restructuration de la source de données peut permettre la reconnaissance de la totalité ou d'une partie du contenu d'une colonne inconnue au départ.
Il existe deux types de modification de la source possibles à savoir l'ajout et la suppression des colonnes.
Le processus de nettoyage permet, d'une part, d'éclater une colonne en plusieurs colonnes si le nombre majoritaire de mots à considérer dans une colonne est supérieur à un ou si ses valeurs appartiennent à plusieurs catégories sémantiques. D'autre part, il est possible de recommander de supprimer une colonne s'il s'agit de deux colonnes identiques selon les mesures et les ratios réalisés (M201).

4.3.1 Ajout de colonnes

Les valeurs dans une colonne d'une source de données peuvent être issues de l'intégration de sources hétérogènes. Ces valeurs peuvent réprésenter plusieurs données. En effet, il est fréquent de combiner la civilité, le nom et le prénom dans une même colonne ou encore d'inverser le nom et le prénom par exemple. Le découpage en plusieurs colonnes permet de mieux détecter les anomalies et vérifier la cohérence avec d'autres colonnes. En effet, la notion de catégorie dominante, présentée la première fois dans [BenSalem, 2015], ne permet pas de convertir une colonne composée de plusieurs concepts, et par conséquent, il est impossible de la valider grâce au dictionnaire de données DDVS. L'orginalité de notre approche consiste donc à modifier la structure de la source de données traitée afin de palier ce type de problème et détecter même une inchoérence entre une colonne et une partie d'une colonne. Le tableau 4.1 permet d'expliciter ce cas de figure.

Exemple 4.1 : ajout de colonnes
L'ajout de colonnes peut se faire si le nombre majoritaire de mots à considérer au sein d'une même colonne (M120) est supérieur à 1. Le nombre majoritaire de mots (M120) dans cet exemple est égal à 3.

Ancien schéma		⇒	**Nouveau schéma**		
UNKNOWN	M120	⇒	**CIVILITY**	**FIRSTNAME**	**UNKNOWN**
Données		⇒	**Données**		
-			-	-	-
Mlle Anne MARTIN	3		Mlle	Anne	MARTIN
Mlle Karine LEBON	3		Mlle	Karine	LEBON
M Robert FORT	3		M	Robert	FORT
M Simon GENEREUX	3		M	Simon	GENEREUX
M Simon GENEREUX	3		M	Simon	GENEREUX
M Simon GENEREUX	3		M	Simon	GENEREUX
Mlle Katia BON	3		Mlle	Katia	BON
Mlle Houda ZAIDI	3		Mlle	Houda	ZAIDI
M Adem LE BON	4		M	Adem	LE BON
M Adem LE BON	4		M	Adem	LE BON
M Robert LEBON	3		M	Robert	LEBON
M Robert DUPONT	3		M	Robert	DUPONT

TABLE 4.1 – COL3 de la source de données DS

Les outils ETL sur le marché permettent l'intégration de données sans vérification de sémantique. Il est par conséquent fréquent de regrouper dans une même colonne des données sémantiquement incohérentes. L'éclatement d'une colonne en plusieurs pourrait être la solution pour palier la problématique de l'appartenance des données à plusieurs catégories dans une même colonne. Le tableau 4.3 ci-dessous permet d'illustrer ce cas.

Transformation de schéma					
Ancien schéma		Nouveau schéma sémantique (Sch2)			
Colonne	**Type de don.**	**Catégorie**	**Sous-Catégorie**	**Type de données**	**DD**
COL1	String	INTEGER		Number	DDRE
COL2	String	FRMUTUAL	ABREVIATION	String	DDVS
COL3$$1	String	CIVILITY	FRENCH	String	DDVS
COL3$$2	String	FIRSTNAME	FRENCH	String	DDVS
COL3$$3	String	UNKNOWN	UNKNOWN	String	UNKNOWN
COL4	String	CIVILITY	FRENCH	String	DDVS
COL5	String	GENDER	FRENCH	String	DDVS
COL6	String	DATE	DDMMYYYY	Date	DDRE
COL7	String	CITY	FRENCH	String	DDVS
COL8	String	COUNTRY	FRENCH	String	DDVS
COL9	String	CONTINENT	FRENCH	String	DDVS
COL10	String	DATE	DDMMYYYY	Date	DDRE

TABLE 4.2 – Nouveau schéma sémantique de DS (sch2)

Ancien schéma : schéma de départ de la source de données

Colonne	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
Type de données	String	String	String	String	String	String	String	String	String	String
Contraintes										
Commentaires										

Nouveau schéma (Sch1) : schéma issu de l'étape de diagnostic

Colonne	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
Catégorie	INTEGER	FRMUTUAL	UNKNOWN	CIVILITY	GENDER	DATE	CITY	COUNTRY	CONTINENT	DATE
Sous-Catégorie	UNKNOWN	ABREVIATION	UNKNOWN	FRENCH	FRENCH	UNKNOWN	FRENCH	FRENCH	FRENCH	UNKNOWN
Type de données	Number	String	String	String	String	Date	String	String	String	Date
Contraites				IF ...	IF ...	YYY-MM-DD	INITCAP	UPPER	UPPER	YYY-MM-DD
Commentaires				List of civ...	List of...	A date form.	List of ...	List of ...	List of ...	A date form.

Nouveau schéma (Sch2) : shéma issu de la restructuration

Colonne	COL1	COL2	COL3$$1	COL3$$2	COL3$$3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
Catégorie	INTEGER	FRMUTUAL	CIVILITY	FIRSTNAME	UNKNOWN	CIVILITY	GENDER	DATE	CITY	COUNTRY	CONTINENT	DATE
Sous-Catégorie	UNKNOWN	ABREVIATION	FRENCH	FRENCH	UNKNOWN	FRENCH	FRENCH	UNKNOWN	FRENCH	FRENCH	FRENCH	UNKNOWN
Type de données	Number	String	String	String	String	String	String	Date	String	String	String	Date
Contraites			IF ...	INITCAP		IF ...	IF ...	YYY-MM-DD	INITCAP	UPPER	UPPER	YYY-MM-DD
Commentaires			List of civ...	List of civ...		List of civ...	List of...	A date form.	List of ...	List of ...	List of ...	A date form.

FIGURE 4.2 – Nouveaux schémas sémantiques

Exemple 4.2 : ajout de colonnes

L'ajout de colonne peut être fait si le nombre de catégories par colonne (M108) est supérieur à 1.

Le nombre de catégories par colonne (M108) dans cet exemple est égal à 2.

Ancien schéma	⇒	**Nouveau schéma sémantique**	
EMAIL-FRPHONE	⇒	**EMAIL**	**FRPHONE**
Données	⇒	**Données**	**Données**
houda.z@cnam.fr		houda.z@cnam.fr	-
hz@yahoo.fr		hz@yahoo.fr	-
nn.pn@yahoo.fr		nn.pn@yahoo.fr	-
abc@cnam.fr		abc@cnam.fr	-
hhhh@gmail.com		hhhh@gmail.com	
003313007085012			003313007085012
00331305609010			00331305609010
00331805581011			00331805581011
003315003541010			003315003541010
00331900254401			00331900254401

TABLE 4.3 – Valeurs d'une colonne qui appartiennent à plusieurs catégories

4.3.2 Suppression de colonnes

Le schéma de la source de données peut ainsi être modifié. Les nouvelles colonnes peuvent être égales à des colonnes existantes. Il est possible donc d'étudier la cohérence globale et éventuellement supprimer les colonnes en doubles. L'étape de diagnostic de toute la source doit être relancée après modification de la structure de données.
L'indicateur du taux de remplissage d'une colonne (Mesure M101 réalisée lors du diagnostic) permet donc de garder ou de supprimer une nouvelle colonne.

Exemple 4.3 : suppression de colonnes : Relation sémantique (=) entre deux colonnes (M201) Après l'éclatement de la colonne COL3 en trois colonnes, nous avons constaté que la colonne COL4 est égale à COL3$$1, il est recommndé de supprimer une des deux. Le nouveau schéma sémantique (Sch3) de la source est présenté ci-dessous.

Ancien schéma : schéma de départ de la source de données

Colonne	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
Type de données	String	String	String	String	String	String	String	String	String	String
Contraintes										
Commentaires										

Nouveau schéma (Sch1) : schéma issu de l'étape de diagnostic

Colonne	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
Catégorie	INTEGER	FRMUTUAL	UNKNOWN	CIVILITY	GENDER	DATE	CITY	COUNTRY	CONTINENT	DATE
Sous-Catégorie	UNKNOWN	ABREVIATION	UNKNOWN	FRENCH	FRENCH	UNKNOWN	FRENCH	FRENCH	FRENCH	UNKNOWN
Type de données	Number	String	String	String	String	Date	String	String	String	Date
Contraites				IF ...	IF ...	YYY-MM-DD	INITCASE	UPPER	UPPER	YYY-MM-DD
Commentaires				List of civ...	List of...	A date form.	List of ...	List of ...	List of ...	A date form.

Nouveau schéma (Sch2) : shéma issu de la restructuration

Colonne	COL1	COL2	COL3$$1	COL3$$2	COL3$$3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
Catégorie	INTEGER	FRMUTUAL	CIVILITY	FIRSTNAME	UNKNOWN	CIVILITY	GENDER	DATE	CITY	COUNTRY	CONTINENT	DATE
Sous-Catégorie	UNKNOWN	ABREVIATION	FRENCH	FRENCH	UNKNOWN	FRENCH	FRENCH	UNKNOWN	FRENCH	FRENCH	FRENCH	UNKNOWN
Type de données	Number	String	String	String	String	String	String	Date	String	String	String	Date
Contraites			IF ...	INITCASE		IF ...	IF ...	YYY-MM-DD	INITCASE	UPPER	UPPER	YYY-MM-DD
Commentaires			List of civ...	List of civ...		List of civ...	List of...	A date form.	List of ...	List of ...	List of ...	A date form.

Nouveau schéma (Sch3) : shéma issu de la restructuration ---- Schéma final

Colonne	COL1	COL2	COL3$$2	COL3$$3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
Catégorie	INTEGER	FRMUTUAL	FIRSTNAME	UNKNOWN	CIVILITY	GENDER	DATE	CITY	COUNTRY	CONTINENT	DATE
Sous-Catégorie	UNKNOWN	ABREVIATION	FRENCH	UNKNOWN	FRENCH	FRENCH	UNKNOWN	FRENCH	FRENCH	FRENCH	UNKNOWN
Type de données	Number	String	String	String	String	String	Date	String	String	String	Date
Contraites			INITCASE		IF ...	IF ...	YYY-MM-DD	INITCASE	UPPER	UPPER	YYY-MM-DD
Commentaires			List of civ...		List of civ...	List of...	A date form.	List of ...	List of ...	List of ...	A date form.

FIGURE 4.3 – Nouveau schéma sémantique (Sch3)

4.4 Traitement des dépendances fonctionnelles

Nous proposons une solution qui permet de traiter les valeurs nulles et certaines anomalies causées par la violation de dépendances fonctionnelles. Nous exploitons les connaissances déduites à partir de l'étape de la catégorisation sémantique à savoir la liste de dépendances fonctionnelles inférées pour corriger ces anomalies.

Nous utilisons les valeurs valides stockées dans notre dictionnaire de données (DDVSLINKS). Il s'agit de faire une jointure entre la source de données et le dictionnaire DDVSLINKS et remplacer les valeurs de la partie gauche d'une dépendance fonctionnelle dans la source par les valeurs valides dans DDVSLINKS. Les détails du traitement des dépendances fonctionnelles sont donnés dans l'algorithme 13.
Les dépendances fonctionnelles détectées lors du diagnostic de la source sont :
$COL2 \rightarrow COL8$, $COL7 \rightarrow COL8$, $COL8 \rightarrow COL9$ et $COL7 \rightarrow COL9$.

Elles correspondent sémantiquement aux trois dépendances fonctionnelles préstockées dans le DDVSLINKS, à savoir $FRMUTUAL \rightarrow COUNTRY$, $CITY \rightarrow COUNTRY$, $COUNTRY \rightarrow CONTINENT$ et $CITY \rightarrow CONTINENT$.
On en déduit que plusieurs jointures doivent être réalisées afin de corriger certaines anomalies syntaxiques et déduire les valeurs exactes de certaines valeurs nulles.

Une ou plusieurs jointures de la source de données avec le dictionnaire DDVSLINKS doivent être établies afin de corriger une partie de valeurs dépendantes de la source de dépendance fonctionnelle. La première jointure permet, par exemple, de corriger les valeurs {Franc, Fr, Frence} en la valeur France, vu que la valeur France est associée à Paris. Toutes les valeurs nulles correspondantes à Paris seront remplacées par France.

Il faut remarquer que quand la valeur de la source de la dépendance fonctionnelle n'existe pas dans le dictionnaire DDVS ou qu'elle est erronée aucune correction dans la colonne dépendante n'est réalisée. Ainsi, les valeurs {Parisss, Pari, Bruxelle, Vill, Beijing} ne sont pas corrigées à ce stade du traitement des anomalies. Il est évident que les valeurs dépendantes de celles-ci ne le sont pas non plus.

Nous exploitons les informations déduites à partir des mesures et ratios calculés dans les étapes précedentes de processus de nettoyage pour proposer certain ordre de traitement de dépendances fonctionnelles. Par exemple, il est recommandé de commencer par la dépendance avec la partie gauche qui contient moins de valeurs nulles (R101) et moins de valeurs syntaxiquement incorrectes (R104) afin de corriger le maximum de valeurs dans les colonnes dépendantes.

Ancien schéma			⇒	**Nouveau schéma**		
Données erronées			⇒	**Données corrigées partiellement**		
COL7	**COL8**	**COL9**	⇒	**CITY**	**COUNTRY**	**CONTINENT**
Parisss	France	-		Parisss	France	Europe
Paris	Franc	-		Paris	France	Europe
Loiret	France	Europe		Loiret	France	Europe
Paris	Fr	Europe		Paris	France	Europe
Beijing	Chine	Asie		Beijing	Chine	Asie
-	China	Afrique		-	China	Asia
Pari	Frence	Europe		Pari	Frence	Europe
Bruxelle	France	-		Bruxelle	France	-
Paris	-	Eurape		Paris	France	Europe
Paris	-	Europe		Paris	France	Europe
Calvados	-	-		Calvados	France	Europe
Vill	Pai	Conti		Vill	Pai	Conti
Pékin	Chine	Asia		Pékin	Chine	Asie
Beijing	Chin	Asia		Beijing	Chin	Asia
Bruxelle	France	-		Bruxelle	France	Europe
Bruxelle	France	-		Bruxelle	France	Europe

TABLE 4.4 – Exemple de traitement des dépendances fonctionnelles

Algorithm 13 Procedure DFCorrection

Require: DS, DDVSLINKS, COLLEFT, COLRIGHT, CATEGORYL, CATEGORYL, SUBCATEGORY

Ensure: DS'

```
Begin
//Join DS, DDVSLINKS
// DS' : Source corrected
for each v_i in DS.COLLEFT do
  if CATEGORYL=DDVSLINKS.CATEGORYL AND CATEGORYR=DDVSLINKS.CATEGORYL then
    if SUBCATEGORY="ENGLISH" AND v_i =DDVSLINKS.ENGLISHL then
      DS' ← Insert(DS'.COLRIGHT, DDVSLINKS.ENGLISHR)
    else
      if SUBCATEGORY="FRENCH" AND v_i =DDVSLINKS.FRENCHL then
        DS' ← Insert(DS'.COLRIGHT, DDVSLINKS.FRENCHR)
      end if
    end if
  end if
end for
End Procedure DFCorrection
```

4.5 Détection et correction des anomalies intra-colonne

Dans ce paragraphe, nous présentons l'étape de la correction des anomalies intra-colonne. Notre approche permet d'exploiter les connaissances sémantiques déduites à partir de l'étape de la catégorisation sémantique pour détecter les anomalies au niveau intra-colonne.

Les rapports d'anomalies fournis par l'étape de la catégorisation sémantique montrent différents types d'erreurs sur les données à savoir :

- Les valeurs hétérogènes appartenant à plusieurs catégories : il s'agit de transformer les données en une seule sous-catégorie.
- Les valeurs hétérogènes avec différents formats de codage. Ces dernière peuvent être issues de différentes source hétérogènes. Il s'agit de la standardisation des données.
- Les valeurs syntaxiquement invalides. La correction syntaxique des données se fait en rapprochant les valeurs mal orthographiées avec des valeurs similaires dans le dictionnaire de données.

Plusieurs transformations sont déclenchées automatiquement grâce aux mesures et ra-

tios réalisées dans l'étape de la catégorisation sémantique de données. Nous présentons ci-dessous la plus importante d'entre elles.

4.5.1 Transformation en une seule sous-catégorie

L'étape de la catégorisation sémantique des données permet de reconnaître la sous-catégorie dominante dans la source de données. Cette information est déuite à partir de mesures globales (M011). Nous proposons d'unifier les données dans une même sous-catégorie. Il s'agit de traduire les valeurs qui n'appartiennent pas à la sous-catégorie dominante par leur synonyme dans la langue dominante.

La détermination de la sous-catégorie dominante de toute la source de données (M011) se fait en calculant pour chaque colonne Col_i et pour chaque sous-catégorie dans la colonne un ratio α_i :

α_i=$M109_i/M105_i$
M011=Max(Moyenne$(\alpha_i)_{ENGLISH}), Moyenne(\alpha_i)_{FRENCH}))$

Algorithm 14 Procedure TRANSFORMATION_SUBCAT
Require: DSPrim, DDVS, CATEGORY, COL
Ensure: DSPrim
//Transform the data in a column into the same SubCategory (M011)
Begin
for each v_i in COL **do**
if $v_i \in DDVS.CATEGORY$ AND $v_i \in DDVS.M011$ **then**
v_i :=DDVSTOT.M011
end if
end for
End Procedure TRANSFORMATION_SUBCAT

Example 4.3 : unfication des données en une même langue

Contraintes	Catégorie	Sous-Catégorie	Valeur	Valeur unifiée
INITCAP	CITY	FRENCH	Beijing	Pékin
UPPER	COUNTRY	FRENCH	China	Chine
UPPER	CONTINENT	FRENCH	Asia	Asie
INITCAP	CITY	FRENCH	Pai	Pai

TABLE 4.5 – Transformation dans la sous-catégorie dominante

4.5.2 Transformations selon les contraintes

Différents formats peuvent être utilisés dans une colonne. Nous proposons une unification de la codification des valeurs de deux manières : soit en appliquant le concept de dépendance fonctionnelle soit en appliquant les règles relatives aux contraintes stockées. L'algorithme 15 résume les transformations selon les contraintes.
Par exemple pour la catégorie CIVILITY, les contraintes stockées dans le dictionnaire DD-CATCONSTR permettent d'unifier la codification (Mme. et M.). Le tableau 4.6 permet de donner des exemples d'unification de la catégorie CIVILITY. Les valeurs qui ne sont pas dans le dictionnaire sont traitées avec les méthodes de calcul de distance de similarité.

Algorithm 15 Procedure TRANSFORMATION_CONSTR

Require: DSPrim, COL, Category

Ensure: DSPrim

```
//Transformations according to constraints
Begin
if Category ∈ DDCATCONSTR.CATEGORY then
  if TYPECONST="TR" then
    for each v_i in COL do
      switch FUNCTIONCONSTR1
      case INITCAP
      v_i := INITCAP(v_i)
      case UPPER
      v_i := UPPER(v_i)
      case 'YYYY-MM-DD'
      TO_DATE(COL, 'YYYY-MM-DD')
      case DIVISION
      v_i := v_i/ FUNCTIONCONSTR2
      case MULTIPLICATION
      v_i := v_i * FUNCTIONCONSTR2
      case FORMULE
      v_i := FUNCTIONCONSTR2
      end switch
    end for
  else
    if TYPECONST="IF" then
      for each v_i in COL do
        if v_i=FUNCTIONCONSTR1 then
          v_i := FUNCTIONCONSTR2
        end for
      end if
    end if
  end if
End Procedure TRANSFORMATION_CONSTR
```

Catégorie	Valeur	Valeur unifiée
CIVILITY	Miss	Mme
CIVILITY	Mrs	Mme
CIVILITY	Mademoiselle	Mme
CIVILITY	Monsieur	M.
CIVILITY	Mr	M.
CIVILITY	Madame	Mme
CIVILITY	Mademoisel	Mme
CIVILITY	Mademoisell	Mme
CIVILITY	Mensieur	M.

TABLE 4.6 – Unification des valeurs de la catégorie CIVILITY

L'unification des valeurs de la catégorie GENDER, par exemple, peut être faite de la même manière selon les contraintes stockées dans DDCATCONSTR. Cela permet de transformer les différentes valeurs correspondante à cette catégorie vers « M » ou « F ».
L'application du principe de dépendance fonctionnelle (GENDER dépend de CIVILITY) permet aussi de corriger toutes les valeurs de GENDER en fonction de CIVILITY : si CIVILITY=Mme. alors GENDER=F sinon GENDER=M.
Le tableau 4.7 permet d'expliciter différents cas de figure.

Exemple 4.5 : unification de la codification des valeurs

Catégorie	Valeur	Valeur unifiée
GENDER	Homme	M
GENDER	Male	M
GENDER	1	M
GENDER	Femme	F
GENDER	0	F
GENDER	Homm	M
GENDER	Femele	F

TABLE 4.7 – Unification des valeurs de la catégorie GENDER

Les contraintes stockées dans DDCATCONSTR permettent d'unifier les valeurs de certaines catégories par exemple les données qui appartiennent à la catgégorie COUNTRY ou CONTINENT doivent être en majuscule et la première lettre des valeurs de la catégorie CITY doit être en majuscule

Contraintes	Catégorie	Valeur	Valeur unifiée
UPPER	COUNTRY	France	FRANCE
UPPER	COUNTRY	Tunisie	TUNISIE
UPPER	CONTINENT	Asie	ASIE
INITCAP	FIRSTNAME	HOUDA	Houda

TABLE 4.8 – Transformation dans la sous-catégorie dominante

Les contraintes stockées dans DDCATCONSTR permettent d'unifier les valeurs de la catégorie DATE sous le format « YYYY-MM-DD » . Nous proposons donc de transformer toutes les colonnes qui appartiennent à la catégorie DATE sous cet format.

Example 4.4 : unification des données en un seul format de la date

Contraintes	Catégorie	Valeur	Valeur unifiée
YYYY-MM-DD	DATE	12/03/1971	1971-03-12
YYYY-MM-DD	DATE	30/02/2009	2009-02-30
YYYY-MM-DD	DATE	01/02/2000	2000-02-01
YYYY-MM-DD	DATE	30-01-2017	2017-01-30

TABLE 4.9 – Unification du format de la date

Les contraintes stockées dans DDCATCONSTR permettent aussi d'unifier les unités de mesure. Par exemple, nous proposons d'unifier les valeurs de la distance en mètre.

Contraintes	Catégorie	Valeur	Valeur unifiée
DIVISION	DISTANCECM	1000cm	10m
MULTIPLICATION	DISTANCEM	100m	100m
MULTIPLICATION	DISTANCEKM	1km	1000m
FORMULE	TEMPFAHRENHEIT	90°F	32°C
MULTIPLICATION	TEMPCELSISUS	32°C	32°C

TABLE 4.10 – Transformations qui nécessitent des calculs

4.5.3 Transformation syntaxique selon les algorithmes de similarités

La correction syntaxique est basée sur le rapprochement de valeurs de la source et les valeurs de dictionnaire DDVS. Nous faisons appel à certaines méthodes de mesures de distances de similarités qui existent dans la littérature telles que les mesures lexicographiques (Levenshtein, Edit-Distance, Jaro-Winkler, Q-Gram), ou les mesures phonétiques

(Soundex, Double Metaphone). Nous signalons que toutes ces méthodes ne prennent pas en considération la sémantique contextuelle des données à comparer.

Nous avons effectué plusieurs calculs de distance de similarités en combinant les méthodes Levenshtein, Edit-Distance (équivalente à Levenshtein), Jaro-Winkler et Soundex afin de dégager un seuil de similarité en rapport avec la sémantique. Les tableaux (Table 4.4, Table4.5, Table4.6, Table4.7) ci-dessous soulignent la difficulté de trouver non seulement un algorithme adéquoit mais aussi un seuil à partir duquel on peut considérer que deux chaînes de caractères sont similaires.

La requête SQL ci-dessous illustre la difficulté. En effet, la comparaison des catégories GENDER, DATE et EMAIL pose problème tel que mentionné dans la figure 4.4.

Une colonne dont les valeurs ne font pas partie d'un catégorie connue (Category= « UNKOWN »), dont le type est String et dont la longeur maximale ainsi que le nombre majoritaire de mots est important, peut nécessiter la combinaison de plusieurs algorithmes de distance de similarité (à savoir Soundex plus Jaro-Winkler) afin de comparer au mieux les chaînes de caractères.

```
SELECT categorieval Category, UPPER(valeur1) VALUP1, UPPER(valeur2) VALUP2,
UTL_MATCH.edit_distance(UPPER(valeur1), UPPER(valeur2)) AS ED,
UTL_MATCH.jaro_winkler(UPPER(valeur1), UPPER(valeur2)) AS JW,
UTL_MATCH.edit_distance_similarity(UPPER(valeur1), UPPER(valeur2)) AS EDS,
UTL_MATCH.jaro_winkler_similarity(UPPER(valeur1), UPPER(valeur2)) AS JWS,
UTL_MATCH.edit_distance_similarity(soundex(valeur1), soundex(valeur2)) AS SDX
FROM matchval
WHERE UTL_MATCH.edit_distance(UPPER(valeur1), UPPER(valeur2))<=2
ORDER BY Idval;
```

CATEGORY	VALUP1	VALUP2	ED	JW	EDS	JWS	SDX
GENDER	M	M	0	1E+000	100	100	100
GENDER	M	F	1	0	0	0	75
GENDER	M	1	1	0	0	0	0
DATE	11-11-2016	11-11-2016	0	1E+000	100	100	100
DATE	11-11-2016	11-11-2000	2	9,2E-001	80	92	100
DATE	11-11-2016	11-12-2016	1	9,378E-001	90	93	100
DATE	11-11-2016	12-11-2016	1	8,4E-001	90	84	100
DATE	11-11-2016	11-11-2017	1	9,6E-001	90	96	100
EMAIL	FB@LIPN.UNIV-PARIS13.FR	FB@LIPN.UNIV-PARIS13.FR	0	1E+000	100	100	100
EMAIL	FB@LIPN.UNIV-PARIS13.FR	YB@LIPN.UNIV-PARIS13.FR	1	9,71E-001	96	97	25

FIGURE 4.4 – Distances de similarités 1

La méthode Edit-Distance semble être la plus appropriée pour traduire la similarité en termes de nombre de caractères différents quelque soient leurs positions dans la chaîne. Nous proposons alors un seuil pour chaque catégorie, celui-ci est préstocké dans le dictionnaire

DDCAT. Le tableau (table 4.11) est un exemple. Une étude plus détaillée devrait être menée afin de d'automatiser tout ou partie de cette expertise avec des algorithmes d'apprentissage [Ref 2016].

Catégorie	Algorithme de similarités	Distance de similarités
CITY	Edit-distance	2
FIRSTNAME	Edit-distance	2
GENDER	Edit-distance	1
EMAIL	Edit-distance	0
NUMBER	Edit-distance	0
DATE	Edit-distance	1
URL	Edit-distance	0
UNKNOWN, M120>=ϵ_1, M110>=ϵ_2,	Soundex+Jaro-Winkler	10

TABLE 4.11 – Seuils de similarité selon la catégorie

CATEGORY	VALUP1	VALUP2	ED	JW	EDS	JWS	SDX
FIRSTNAME	ADAM	ADAM	0	1E+000	100	100	100
FIRSTNAME	ADAM	ADEM	1	8,667E-001	75	86	100
FIRSTNAME	ADAM	ADAMS	1	9,6E-001	80	96	75
FIRSTNAME	RAHMA	RAMA	1	9,467E-001	80	94	100
FIRSTNAME	MARIE-NOEL	MARIE NOEL	1	9,6E-001	90	96	100
FIRSTNAME	FRANC	FRANK	1	9,2E-001	80	92	100
FIRSTNAME	MBARAK	MOUBARAK	2	9,25E-001	75	92	100
FIRSTNAME	INÈS	INES	1	8,667E-001	75	86	100
FIRSTNAME	INÈS	INESS	2	8,267E-001	60	82	100
FIRSTNAME	INÈS	YNEÈS	2	7,833E-001	60	78	75
FIRSTNAME	INÈS	AGNÈS	2	7,833E-001	60	78	25
CITY	PARIS	PARISSS	2	9,429E-001	72	94	100
CITY	PARIS	PARI	1	9,6E-001	80	96	75
CITY	PÉKIN	BEIJING	5	5,619E-001	29	56	50
CITY	LONDRES	LONDRE	1	9,714E-001	86	97	100
CITY	LONDRES	LONDON	3	8,476E-001	58	84	75
GENDER	M	M	0	1E+000	100	100	100
GENDER	M	F	1	0	0	0	75
GENDER	M	1	1	0	0	0	0
GENDER	M	MALE	3	7,75E-001	25	77	75

FIGURE 4.5 – Distances de similarités 2 (catégories appartant au dictionnaire DDVS)

CATEGORY	VALUP1	VALUP2	ED	JW	EDS	JWS	SDX
DATE	10-11-2016	10-NOV-16	5	7,9E-001	50	79	0
DATE	10-11-2016	,JEUDI 10 NOVEMBRE 2016	17	4,29E-001	27	42	0
DATE	11-11-2016	11-11-2016	0	1E+000	100	100	100
DATE	11-11-2016	11-11-2000	2	9,2E-001	80	92	100
DATE	11-11-2016	11-12-2016	1	9,378E-001	90	93	100
DATE	11-11-2016	12-11-2016	1	8,4E-001	90	84	100
DATE	11-11-2016	11-11-2017	1	9,6E-001	90	96	100
EMAIL	FB@LIPN.UNIV-PARIS13.FR	FB@LIPN.UNIV-PARIS13.FR	0	1E+000	100	100	100
EMAIL	FB@LIPN.UNIV-PARIS13.FR	YB@LIPN.UNIV-PARIS13.FR	1	9,71E-001	96	97	25
EMAIL	FB@LIPN.UNIV-PARIS13.FR	FB@IUTV.UNIV-PARIS13.FR	4	9,158E-001	83	91	75
URL	WWW.UNIV-PARIS13.FR	WWW.UNIV-PARIS13.FR	0	1E+000	100	100	100
URL	WWW.UNIV-PARIS13.FR	WWW.UNIV-PARIS1.FR	1	9,895E-001	95	98	100
URL	WWW.UNIV-PARIS13.FR	WWW.UNIV-PARIS3.FR	1	9,895E-001	95	98	100
URL	WWW.UNIV-PARIS13.FR	WWW.UNIV-PARI13.FR	1	9,895E-001	95	98	100
URL	WWW.UNIV-PARIS13.FR	WWW.UNIVPARIS13.FR	1	9,895E-001	95	98	75
NUMBER	17091958	17091958	0	1E+000	100	100	100
NUMBER	17091958	27091958	1	9,167E-001	88	91	100
NUMBER	12345679	12345678	1	9,5E-001	88	95	100
NUMBER	0.000012345679	0.000012345678	1	9,714E-001	93	97	100

FIGURE 4.6 – Distances de similarités 3 (catégories appartant au dictionnaire DDRE)

CATEGORY	VALUP1	VALUP2	ED	JW	EDS	JWS	SDX
FIRSTLASTNAME	ADAM EVE	EVE ADAM	8	4,722E-001	0	47	25
FIRSTLASTNAME	PETER PARKER	PETE PARKER	1	9,288E-001	92	92	50
FIRSTLASTNAME	PETER PARKER	PETER PARKER	0	1E+000	100	100	100
FIRSTLASTNAME	CLARK KENT	CLAIRE KENT	2	9,083E-001	82	90	100
FIRSTLASTNAME	WONDER WOMAN	PONDER WOMAN	1	9,444E-001	92	94	75
FIRSTLASTNAME	SUPERMAN	SUPERMAN	0	1E+000	100	100	100
FIRSTLASTNAME	THE HULK	IRON MAN	8	4,167E-001	0	41	0
FIRSTLASTNAME	HARISSA FORD	HARISSON FORD	2	9,344E-001	85	93	75
FIRSTLASTNAME	BUS WILLY	BRUCE WILLY	3	8,848E-001	73	88	50
FIRSTLASTNAME	BRIGITTE BARDO	BRIGITTE FARDO	1	9,714E-001	93	97	100
FIRSTLASTNAME	HEDI MUFTI	EDDY MURFI	5	8E-001	50	80	50
FIRSTLASTNAME	ALAIN DE LOIN	ALAIN DELON	2	9,692E-001	85	96	100
ADDRESS	17 BOULEVARD FOCH EPINAY/SEINE	17 BOULEVARD FOCH 93800 EPINAY-SUR-SEINE	11	9,245E-001	73	92	100
ADDRESS	17 BOULEVARD FOCH EPINAY/SEINE	17 BLD FOCH 93800 EPINAY-SUR-SEINE	17	8,104E-001	50	81	50
ADDRESS	17 BOULEVARD FOCH EPINAY/SEINE	BOULEVARD FOCH 93800 EPINAY-SUR-SEINE	14	7,655E-001	63	76	100

FIGURE 4.7 – Distances de similarités 4 (catégories non reconnues)

Les corrections qui n'ont pas été faites par l'application des traitements de dépendances fonctionnelles sont alors gérées au moyen des algorithmes de distances de similarités. L'exemple ci-dessous en est une illustration.

Example 4.6 : correction syntaxique

Catégorie	Valeurs invalides syntaxiquement	Valeurs corrigées
CITY	Parisss	Paris
CITY	Pari	Paris
COUNTRY	Franc	France
CONTINENT	Eurape	Europe
FIRSTNAME	MARIE NOEL	MARIE-NOEL

TABLE 4.12 – Correction syntaxique des données

4.6 Détection et correction des anomalies inter-colonnes

L'étape de la catégorisation permet de déduire les dépendances plausibles. Ces connaissances permettent de guider l'utilisateur et de diminuer l'espace de recherche dans le processus de la découverte des contraintes de dépendances ; contrairement à certains travaux qui cherchent à vérifier les dépendances fonctionnelles entre toutes les colonnes [Novelli and Cicchetti 2001] [Diallo and Novelli 2010] [Simonenko and Novelli 2012a].
Par exemple, chercher à vérifier les dépendances $FIRSTNAME \rightarrow CITY$ ou même $Date \rightarrow COUNTRY$ n'a aucun sens.

4.6.1 La vérification des contraintes de dépendances

L'étape qui précède la correction inter-colonnes est la vérification des contraintes de dépendances explicitées lors de la reconstruction de schéma de la source.
Par exemple, dans le fichier Patient.csv, seules les dépendances sémantiques $Civility \rightarrow SEX$, $City \rightarrow COUNTRY$, $CITY \rightarrow CONTINENT$ et $COUNTRY \rightarrow CONTINENT$ ont été retrouvées.
Rappelons que DS(C) une source de données telle que C est l'ensemble des colonnes, X et Y deux sous-ensembles de colonnes tels que $X \subseteq C$ et $Y \subseteq C$, $X \bigcap Y = \emptyset$.
On dit que X détermine fonctionnellement Y ($X \rightarrow Y$) si et seulement si pour tout x_i=x_j alors y_i=y_j, $i \neq j$. En d'autre termes, pour toute valeur x_i de X, il n'existe qu'une seule valeur correspondante y_i de Y.

La vérification d'une dépendance fonctionnelle dans une source de données entre les colonnes X (COLLEFT) et Y (COLRIGHT) se fait selon l'algorithme ci-dessous.

Algorithm 16 Procedure VERIFYDEPENDANCE

Require: DS, COLLEFT, COLRIGHT

Begin
for each x_i in DS.COLLEFT **do**
$v_i \leftarrow compute\_COLRIGHT(y_i)$
end for
if $MAX(\mathrm{v}_i) \geq 2$ **then**
Write('Dependence is not verified') ;
else
Write('Dependence is verified') ;
end if
End Procedure VERIFYDEPENDANCE

Nous avons proposé une nouvelle version de l'algorithme de vérification d'une dépendance fonctionnelle en utilisant la technologie MapReduce. L'algorithme consiste à compter le nombre α_i de valeurs différentes de y_i pour chaque x_i. S'il existe un α_i supérieur à 1 alors la dépendance n'est pas vérifiée.
L'algorithme contient deux phases *Map* et deux phases *Reduce*. Les deux phases Map1 et Reduce1 prennent en entrée la source de données et elles renvoient le nombre d'occurrences (x_i ;y_i, α_i). Les deux phases Map2 et Reduce2 prennent en entrée le résultat de deux phases précédentes pour compter le nombre d'occurrences de chaque x_i, s'il est supérieur à 1 alors il s'agit de violation de contrainte de dépendance $X \rightarrow Y$.

Algorithm 17 VerificationDependenciesConstraints(MapReduce)

Map1(key,value)
Begin
for each each tuple in value **do**
EmitIntermediate("Xi ;Yij ", "1")// count occurences of Xi ;Yij
end for
End Map1
Reduce1(key,values)
// key : Xi ;Yij
//value : a list of counts
int $\alpha_i = 0$
file f
for each Xi ;Yij in values **do**
α_i += ParseInt(Xi ;Yij)
end for
Emit(α_i)
Write(f,α_i)
End Reduce1

Algorithm 18 VerificationDependenciesConstraints (MapReduce)

```
Map2(key,value)
// key : f
// value : f contents
for each word Xi in value do
   EmitIntermediate(Xi, "1") // count occurences of Xij
end for
End Map2
Reduce2(key, values )
//key : Xi
//values : a list of counts
for each v in values do
   result += ParseInt(Xi) // count occurences of Xi
end for
Emit(result)
if result=1 then
   validDF=1
end if
EndReduce2
End VerificationDependenciesConstraints
```

4.6.2 Correction des anomalies inter-colonnes

Notre approche permet de proposer des actions correctives au sein d'une même colonne et inter-colonnes. Concrètement, il s'agit, premièrement, de corriger les violations des contraintes de dépendance et certaines valeurs nulles. Ensuite, il s'agit d'homogénéiser les données au sein d'une même colonne (corrections syntaxiques et sémantiques pour unifier les formats et les sous-catégories). Enfin, nous proposons de revérifier les dépendances fonctionnelles et corriger les violations de ces contraintes.

Le processus de la correction des anomalies est composé par les étapes suivantes :

Ètape1 : correction des anomalies grâce aux dépendances fonctionnelles Dans cette étape, nous exploitons les connaissances stockées dans le DDVSLINKS afin de corriger des anomalies syntaxiques, sémantiques et certaines valeurs nulles. Les tableaux (Table 4.13, Table 4.14) ci-dessous présentent les corrections effectuées à ce niveau de traitement sur notre exemple.

-	CRI	-	-	M.	M	-	Parisss	France	-	-
-	AGRR	-	-	Mme	F	-	Paris	Fran	-	-
-	CRP	-	-	M.	F	1996-16-10	Loiret	France	Europe	10/12/2014
-	MGEN	Martin	-	-	M	03-avr-80	Paris	Fr	Europe	-
-	-	Anne	-	Mlle	1	12/03/1971	Beijing	Chine	Asie	-
-	OTC	Karine	-	Mlle	1	-	-	China	Afrique	-
-	IPECA	Robert	-	M	0	-	Pari	Frence	Europe	-
-	IPECA	Simon	-	M	0	-	Bruxelle	France	-	-
-	IPECA	Simon	-	M	-	16-10-1996	Paris	-	Eurape	01/02/2000
-	IPECA	Simon	-	M	-	10-16-1996	Paris	-	Europe	23/11/2015
-	-	Katia	-	Mademoisele	Femme	24/06/1983	Calvados	-	-	-
-	-	Houda	-	Mlle	F	30/02/2000	Vill	Pai	Conti	-
-	-	Adem	-	M.	0	-	Pékin	Chine	Asia	-
-	-	Adem	-	M.	0	-	Beijing	Chin	Asia	-
-	-	Robert	-	M.	0	-	Bruxelle	France	-	-
-	-	Robert	-	M.	0	-	Bruxelle	France	-	-

TABLE 4.13 – DS après la restructuration

-	-	-	-	M	M	-	Parisss	France	**Europe**	-
-	CRI	-	-	Mme	F	-	Paris	**France**	**Europe**	-
-	AGRR	-	-	M	**M**	1996-16-10	Loiret	France	Europe	10/12/2014
-	CRP	Martin	-	-	M	03-avr-80	Paris	**France**	Europe	-
-	MGEN	Anne	-	Mlle	**F**	12/03/1971	Beijing	Chine	**Asia**	-
-	-	Karine	-	Mlle	**F**	-	-	China	**Asia**	-
-	OTC	Robert	-	M	**M**	-	Pari	Frence	Europe	-
-	IPECA	Simon	-	M	**M**	-	Bruxelle	France	**Europe**	-
-	IPECA	Simon	-	M	**M**	16-10-1996	Paris	**France**	**Europe**	01/02/2000
-	IPECA	Simon	-	M	**M**	10-16-1996	Paris	**France**	Europe	23/11/2015
-	IPECA	Katia	-	Mademoisele	**F**	24/06/1983	Calvados	**France**	**Europe**	-
-	-	Houda	-	Mlle	F	30/02/2000	Vill	Pai	Conti	-
-	-	Adem	-	M	**M**	-	Pékin	Chine	**Asie**	-
-	-	Adem	-	M	**M**	-	Beijing	Chin	Asia	-
-	-	Robert	-	M	**M**	-	Bruxelle	France	**Europe**	-
-	-	Robert	-	M	**M**	-	Bruxelle	France	**Europe**	-

TABLE 4.14 – DS après la correction des anomalies grâce aux dépendances fonctionnelles

Le traitement grâce aux dépendances fonctionnelles permet la correction des anomalies d'une manière partielle. Il est nécessaire de compléter le processus de nettoyage par un traitement au niveau intra-colonne pour corriger des erreurs syntaxiques et sémantiques telles que Bruxelle, Pari, Beijing et Mademoisele.

Ètape1 : correction des anomalies intra-colonne

La correction au niveau intra-colonne permet de corriger plusieurs types d'anomalies

telles que les valeurs syntaxiquement invalides, les valeurs hétérogènes. Plusieurs valeurs de la source de la dépendance fonctionnelle peuvent être donc corrigées. Par conséquent, l'application de traitement grâce aux dépendances une deuxième fois est nécessaire pour corriger certaines anomalies qui ne sont pas traitées à la première étape. L'étape la correction inter-colonnes facilite le traitement inter-colonnes.

-	-	-	-	**M.**	M	-	**Paris**	**FRANCE**	**EUROPE**	-
-	CRI	-	-	Mme	F	-	Paris	**FRANCE**	**EUROPE**	-
-	AGRR	-	-	**M.**	M	**1996-10-16**	Loiret	**FRANCE**	**EUROPE**	**2014-12-10**
-	CRP	Martin	-	-	M	03-avr-80	Paris	**FRANCE**	**EUROPE**	-
-	MGEN	Anne	-	**Mme**	F	**1971-03-12**	**Pékin**	**CHINE**	**ASIE**	-
-	-	Karine	-	**Mme**	F	-	-	**CHINE**	**ASIE**	-
-	OTC	Robert	-	**M.**	M	-	**Paris**	**FRANCE**	**EUROPE**	-
-	IPECA	Simon	-	**M.**	M	-	**Bruxelles**	**FRANCE**	**EUROPE**	-
-	IPECA	Simon	-	**M.**	M	**1996-10-16**	Paris	**FRANCE**	**EUROPE**	**2000-02-01**
-	IPECA	Simon	-	**M.**	M	10-16-1996	Paris	**FRANCE**	**EUROPE**	**2015-11-23**
-	IPECA	Katia	-	**Mme**	F	**1983-06-24**	Calvados	**FRANCE**	**EUROPE**	-
-	-	Houda	-	**Mme**	F	30/02/2000	Vill	Pai	Conti	-
-	-	Adem	-	M.	M	-	Pékin	**CHINE**	**ASIE**	-
-	-	Adem	-	M.	M	-	**Pékin**	**CHINE**	**ASIE**	-
-	-	Robert	-	M.	M	-	**Bruxelles**	**FRANCE**	**EUROPE**	-
-	-	Robert	-	M.	M	-	**Bruxelles**	**FRANCE**	**EUROPE**	-

TABLE 4.15 – DS après la correction des anomalies intra-colonne

Ètape3 : correction des anomalies inter-colonnes

L'étape de la correction des anomalies inter-colonnes consiste à corriger les valeurs qui ne vérifient pas les dépendances fonctionnelles. Nous appliquons à nouveau la procédure de traitement des valeurs nulles et certaines dépendances fonctionnelles, si une dépendance fonctionnelle donnée n'est pas vérifiée. Nous utilisons les valeurs valides stockées dans notre dictionnaire de données (DDVSLINKS).
Notre solution consiste à vérifier l'ensemble de dépendances fonctionnelles inférées suite à la catégorisation sémantique de données. Ensuite, il s'agit de remplacer les valeurs qui ne vérifient pas les dépendances par les valeurs valides préstockées dans DDVSLIKNS.

Algorithm 19 Procedure inter-ColumnsCorrection

Require: DS

Ensure: DS'

Begin

//verification of dependencies constraints

// correction of inter-columns anomalies

// DS' : Source corrected

VerificationDependenciesConstraints(COLLEFT,COLRIGHT) ;

DFCorrection(DS, DDVSLINKS, COLLEFT, COLRIGHT, CATEGORYL, CATEGORYL, SUBCATEGORY) ;

End Procedure inter-ColumnsCorrection

-	-	-	-	M.	M	-	Paris	FRANCE	EUROPE	-
-	CRI	-	-	Mme	F	-	Paris	FRANCE	EUROPE	-
-	AGRR	-	-	M.	M	1996-10-16	Loiret	FRANCE	EUROPE	2014-12-10
-	CRP	Martin	-	-	M	03-avr-80	Paris	FRANCE	EUROPE	-
-	MGEN	Anne	-	Mme	F	1971-03-12	Pékin	CHINE	ASIE	-
-	-	Karine	-	Mme	F	-	-	CHINE	ASIE	-
-	OTC	Robert	-	M.	M	-	Paris	FRANCE	EUROPE	-
-	IPECA	Simon	-	M.	M	-	Bruxelles	**BELGIQUE**	EUROPE	-
-	IPECA	Simon	-	M.	M	1996-10-16	Paris	FRANCE	EUROPE	2000-02-01
-	IPECA	Simon	-	M.	M	10-16-1996	Paris	FRANCE	EUROPE	2015-11-23
-	IPECA	Katia	-	Mme	F	1983-06-24	Calvados	FRANCE	EUROPE	-
-	-	Houda	-	Mme	F	30/02/2000	Vill	Pai	Conti	-
-	-	Adem	-	M.	M	-	Pékin	CHINE	ASIE	-
-	-	Adem	-	M.	M	-	Pékin	CHINE	ASIE	-
-	-	Robert	-	M.	M	-	Bruxelles	**BELGIQUE**	EUROPE	-
-	-	Robert	-	M.	M	-	Bruxelles	**BELGIQUE**	EUROPE	-

TABLE 4.16 – DS après la correction des anomalies inter-colonnes

Nous signalons qu'il existe encore d'autre types d'anomalies, d'une part, l'incohérence dans la ligne 5 entre la valeur de mutuelle « MGEN » et la valeur de pays « CHINE ». Ces anomalies peuvent être causées par la violation de dépendances fonctionnelle ou il est nécessaire d'enrichir d'avantage la sémantique de données. D'autre part, la source de données contient encore des valeurs invalides telles que Vill, Pai et Conti qui n'existent pas dans le dictionnaire DDVSLINKS. Ces anomalies peuvent être traitées par des algorithmes d'apprentissage.

4.7 Bilan

Notre approche permet la reconnaissance sémantique des structures de données, l'établissement de plusieurs rapports d'anomalies et le nettoyage de données.
La reconnaissance sémantique des structures de données consiste à déterminer le sens de chaque colonne d'une source de données. Nous utilisons la sémantique existante dans les dictionnaires de données préétablis afin de reconnaître le type de données, la catégorie sémantique et la sous-catégorie (telle que la langue utilisée), les contraintes et les commentaires de chaque colonne. Nous recommandons aussi les algorithmes de distance de similarité ainsi que le seuil à utiliser lors des comparaisons.

Cette étape permet d'inférer aussi les liens sémantiques qui peuvent exister entre les colonnes d'une source de données.
L'objectif de notre approche est d'établir plusieurs rapports qui portent sur les anomalies détectées.
Notre but final est le nettoyage de données. Il s'agit d'exploiter les connaissances sémantiques déduites à partir de l'étape précédente pour corriger les anomalies au niveau intra et inter-colonnes et traiter certaines valeurs nulles. La première grande étape de nettoyage consiste à corriger les anomalies causées par la violation des contraintes de dépendances. La deuxième grande étape permet l'homogénéisation des données. Nous proposons des corrections de valeurs syntaxiquement et sémantiquement incorrectes (unification et standardisation des données en un seul format et une seule langue) au sein d'une même colonne. L'orginalité de notre approche est que nous utilisons les données pour construire un nouveau schéma sémantique. Ce schéma contient le domaine sémantique (la catégorie et la sous-catégorie telle que la langue) et syntaxique (le type de données) de chaque colonne d'une manière explicite. Ce schéma sémantique facilite la détection des colonnes similaires et l'intégration de données dans les bases et les entrepôts de données.

Nous avons constaté, après l'application de notre processus de nettoyage, qu'il est possible de détecter des incohérences sur les données. Ceci peut être dû au choix de l'ordre dans le traitement de dépendances fonctionnelles. Une étude approfondie, sur la priorité des traitements de dépendances fonctionnelles à mener, devrait donc être faite. Nous pensons par ailleurs, que le traitement des doublons et des similaires (inter-lignes) devrait être fait après avoir abordé les corrections intra et inter-colonnes.

L'enrichissement automatique des différents dictionnaires constitue une prochaine étape nécessaire afin d'améliorer notre processus de nettoyage.

Dans la mesure où le processus de catégorisation ne permet pas de reconnaitre le sens d'une colonne avec les dictionnaires DDVS et DDRE, un traitement spécial est lancé pour tenter de découvrir la sémantique manquante.
Le dictionnaire DDKW est la base de ce traitement par mots clés.
Par exemple, une colonne Col_i dans la source de données peut réprésenter une adresse si la majorité des valeurs dans Col_i contiennent des mots clés tels que {Street, Avenue, Boulvard, Rue, Place, Square, Quai, Pont}.

Une étude plus détaillée devra être menée, afin de permettre la découverte de la sémantique d'une colonne, dans le cas où les différents dictionnaires ne le permettent pas. Des algorithmes d'apprentissage pourraient aider à la découverte du sens de la donnée.

4.8 Conclusion

Nous avons débuté le processus de nettoyage de données par le traitement de dépendances fonctionnelles en se basant, d'une part, sur les noms sémantiques de colonnes inférées à partir de l'étape de la catégorisation et, d'autre part, sur les dépendances fonctionnelles préstokées dans DDVSLINKS. Ce traitement permet de corriger certaines valeurs nulles et de valeurs qui ne vérifient pas les contraintes de dépendances.
La détection et la correction des anomalies intra-colonne consiste à standardiser les données homogènes et unifier les données en un format de codage unique et dans une seule langue. Nous avons utilisé les méthodes de calcul de distance de similarités pour corriger les valeurs syntaxiquement incorrectes et qui sont similaires à des valeurs qui existent dans le dictionnaire de données.

L'objectif de nettoyage de données au niveau inter-colonnes est de corriger les anomalies causées par la violation des contraintes de dépendances.

Chapitre 5

Expérimentation

Sommaire

5.1 Introduction

L'objectif de notre travail est donc d'assister l'utilisateur dans le processus de détection et de correction de certaines anomalies qui peuvent exister dans une source de données (Data Source : DS). Rappelons que les données rassemblées peuvent être issues de l'intégration de différentes sources avec différents niveaux de discriptions des métadonnées. Ces dernières peuvent être totalement absentes et le plus souvent non suffisantes pour réfleter le contenu réel des données et donc de traiter les éventuelles anomalies.

Notre but final est de contribuer au développement de nouveaux outils ETL qui n'imposent pas à l'utilisateur la connaissance des structures et des sémantiques des données manipulées. Ces outils devront permettre de guider les utilisateurs dans les processus de détection et de correction des anomalies. Les données rassemblées dans les bases et les entrepôts devraient avoir ainsi plus de crédibilité.

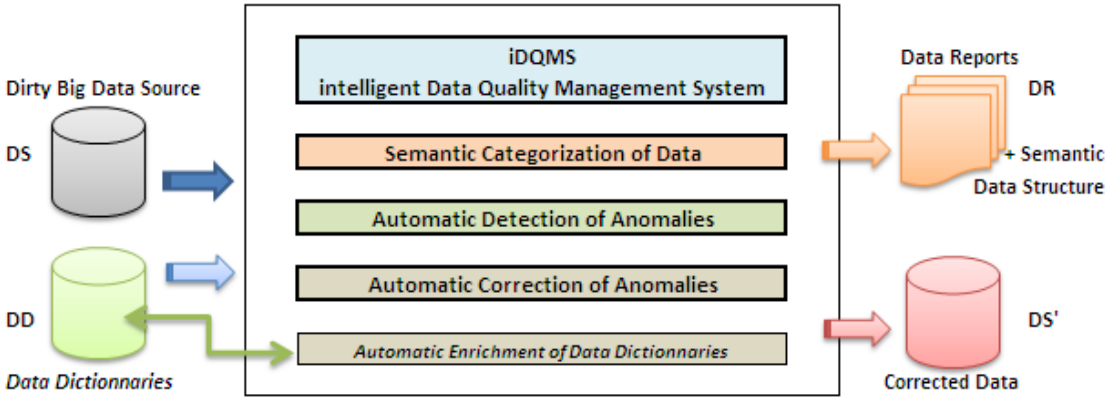


FIGURE 5.1 – Les étapes de processus de nettoyage de données

Afin de mettre en oeuvre notre approche, nous développons un outil (figure 5.1) appelé iDQMS (intelligent Data Quality Management System), celui-ci permet :

— La reconnaissance sémantique des structures de données grâce à un ensemble de dictionnaires de données (Data Dictionnaries : DD). C'est l'ensemble des donnés préstockés afin d'aider à la correction. Il s'agit de chaînes de caractères valides catégorisées, d'expressions régulières catégorisées aussi et enfin de liens sémantiques entre ces données. La structure de ces méta-données est réprésentée dans le chapitre 3.

— L'établissement de plusieurs rapports (Data Reports : DR) qui portent sur les ano-

malies détectées.

— Le nettoyage de données.

Dans le présent chapitre, nous allons présenter l'expérimentation faite sur les différents algorithmes. Les tests ont été réalisés en PL/SQL sous oracle (voir Annexe). Dans notre expérimentation l'accent est mis sur l'utilisation de la technologie Big Data MapReduce afin d'améliorer les perfermances des algorithmes très coûteux de détection et de correction de dépendances fonctionnelles.

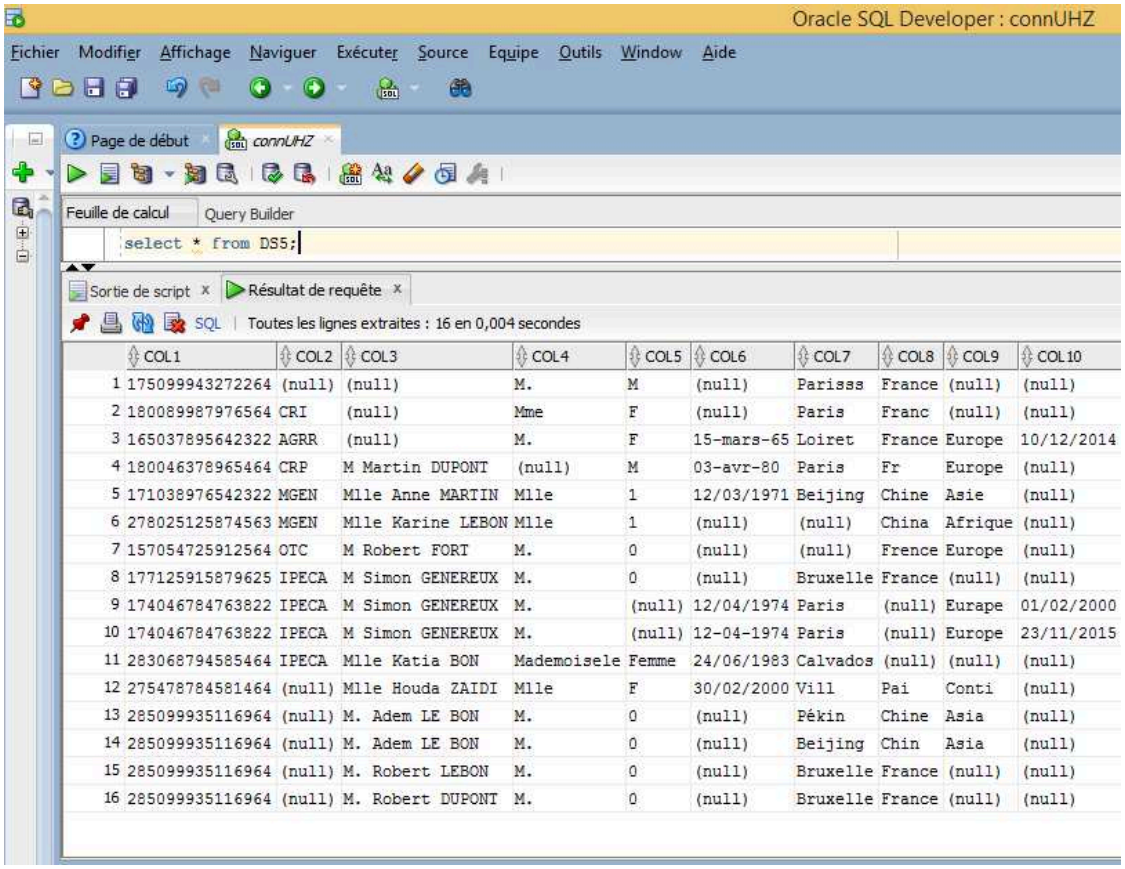
5.2 Catégorisation sémantique de données

L'étape de la catégorisation permet de mieux comprendre la structure et la sémantique d'une source de données. Elle consiste à reconnaître la catégorie sémantique, la sous-catégorie, le type de données et les contraintes de chaque colonne, ainsi que les dépendances fonctionnelles et les liens sémantiques qui peuvent exister entre les colonnes. Cette étape est basée sur les connaissances pré-stockées dans les dictionnaires de données. La catégorisation sémantique des données est une étape primordiale pour détecter et corriger les anomalies dans les données.

L'outil iDQMS prend en entrée la source de données DS. L'exemple traité dans toute la thèse est donné dans la figure (5.2). DS est dupliquée en créant DSPrim. Les corrections seront ainsi faites sur la copie.

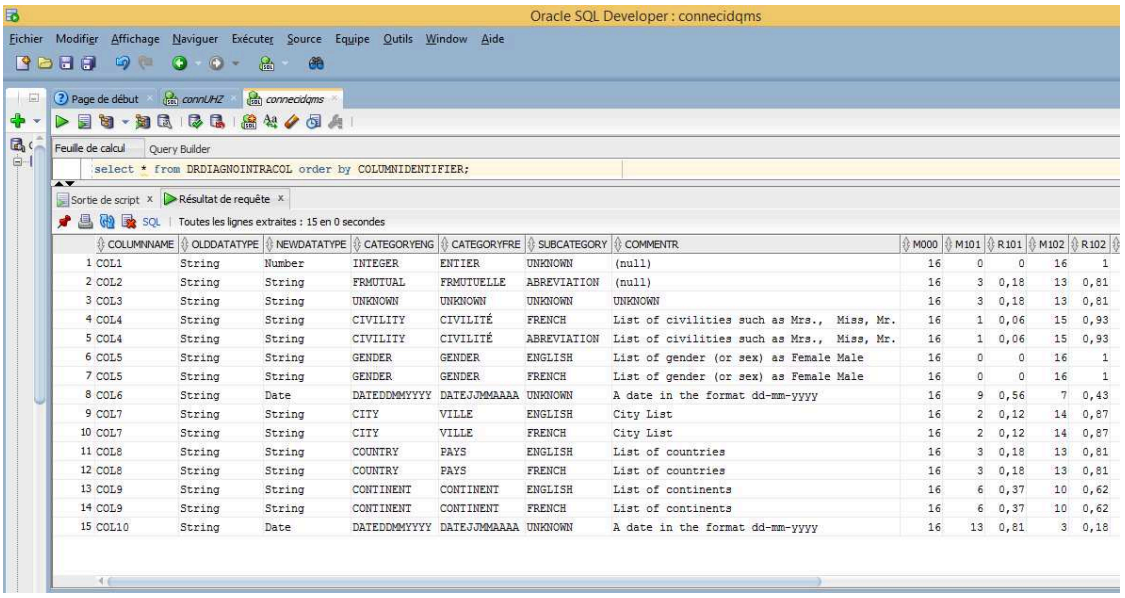
Un ensemble de diagnostics est lancé automatiquement. Ces derniers sont intra et inter-colonnes. L'objectif est d'analyser le sens de chaque colonne et ensuite de découvrir les liens éventuels entre les colonnes. La structure de la source est totalement remise en cause, une nouvelle architecture, plus riche sémantiquement est à redécouvrir. Le but final étant d'assister l'utilisateur dans sa démarche corrective et d'améliorer la qualité des données.

Les premiers diagnostics sont basés sur les dictionnaires DDCAT, DDVS, DDVSTOT et DDRE afin de construire le rapport de nom DRDIAGNOINTRACOL. Les figures (figure 5.3, figure 5.4) ci-dessous résument les résultats obtenus à savoir que l'on obtient les nouveaux types de données, les sémantiques de colonnes, les commentaires, les mesures et les ratios qui donnent une idée sur le contenu de la source.



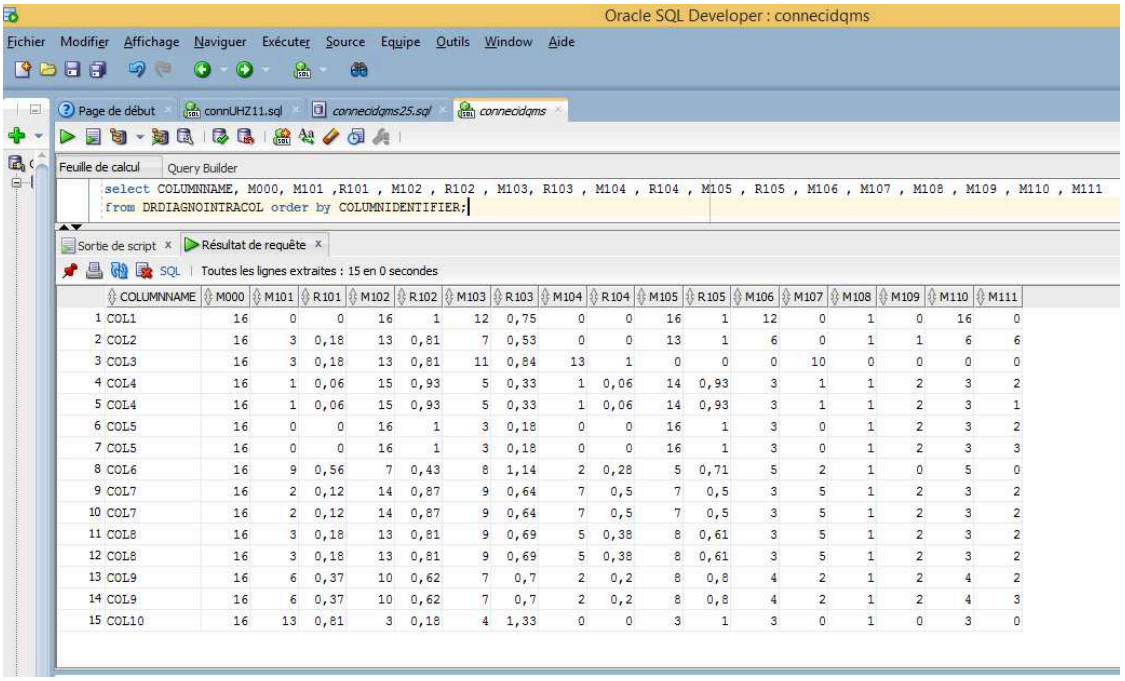
	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
1	175099943272264	(null)	(null)	M.	M	(null)	Parisss	France	(null)	(null)
2	180089987976564	CRI	(null)	Mme	F	(null)	Paris	Franc	(null)	(null)
3	165037895642322	AGRR	(null)	M.	F	15-mars-65	Loiret	France	Europe	10/12/2014
4	180046378965464	CRP	M Martin DUPONT	(null)	M	03-avr-80	Paris	Fr	Europe	(null)
5	171038976542322	MGEN	Mlle Anne MARTIN	Mlle	1	12/03/1971	Beijing	Chine	Asie	(null)
6	278025125874563	MGEN	Mlle Karine LEBON	Mlle	1	(null)	(null)	China	Afrique	(null)
7	157054725912564	OTC	M Robert FORT	M.	0	(null)	(null)	Frence	Europe	(null)
8	177125915879625	IPECA	M Simon GENEREUX	M.	0	(null)	Bruxelle	France	(null)	(null)
9	174046784763822	IPECA	M Simon GENEREUX	M.	(null)	12/04/1974	Paris	(null)	Eurape	01/02/2000
10	174046784763822	IPECA	M Simon GENEREUX	M.	(null)	12-04-1974	Paris	(null)	Europe	23/11/2015
11	283068794585464	IPECA	Mlle Katia BON	Mademoisele	Femme	24/06/1983	Calvados	(null)	(null)	(null)
12	275478784581464	(null)	Mlle Houda ZAIDI	Mlle	F	30/02/2000	Vill	Pai	Conti	(null)
13	285099935116964	(null)	M. Adem LE BON	M.	0	(null)	Pékin	Chine	Asia	(null)
14	285099935116964	(null)	M. Adem LE BON	M.	0	(null)	Beijing	Chin	Asia	(null)
15	285099935116964	(null)	M. Robert LEBON	M.	0	(null)	Bruxelle	France	(null)	(null)
16	285099935116964	(null)	M. Robert DUPONT	M.	0	(null)	Bruxelle	France	(null)	(null)

FIGURE 5.2 – DS : source de données qui contient des anomalies



	COLUMNNAME	OLDDATATYPE	NEWDATATYPE	CATEGORYENG	CATEGORYFRE	SUBCATEGORY	COMMENTR	M000	M101	R101	M102	R102
1	COL1	String	Number	INTEGER	ENTIER	UNKNOWN	(null)	16	0	0	16	1
2	COL2	String	String	FRMUTUAL	FRMUTUELLE	ABREVIATION	(null)	16	3	0,18	13	0,81
3	COL3	String	String	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	16	3	0,18	13	0,81
4	COL4	String	String	CIVILITY	CIVILITÉ	FRENCH	List of civilities such as Mrs., Miss, Mr.	16	1	0,06	15	0,93
5	COL4	String	String	CIVILITY	CIVILITÉ	ABREVIATION	List of civilities such as Mrs., Miss, Mr.	16	1	0,06	15	0,93
6	COL5	String	String	GENDER	GENDER	ENGLISH	List of gender (or sex) as Female Male	16	0	0	16	1
7	COL5	String	String	GENDER	GENDER	FRENCH	List of gender (or sex) as Female Male	16	0	0	16	1
8	COL6	String	Date	DATEDDMMYYYY	DATEJJMMAAAA	UNKNOWN	A date in the format dd-mm-yyyy	16	9	0,56	7	0,43
9	COL7	String	String	CITY	VILLE	ENGLISH	City List	16	2	0,12	14	0,87
10	COL7	String	String	CITY	VILLE	FRENCH	City List	16	2	0,12	14	0,87
11	COL8	String	String	COUNTRY	PAYS	ENGLISH	List of countries	16	3	0,18	13	0,81
12	COL8	String	String	COUNTRY	PAYS	FRENCH	List of countries	16	3	0,18	13	0,81
13	COL9	String	String	CONTINENT	CONTINENT	ENGLISH	List of continents	16	6	0,37	10	0,62
14	COL9	String	String	CONTINENT	CONTINENT	FRENCH	List of continents	16	6	0,37	10	0,62
15	COL10	String	Date	DATEDDMMYYYY	DATEJJMMAAAA	UNKNOWN	A date in the format dd-mm-yyyy	16	13	0,81	3	0,18

FIGURE 5.3 – DRDIAGNOINTRACOL : résultat de la catégorisation sémantique des données [Catégorie, Type, Commentaire]

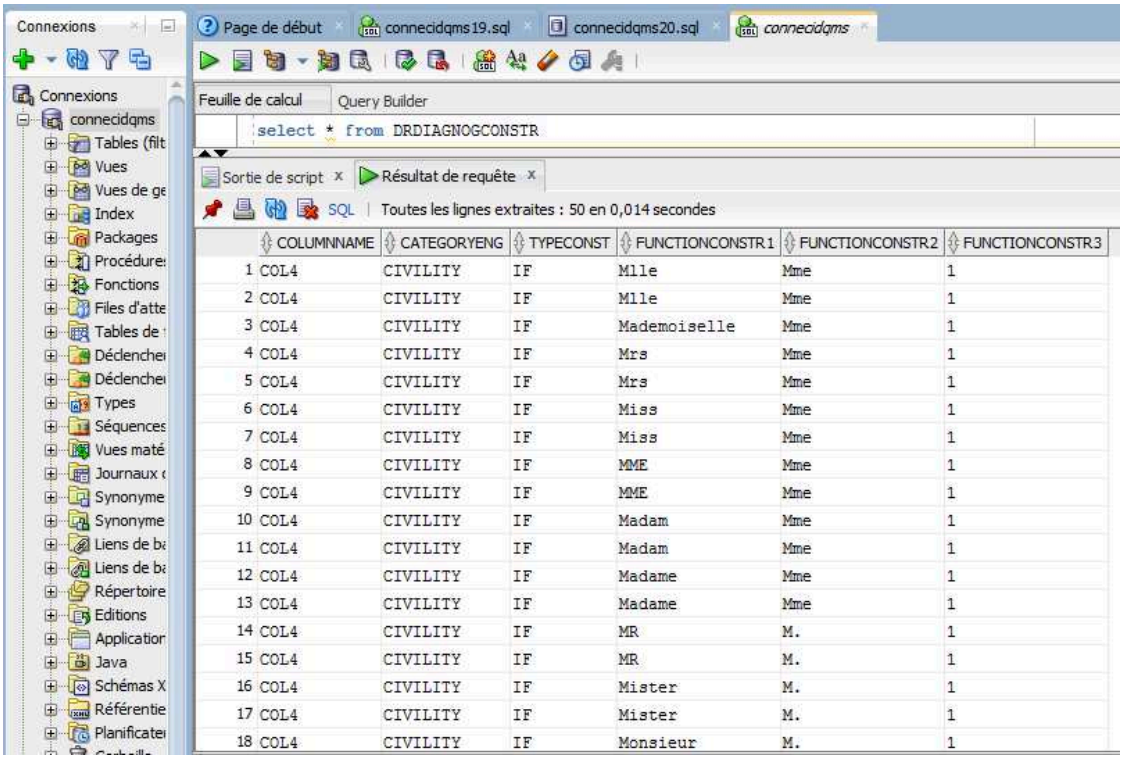


	COLUMNNAME	M000	M101	R101	M102	R102	M103	R103	M104	R104	M105	R105	M106	M107	M108	M109	M110	M111
1	COL1	16	0	0	16	1	12	0,75	0	0	16	1	12	0	1	0	16	0
2	COL2	16	3	0,18	13	0,81	7	0,53	0	0	13	1	6	0	1	1	6	6
3	COL3	16	3	0,18	13	0,81	11	0,84	13	1	0	0	0	10	0	0	0	0
4	COL4	16	1	0,06	15	0,93	5	0,33	1	0,06	14	0,93	3	1	1	2	3	2
5	COL4	16	1	0,06	15	0,93	5	0,33	1	0,06	14	0,93	3	1	1	2	3	1
6	COL5	16	0	0	16	1	3	0,18	0	0	16	1	3	0	1	2	3	2
7	COL5	16	0	0	16	1	3	0,18	0	0	16	1	3	0	1	2	3	3
8	COL6	16	9	0,56	7	0,43	8	1,14	2	0,28	5	0,71	5	2	1	0	5	0
9	COL7	16	2	0,12	14	0,87	9	0,64	7	0,5	7	0,5	3	5	1	2	3	2
10	COL7	16	2	0,12	14	0,87	9	0,64	7	0,5	7	0,5	3	5	1	2	3	2
11	COL8	16	3	0,18	13	0,81	9	0,69	5	0,38	8	0,61	3	5	1	2	3	2
12	COL8	16	3	0,18	13	0,81	9	0,69	5	0,38	8	0,61	3	5	1	2	3	2
13	COL9	16	6	0,37	10	0,62	7	0,7	2	0,2	8	0,8	4	2	1	2	4	2
14	COL9	16	6	0,37	10	0,62	7	0,7	2	0,2	8	0,8	4	2	1	2	4	3
15	COL10	16	13	0,81	3	0,18	4	1,33	0	0	3	1	3	0	1	0	3	0

FIGURE 5.4 – DRDIAGNOINTRACOL : résultat de la catégorisation sémantique des données (suite) [Mesure, Ratio]

L'étape de la catégorisation permet aussi d'expliciter les contraintes sur les données

selon les catégories inférées. Ces résultats sont stockés dans le rapport DRDIAGNOG-CONSTR. Des recommandations sur les algorithmes de similarité et les seuils selon la catégorie sont donnés dans le rapport DRDIAGNOALGOSIM. Des exemples de ces rapports sont donnés dans les figures (figure 5.6, figure 5.7).



	COLUMNNAME	CATEGORYENG	TYPECONST	FUNCTIONCONSTR1	FUNCTIONCONSTR2	FUNCTIONCONSTR3
1	COL4	CIVILITY	IF	Mlle	Mme	1
2	COL4	CIVILITY	IF	Mlle	Mme	1
3	COL4	CIVILITY	IF	Mademoiselle	Mme	1
4	COL4	CIVILITY	IF	Mrs	Mme	1
5	COL4	CIVILITY	IF	Mrs	Mme	1
6	COL4	CIVILITY	IF	Miss	Mme	1
7	COL4	CIVILITY	IF	Miss	Mme	1
8	COL4	CIVILITY	IF	MME	Mme	1
9	COL4	CIVILITY	IF	MME	Mme	1
10	COL4	CIVILITY	IF	Madam	Mme	1
11	COL4	CIVILITY	IF	Madam	Mme	1
12	COL4	CIVILITY	IF	Madame	Mme	1
13	COL4	CIVILITY	IF	Madame	Mme	1
14	COL4	CIVILITY	IF	MR	M.	1
15	COL4	CIVILITY	IF	MR	M.	1
16	COL4	CIVILITY	IF	Mister	M.	1
17	COL4	CIVILITY	IF	Mister	M.	1
18	COL4	CIVILITY	IF	Monsieur	M.	1

FIGURE 5.5 – DRDIAGNOGCONSTR : résultat de la catégorisation sémantique des données (suite) [Contraintes]

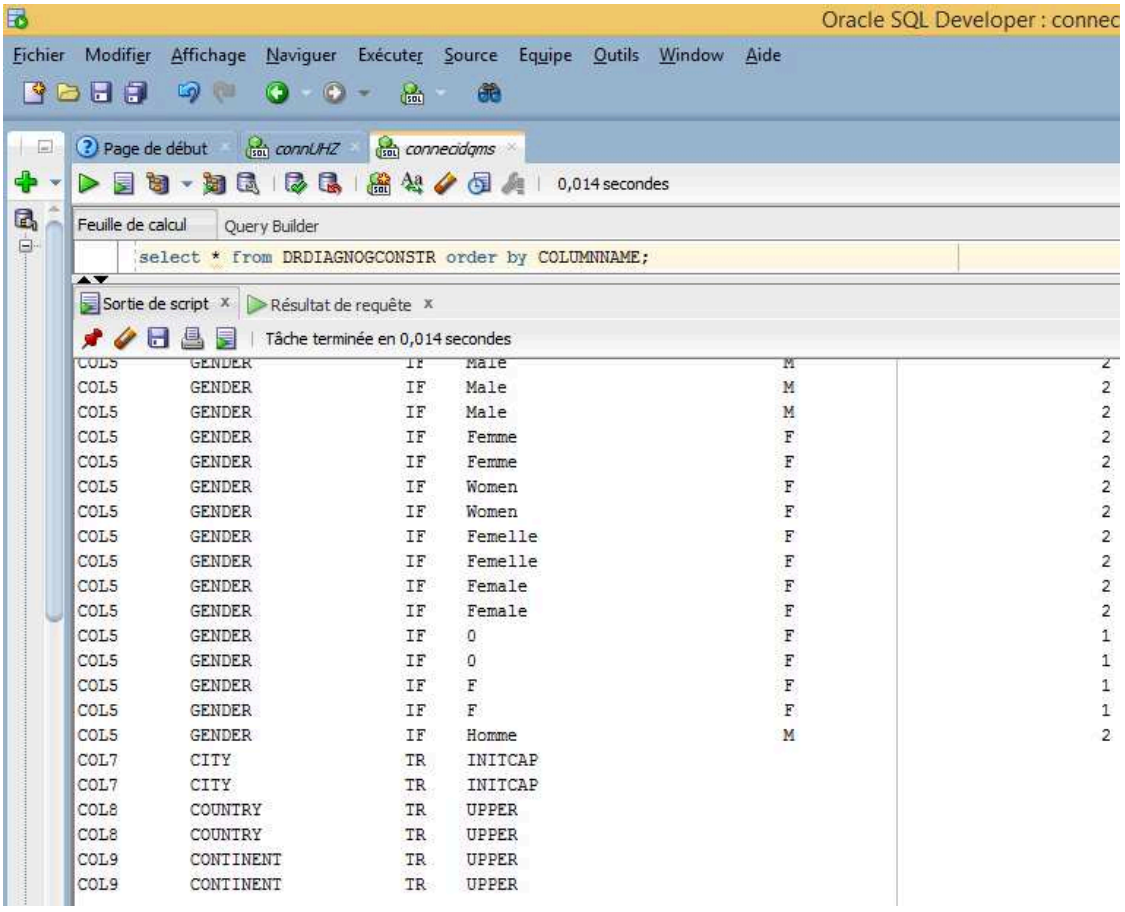
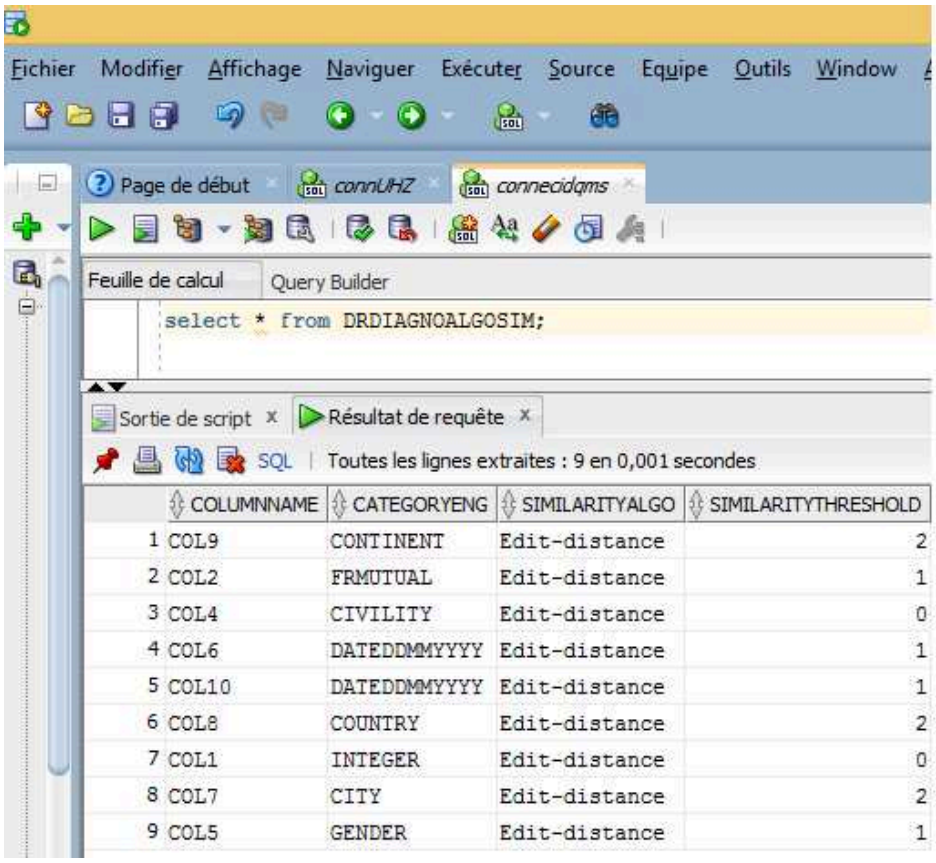


FIGURE 5.6 – DRDIAGNOGCONSTR : résultat de la catégorisation sémantique des données (suite) [Contraintes]



	COLUMNNAME	CATEGORYENG	SIMILARITYALGO	SIMILARITYTHRESHOLD
1	COL9	CONTINENT	Edit-distance	2
2	COL2	FRMUTUAL	Edit-distance	1
3	COL4	CIVILITY	Edit-distance	0
4	COL6	DATEDDMMYYYY	Edit-distance	1
5	COL10	DATEDDMMYYYY	Edit-distance	1
6	COL8	COUNTRY	Edit-distance	2
7	COL1	INTEGER	Edit-distance	0
8	COL7	CITY	Edit-distance	2
9	COL5	GENDER	Edit-distance	1

FIGURE 5.7 – DRDIAGNOALGOSIM : résultat de la catégorisation sémantique des données (suite) [Algorithmes de similarité, Seuils]

Les analyses intra-colonne permettent ainsi d'enrichir sémantiquement la description de la source. La tâche de l'étape suivante est de mettre l'accent sur les liens sémantiques inter-colonnes. En effet, les diagnostics vont porter sur la découverte des relations inter-colonnes afin de dresser les rapports de noms DRDIAGNOINTERCOLDF (figure 5.8) et DRDIAGNOINTERCOLLS (figure 5.9).

Le processus de catégorisation sémantique sur une source de données sans schéma permet d'enrichir la description en établissant les liens sémantiques qui peuvent exister entre les colonnes à savoir les dépendances fonctionnelles et les relations d'ordre ($<$, $=$, $>$). L'ensemble des diagnostics inter-colonnes est basée sur le dictionnaire DDCATLINKS.

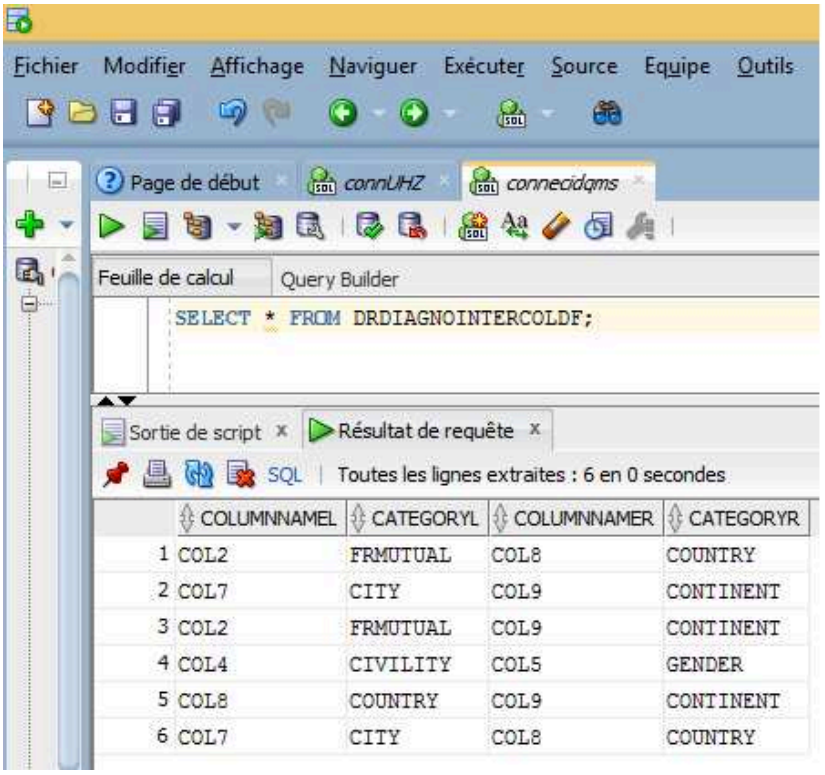


FIGURE 5.8 – DRDIAGNODF : résultat de la catégorisation sémantique des données (suite) [Dépendance fonctionnelle]

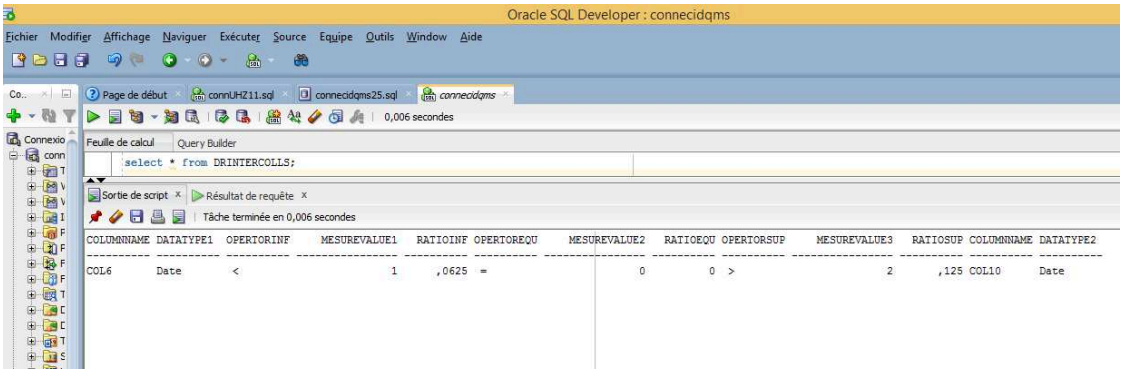


FIGURE 5.9 – DRDIAGNOINTERCOLLS : résultat de la catégorisation sémantique des données (suite) [Relation d'ordre $<$, $=$, $>$]

Des informations globales sur la source de données sont établies dans le rapport DRDIAGNOGLOBAL. Plusieurs mesures sont calculées afin de donner une idée sur le contenu de la source de données telles que le taux de remplissage des colonnes, les validités syntaxiques et sémantiques, la sous-catégorie (la langue) dominante de la source de données.

La figure (5.10) permet de récapituler.

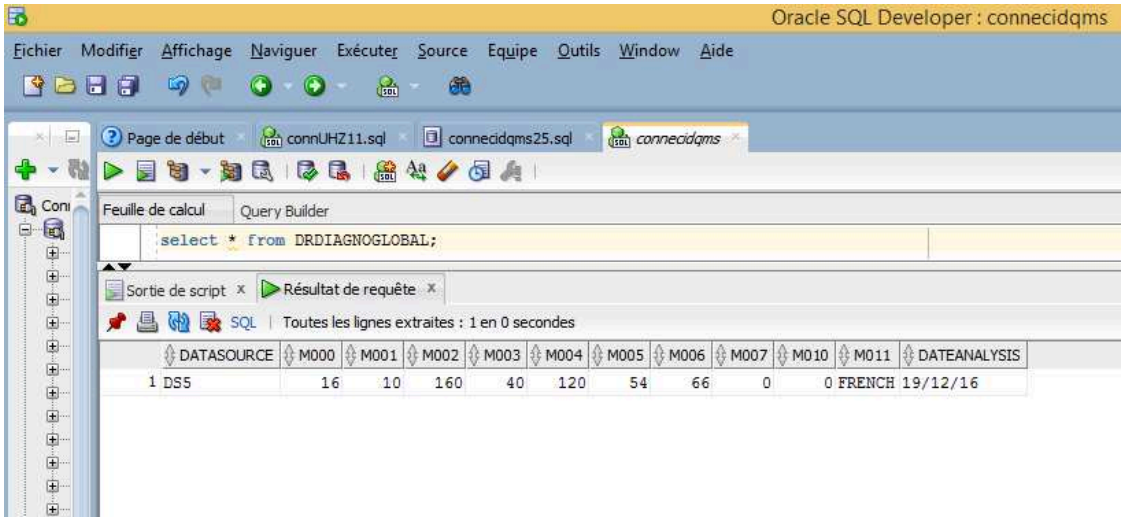
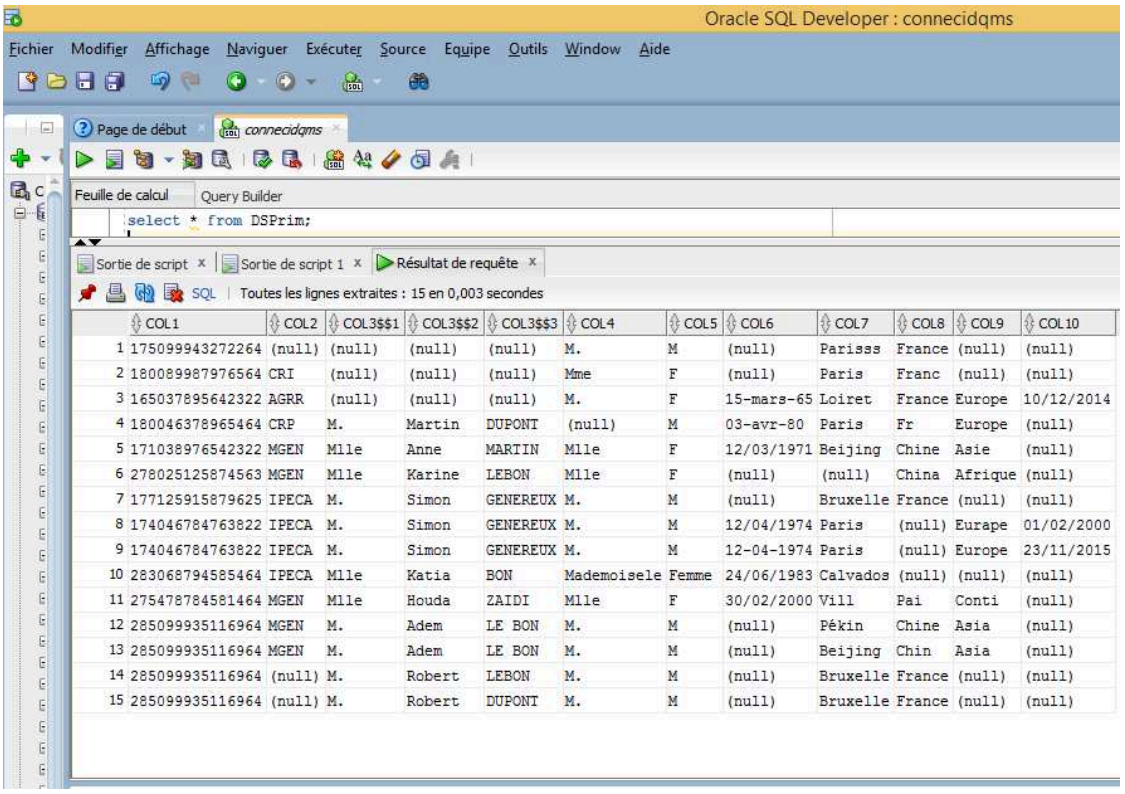


FIGURE 5.10 – DRDIAGNOGLOBAL : résultat de la catégorisation sémantique des données

Les codes source des procédures concernées sont données en annexes et sur le lien **http ://www.lipn.univ-paris13.fr/∼ boufares/Zaidi**

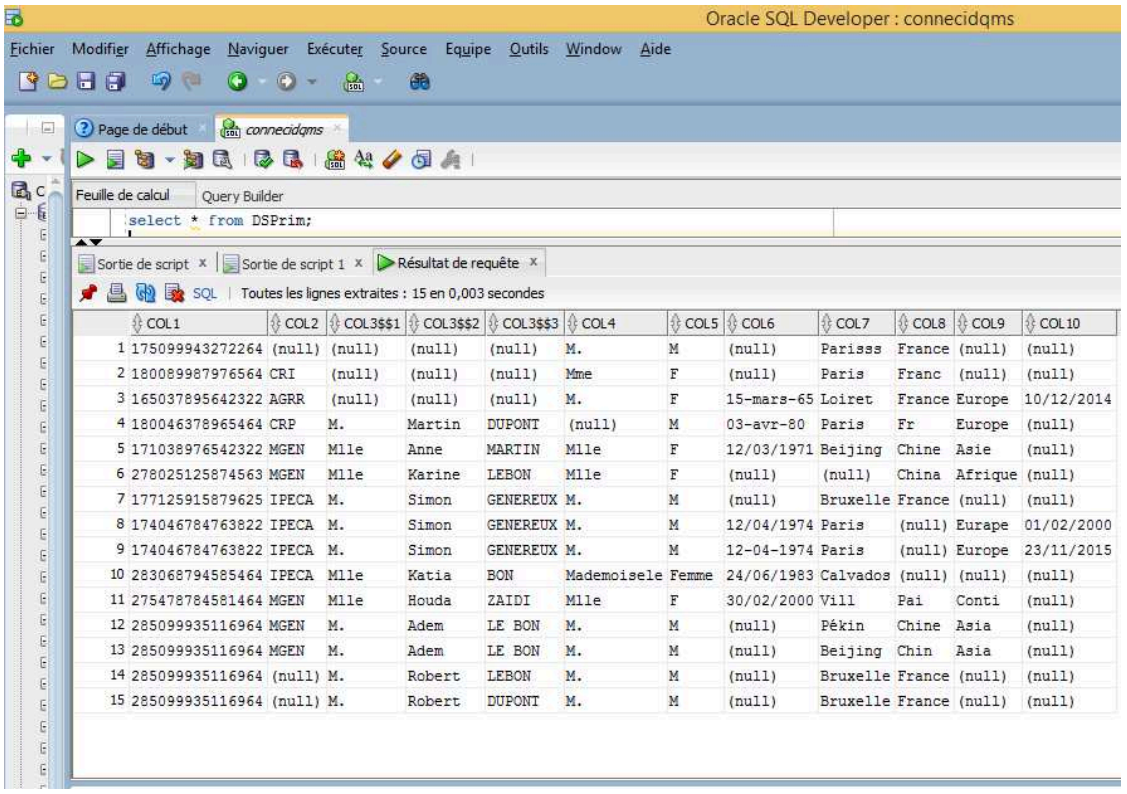
Les mesures intra-colonne permettent d'avoir une idée sur les données contenues dans la colonne telles que (i) les taux de valeurs nulles, (ii) les taux de valeurs syntaxiquement et sémantiquement valides, (iii) le nombre de valeurs distinctes et (iv) les types de données. Toutes ces mesures permettent la catégorisation automatique des données.
D'autres indicateurs permettent de déduire certaines actions qui portent sur la restructuration du schéma de la source de données à savoir le nombre de mots à considérer dans une même colonne ainsi que le nombre de catégories par colonne. On propose alors un nouveau schéma.

La nouvelle source de données, après modification de la structure (ajout de deux colonnes et suppression d'une colonne), est donnée selon les figures suivantes (5.11, 5.12). Une nouvelle étape de diagnostic est réalisée. Le résultat de ces diagnostics permet d'enrichir la description de la source avec les contraites, les commentaires et les algorithmes (Edit-Distance, Jaro-Winkler) à utiliser lors des comparaisons ainsi que les seuils recommandés pour chaque colonne.



	COL1	COL2	COL3$$1	COL3$$2	COL3$$3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
1	175099943272264	(null)	(null)	(null)	(null)	M.	M	(null)	Parisss	France	(null)	(null)
2	180089987976564	CRI	(null)	(null)	(null)	Mme	F	(null)	Paris	Franc	(null)	(null)
3	165037895642322	AGRR	(null)	(null)	(null)	M.	F	15-mars-65	Loiret	France	Europe	10/12/2014
4	180046378965464	CRP	M.	Martin	DUPONT	(null)	M	03-avr-80	Paris	Fr	Europe	(null)
5	171038976542322	MGEN	Mlle	Anne	MARTIN	Mlle	F	12/03/1971	Beijing	Chine	Asie	(null)
6	278025125874563	MGEN	Mlle	Karine	LEBON	Mlle	F	(null)	(null)	China	Afrique	(null)
7	177125915879625	IPECA	M.	Simon	GENEREUX	M.	M	(null)	Bruxelle	France	(null)	(null)
8	174046784763822	IPECA	M.	Simon	GENEREUX	M.	M	12/04/1974	Paris	(null)	Eurape	01/02/2000
9	174046784763822	IPECA	M.	Simon	GENEREUX	M.	M	12-04-1974	Paris	(null)	Europe	23/11/2015
10	283068794585464	IPECA	Mlle	Katia	BON	Mademoisele	Femme	24/06/1983	Calvados	(null)	(null)	(null)
11	275478784581464	MGEN	Mlle	Houda	ZAIDI	Mlle	F	30/02/2000	Vill	Pai	Conti	(null)
12	285099935116964	MGEN	M.	Adem	LE BON	M.	M	(null)	Pékin	Chine	Asia	(null)
13	285099935116964	MGEN	M.	Adem	LE BON	M.	M	(null)	Beijing	Chin	Asia	(null)
14	285099935116964	(null)	M.	Robert	LEBON	M.	M	(null)	Bruxelle	France	(null)	(null)
15	285099935116964	(null)	M.	Robert	DUPONT	M.	M	(null)	Bruxelle	France	(null)	(null)

FIGURE 5.11 – La nouvelle structure de la source



	COL1	COL2	COL3$$1	COL3$$2	COL3$$3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
1	175099943272264	(null)	(null)	(null)	(null)	M.	M	(null)	Parisss	France	(null)	(null)
2	180089987976564	CRI	(null)	(null)	(null)	Mme	F	(null)	Paris	Franc	(null)	(null)
3	165037895642322	AGRR	(null)	(null)	(null)	M.	F	15-mars-65	Loiret	France	Europe	10/12/2014
4	180046378965464	CRP	M.	Martin	DUPONT	(null)	M	03-avr-80	Paris	Fr	Europe	(null)
5	171038976542322	MGEN	Mlle	Anne	MARTIN	Mlle	F	12/03/1971	Beijing	Chine	Asie	(null)
6	278025125874563	MGEN	Mlle	Karine	LEBON	Mlle	F	(null)	(null)	China	Afrique	(null)
7	177125915879625	IPECA	M.	Simon	GENEREUX	M.	M	(null)	Bruxelle	France	(null)	(null)
8	174046784763822	IPECA	M.	Simon	GENEREUX	M.	M	12/04/1974	Paris	(null)	Eurape	01/02/2000
9	174046784763822	IPECA	M.	Simon	GENEREUX	M.	M	12-04-1974	Paris	(null)	Europe	23/11/2015
10	283068794585464	IPECA	Mlle	Katia	BON	Mademoisele	Femme	24/06/1983	Calvados	(null)	(null)	(null)
11	275478784581464	MGEN	Mlle	Houda	ZAIDI	Mlle	F	30/02/2000	Vill	Pai	Conti	(null)
12	285099935116964	MGEN	M.	Adem	LE BON	M.	M	(null)	Pékin	Chine	Asia	(null)
13	285099935116964	MGEN	M.	Adem	LE BON	M.	M	(null)	Beijing	Chin	Asia	(null)
14	285099935116964	(null)	M.	Robert	LEBON	M.	M	(null)	Bruxelle	France	(null)	(null)
15	285099935116964	(null)	M.	Robert	DUPONT	M.	M	(null)	Bruxelle	France	(null)	(null)

FIGURE 5.12 – La nouvelle structure de la source

5.3 Correction automatique des données

Deux grands types de corrections sont réalisés. Dans un premier temps, le contenu d'une colonne peut être corrigé grâce à une dépendance fonctionnelle détectée. Donc le lien inter-colonnes peut permettre des corrections intra-colonne. Dans un deuxième temps, un ensemble de transformations intra-colonne peut être déclenché, selon les indicateurs, afin de standardiser le contenu (voir figure 5.13).

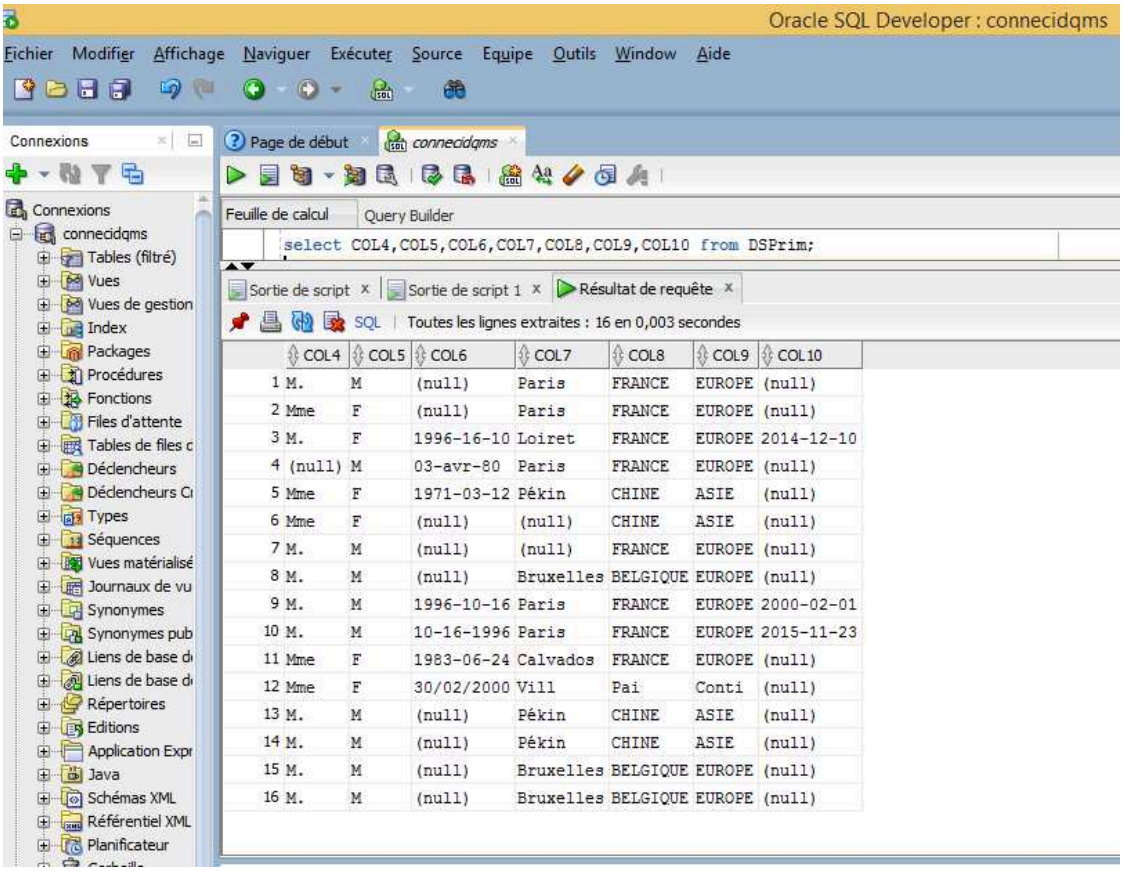


FIGURE 5.13 – Corrections intra et inetr-colonnes

5.4 Générateur de source de données volumineuse

Nous avons développé un générateur de source de données de gros volume (GenBD) afin d'étudier les perfermances des algorithmes.
GenBD permet de créer N lignes (N varie de 1 jusqu'à plusieurs centaines de millions voire plusieurs milliards selon la taille de disque disponible). Il est possible d'insérer des données qui respectent la règle de dépendance fonctionnelle tout comme le non respect Il permet de contrôler le taux de vérification de dépendances fonctionnelle. Des lignes erronnées sont insérées dans des emplacements bien précis de la source (voir lien **http ://www.lipn.univ-paris13.fr/∼ boufares/Zaidi**).

5.5 Vérification de la dépendance fonctionnelle

La vérification d'une DF dans une source de données entre les colonnes X (COLLEFT) et Y (COLRIGHT) se fait selon l'algorithme (le script SQL ci-dessous figure). Rappelons que la DF est vérifiée si pour chaque valeur de X (x=x', x et x' appartiennent à deux lignes différentes), il existe une seule valeur de Y (y=y', y et y' appartiennent à deux lignes différentes).

la première version de l'algorithme de vérification de la DF entre des colonnes d'une source de données est développée sous Oracle en PL/SQL. Il était question ensuite de bénéficier de la technologie de Big Data MapReduce et de l'apport de l'outil Spark afin de développer l'algorithme de vérification d'une DF sur de gros volumes de données.

```
CREATE OR REPLACE PROCEDURE VERIFYDEPENDANCE
(VDATASOURCENAME CHAR, COLLEFT CHAR, COLRIGHT CHAR) AS
TYPE CurTyp IS REF CURSOR;
Cur CurTyp; Vn Number; vmax Number;
Vcol1 Varchar2(10); Vcol2 Varchar2(10); Vcol Varchar2(10);
Vrequete Varchar2(2000); msg Varchar2(500);
BEGIN
Vrequete :='DROP TABLE temp1';
EXECUTE IMMEDIATE Vrequete;
Vrequete :='CREATE TABLE temp1(COLUMNNAME1 VARCHAR(10), Nvaleur NUMBER)';
EXECUTE IMMEDIATE Vrequete;
Open Cur For 'Select '|| COLLEFT || ', Count(*) From
(Select Distinct '|| COLLEFT || ', '|| COLRIGHT || ' from '|| VDATASOURCENAME || ')
Group By '|| COLLEFT || '';
LOOP
FETCH Cur INTO VCol, Vn;
Exit When CurVrequete :='INSERT INTO temp1 VALUES (''|| Vcol ||'', '|| Vn ||')';
EXECUTE IMMEDIATE Vrequete;
END LOOP;
Close Cur;
EXECUTE IMMEDIATE 'COMMIT';
EXECUTE IMMEDIATE 'SELECT MAX(Nvaleur) from temp1' INTO Vmax;
If Vmax > 1 THEN
msg :='la dépendance n'est pas verfiée';
Dbms_Output.Put_Line(msg);
ELSE
msg :='la épendance est verfiée';
Dbms_Output.Put_Line(msg);
End If;
End;
```

Nous avons développé l'équivalent de l'algorithme de vérification de DF en Spark Scala

dont le script est donné dans la figure 5.14 suivante :

build.sbt × SparkApp.scala × data.csv × SparkAppTest.scala ×

SparkApp main(args: Array[String])

```
object SparkApp {

  def main (args: Array[String]) {


    if (args.length != 3) {
      System.err.println("usage <master> <inputfile> <ouputfile>")
      System.err.println(args.length)
      System.exit(1)
    }
    val sparkConf = new SparkConf().setAppName("SparkApp")
    sparkConf.setMaster(args(0)); //Local[x], yarn-client....
    val sc = new SparkContext(sparkConf)
    // split each document into words


    val tokenized = sc.textFile(args(1)).map(_.split(";"))
    val x = tokenized.map(f => (f(0), f(1))).distinct().groupByKey()
    val y = x.map(f=> (f._1,f._2.size))
    y.foreach(println(_))
    y.saveAsTextFile(args(2).toString())

```


FIGURE 5.14 – DF en Spark

Les mesures de performances en utilisant la technologie Big Data, comparées à celles utilisant le SGBD Oracle, seront données dans la figure (5.15).

Vérification d'une Dépendance Fonctionnelle : Oracle VS Spark						
COL7 --> COL8 : (CITY --> COUNTRY)						
Nombre de lignes	100 000	1 000 000	10 000 000	100 000 000	200 000 000	1 000 000 000
Temps d'exécution Oracle	0,03140 s	0,14940 s	5,40020 s	45,59400 s	84,87033 s	
Temps d'exécution Spark (Local)	0,01000 s	0,09000 s	2,00000 s	30,88800 s	55,72300 s	90,5s

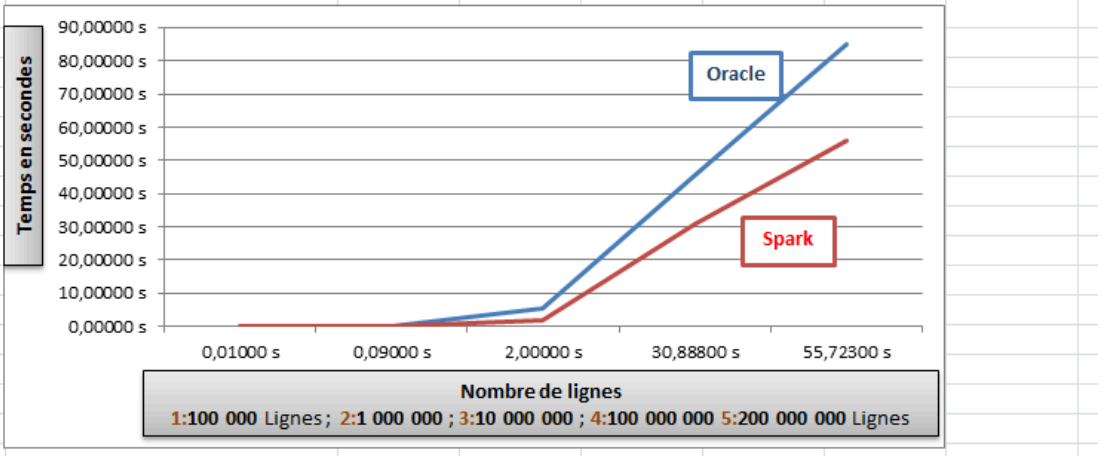


FIGURE 5.15 – Comparaison des perfermances de la vérification de DF Oracle vs Spark

5.6 Correction grâce aux dépendances fonctionnelles

Nous utilisons les valeurs valides stockées dans notre dictionnaire de données (DDVSLINKS). Il s'agit de faire une jointure entre la source de données et le dictionnaire DDVSLINKS et remplacer les valeurs de la partie gauche d'une dépendance fonctionnelle dans la source par les valeurs valides dans DDVSLINKS.

Les dépendances fonctionnelles détectées lors du diagnostic de la source sont : $COL7 \rightarrow COL8$, $COL8 \rightarrow COL9$ et $COL7 \rightarrow COL9$.

Elles correspondent sémantiquement aux trois dépendances fonctionnelles préstockées dans le DDVSLINKS, à savoir $CITY \rightarrow COUNTRY$, $COUNTRY \rightarrow CONTINENT$ et $CITY \rightarrow CONTINENT$.

On en déduit que plusieurs jointures doivent être réalisées afin de corriger certaines anomalies syntaxiques et déduire les valeurs exactes de certaines valeurs nulles.

Exemple : correction grâce aux dépendances fonctionnelles $COL8 \rightarrow COL9$

```
CREATE TABLE TEMP1
(COL1,COL2,COL3,COL4,COL5,COL6,COL7,COL8,COL9,COL10)
AS
(SELECT T2.COL1, T2.COL2, T2.COL3, T2.COL4, T2.COL5,
T2.COL6, T2.COL7, T2.COL8, T1.FRENCHR, T2.COL10
FROM DDVSLINKS T1, DSPrim T2
WHERE T1.CATEGORYL='COUNTRY'
AND T1.CATEGORYR='CONTINENT'
AND T1.FRENCHL=T2.COL8 ) ;
```

Une ou plusieurs jointures de la source de données avec le dictionnaire DDVSLINKS doivent être établies afin de corriger une partie de valeurs dépendantes de la source de dépendance fonctionnelle. La première jointure permet, par exemple, de corriger les valeurs {Franc, Fr, Frence} en la valeur France, vu que la valeur France est associée à Paris. Toutes les valeurs nulles correspondantes à Paris seront remplacées par France.

Les mesures de performances en utilisant la technologie Big Data, comparées à celles utilisant le SGBD Oracle, seront données dans la figure (5.16).

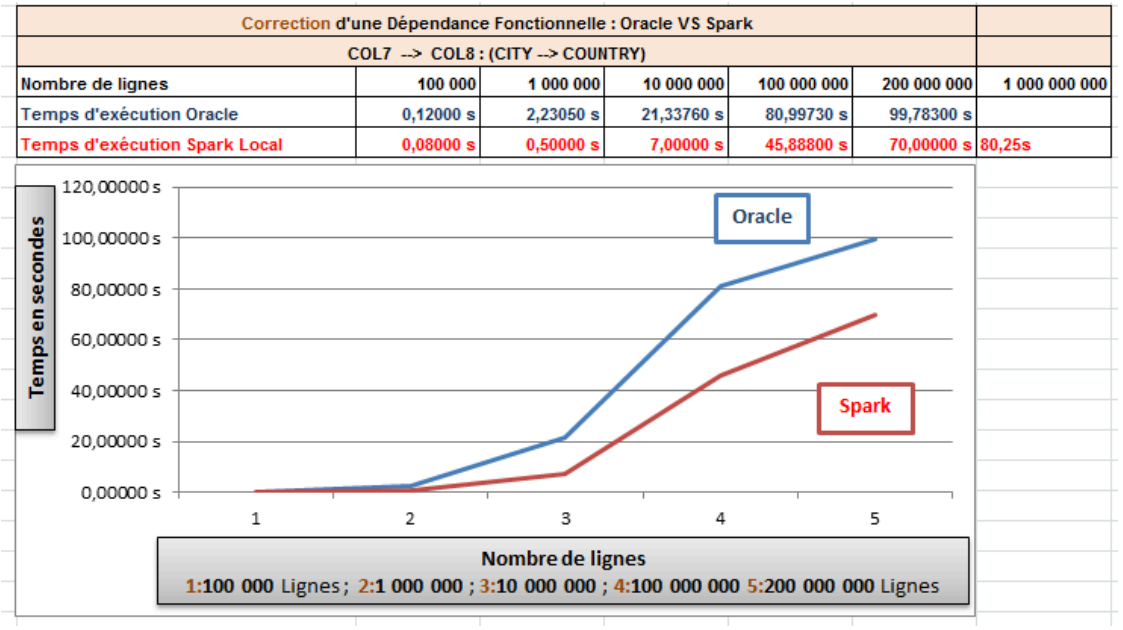


FIGURE 5.16 – Comparaison des perfermances de la correction de DF Oracle vs Spark

Remarque : les mesures devront être refaites à cause d'un problème machine.

5.7 Bilan

L'outil iDQMS permet la reconnaissance sémantique des structures de données, l'établissement de plusieurs rapports d'anomalies et le nettoyage de données au niveau intra et inter-colonnes.
Notre outil permet de guider l'utilisateur pour déterminer le sens de chaque colonne d'une source de données, le type de données, la catégorie sémantique et la sous-catégorie (telle que la langue utilisée), les contraintes et les commentaires de chaque colonne. Nous recommandons aussi les liens sémantiques qui peuvent exister entre les colonnes d'une source de données, les algorithmes de distance de similarité ainsi que le seuil à utliser lors des comparaisons.
Nous exploitons les rapports qui portent sur les anomalies détectées et les connaissances sémantiques déduites à partir de l'étape de la reconnaissance sémantique pour corriger les anomalies au niveau intra et inter-colonnes et traiter certaines valeurs nulles.
Nous avons constaté, après l'application de notre processus de nettoyage, qu'il est possible de détecter des anomalies sur les données. Ceci peut être dû au choix de l'ordre dans le traitement de dépendances fonctionnelles. Une étude approfondie, sur la priorité des traitements de dépendances fonctionnelles à mener, devrait donc être faite.
Une étude plus détaillée devra être menée, afin de permettre la découverte de la sémantique d'une colonne, dans le cas où les différents dictionnaires ne le permettent pas. Des algorithmes d'apprentissage pourraient aider à la découverte du sens de la donnée.
L'amélioration des performances des processus de détection des anomalies et de corrections de ces dernières, en utilisant la technologie Big Data, constitue un des objectifs à atteindre. En effet, les mesures réalisées sur Spark sont très prometteuses. Les différents calculs des indicateurs, afin d'assister les utilisateurs dans les différentes tâches de corrections, doivent être réalisés sur de très gros volumes en un temps raisonnable.

5.8 Conclusion

L'outil iDQMS permet d'assister l'utilisateur dans le processus de la découverte de la structure de données. Cette dernière est composée de noms sémantiques de colonnes, les types de données, les sous-catégories (langues), les liens sémantiques entre les colonnes, les contraintes, les mesures de similarité et les commentaires.
La description détaillée des données devra permettre de guider la détection et la correction des anomalies intra et inter-colonnes.
Les mesures réalisées ont montré que l'utilisation de la technologie BigData-MapReduce-Hadoop-Spark permet d'améliorer d'une manière très significative les performances des

algorithmes de vérification et de correction de certaines anomalies. Il serait donc très intéressant de poursuivre ces travaux.

Chapitre 6

Conclusion et Perspectives

Bilan

Le traitement des anomalies dans les données occupe de nos jours une grande part de l'activité informatique et de gestion des données dans les entreprises. Ce sont des anomalies qui ne sont généralement détectées qu'au niveau de la restitution des données telles que la visualisation des données ou l'analyse de ces dernières. On constate alors, avec beaucoup de retard par exemple, l'incohérence des données des sources et l'imprécision des indicateurs calculés et agrégés. Ceci peut avoir un coût très élevé pour les organisations et les entreprises.
Il faut constater que les outils ETL d'intégration existants sur le marché tels que Talend [Ref. a] ou Informatica [Ref. c] n'assistent malheureusement pas les utilisateurs dans leur démarche de création de nouvelles masses de données issues de l'assemblage de plusieurs sources le plus souvent hétérogènes, distribuées et de qualités variées.
La gestion des données est un des axes stratégiques que les grandes entreprises visent à améliorer continuellement dans le but de garantir des services rapides et fiables en manipulant des données crédibles (cohérentes, mises à jour régulièrement et non obsolètes).

Notre objectif dans cette thèse était alors de participer à la construction de nouveaux outils intelligents qui accompagnent les utilisateurs pendant les différentes phases de construction de données cibles de tailles généralement très importantes (Bases et Entrepôts de données, Masses de données Big Data). La détection et la correction automatique des éventuelles anomalies dans les données étaient notre objectif principal. L'optimisation des performances a constitué notre second objectif. En effet, nous avons appliqué la technologie MapReduce du Big Data afin de traiter la volumétrie importante des données manipulées en utilisant SPARK/Scala.

La grande majorité de ce travail s'est déroulée en collaboration avec la société Talend éditeur de l'ETL "Talend Data Integration et Talend Data Quality" afin d'améliorer les fonctionnalités de cet outil dans les étapes de diagnostic des erreurs et de corrections de ces dernières. Lors de la thèse de Mme Aicha BEN SALEM, l'accent a été mis sur les anomalies iner-lignes dans une source de données, plusieurs algorithmes d'élimination des doubles et des similaires ont été développés. Alors que dans le présent mémoire, nous abordons les anomalies causées par la violation de dépendances fonctionnelles à savoir les anomalies inter-colonnes.

Les sources de données étant le plus souvent mal renseignées, les schémas de description des données sont incompréhensibles voire inexistants, nous avons basé notre réflexion sur la sémantique des données existante dans la ou les sources traitées.
Nous parlons alors de qualité contextuelle des données. Nous essayons de rattacher aux données leurs sémantiques, leurs types, leurs contraintes et des commentaires si possible.
La quasi-totalité des outils ETL (Intégration et Qualité) existants se focalisent le plus souvent uniquement sur la donnée brute et non sur la signification de celle-ci. Ils ne répondent donc pas aux problèmes soulevés tels que le choix des attributs clés pour l'élimination des doubles ou les dépendances sémantiques à vérifier et à corriger. Le diagnostic des problèmes dans les données exigent des utilisateurs la connaissance des structures et une certaine expertise. C'est pourquoi, et afin d'assister l'utilisateur, nous avons abordé d'une manière originale l'extraction de la sémantique des données à partir de toutes les informations disponibles telles que celles de la source de données ou encore les dictionnaires de données préétablis. Dans un second temps, dans le cadre du projet avec Talend, les méthodes de nettoyage (au sein d'une même colonne, entre les lignes et enfin entre les colonnes) ont constitué nos objectifs avec le souci de l'optimisation des performances dans le cas de gros volumes.

La **première contribution** de cette thèse concerne la catégorisation des données qui permet la reconnaissance du schéma d'une source de données considérée au format CSV et donc sans schéma à priori. Cette approche est composée de plusieurs étapes : (i) La découverte de la sémantique de chaque colonne grâce à son contenu. Des mesures (indicateurs) sont calculées afin de rattacher chaque colonne à une catégorie pré-stockée dans les dictionnaires pré-établis (DDCAT, DDVS, DDKW et DDRE) ; (ii) La déduction des liens sémantiques qui peuvent exister entre les colonnes en profitant de la sémantique issue de la première étape. L'originalité de ce travail consiste en la jointure de la source de données avec les dictionnaires DDCAT (liste de toutes les catégories prédéfinies) et DDVSLinks

(issu du dictionnaire DDVS), liste de toutes les dépendances prédéfinies afin de corriger certaines erreurs et surtout traiter les valeurs nulles. Les indicateurs sont alors recalculés afin de valider la sémantique de chaque colonne. Cette reconnaissance de schéma assistera l'utilisateur dans la compréhension des données grâce aux rapports d'anomalies. La correction peut être assistée par ordinateur.

La **deuxième contribution** est la proposition d'actions correctives au sein d'une même colonne et inter-colonnes. Il s'agit évidemment, d'une part, d'homogénéiser les données au sein d'une même colonne (corrections syntaxiques et sémantiques pour unifier les formats et les sous-catégories) pour ensuite, d'autre part, revérifier les dépendances fonctionnelles et corriger les violations de ces contraintes. D'autres tests devront être effectués notamment lorsque la valeur dans une colonne est composée de plusieurs mots dépendants tels que le nom suivi du prénom ou le prénom suivi du nom ou encore le code postal et la ville.

La **troisième contribution** porte sur des questions de performances. Nous avons fait appel aux concepts de MapReduce et donc aux nouvelles notions de Big Data aux niveaux des processus de vérification des dépendances fonctionnelles détectées et de corrections des anomalies. Les performances de l'outil SPARK sont très prometteuses et dépassent de loin celles du système Oracle.

Perspectives

Les perspectives de cette thèse sont nombreuses vu les différents domaines abordés. Nos travaux sont basés sur un ensemble de dictionnaires pré-établis (des données de référence) afin de catégoriser les données et de découvrir les liens qui peuvent exister entre elles dans le but de détecter et corriger les anomalies.

La reconnaissance sémantique des données et de leurs structures nécessite l'enrichissement des données de référence. L'enrichissement automatique des différents dictionnaires constitue une direction à poursuivre. En effet, le Web offre de nos jours une très grande quantité de données à exploiter qui malheureusement demeure de mauvaise qualité. Leurs descriptifs et les commentaires sont quasi inexistants et dépendent de la langue utilisée. La découverte automatique des liens sémantiques est ainsi une tâche très difficile. Il serait très utile de poursuivre l'automatisation de cette tâche.
La catégorisation des données en provenance du Web pourrait se baser sur des algorithmes d'apprentissage.

L'intégration des données hétérogènes (structurées et non structurées) est une piste très intéressante à explorer afin d'améliorer la qualité des données. A l'ère du Big Data, les données sont effectivement abondantes, hétérogènes et sont perpétuellement en activité. Il serait très intéressant d'appliquer des algorithmes tels que ceux de l'apprentissage (supervisé ou semi-supervisé) ou du traitement automatique des langues ou de fouille de textes afin d'extraire les actions qui permettent de mettre à jour régulièrement des bases de données structurées, les données sont alors réactualisées et corrigées.

L'amélioration des performances des processus de détection des anomalies et de corrections de ces dernières, en utilisant la technologie Big Data, constitue un des objectifs à atteindre. En effet, les mesures réalisées sur Spark sont très prometteuses. Les différents calculs des indicateurs, afin d'assister les utilisateurs dans les différentes tâches de corrections, doivent être réalisés sur de très gros volumes en un temps raisonnable. Il en est de même des algorithmes de dépendances fonctionnelles et de l'élimination des doubles ou similaires. Ces actions sont très dépendantes et influencent grandement la qualité du résultat final.
Nous pensions que l'étape de l'élimniation des doubles devait se faire avant l'étude des dépendances fonctionnelles. Il s'est avéré que, dans un grand nombre de cas, le traitement de la violation des dépendances fonctionnelles permet de substituer certaines valeurs nulles par la « bonne » valeur en provenance du dictionnaire. Une étude très approfondie, sur la prioriété des actions de corrections à mener, devrait être faite sur tous les niveaux (calculs des mesures, actions de nettoyage).

Une étude plus détaillée sur les liens sémantiques entre les données devra être menée. Les dépendances sémantiques entre les colonnes d'une source de données sont de plusieurs types tels que les dépendances fonctionnelles exactes, les dépendances fonctionnelles conditionnelles, les dépendances d'inclusions. Cette étude devra permettre de proposer des actions correctives dans le cas d'anomalies dans les données et d'inférer des mises à jour des dictionnaires. Les méthodes de calculs de distances de similarité devraient prendre en considération les aspects sémantiques.

Bibliographie

Talend, a. URL https://www.talend.com/.

Pentahodataintegration, b. URL http://www.pentaho.fr/explore/pentaho-data-integration.

informatica, c. URL https://www.informatica.com/fr/.

Digital data processing, 2016. URL http://lipn.univ-paris13.fr/~bennani/tmpc/TND/TND1.pdf.

C. Batini, C. Cappiello, and C. Francalanci. Methodologies for data quality assessment and improvement. *Journal ACM Computing Surveys*, Volume 41 :16 :1–16 :52, 2009.

A. Ben-Salem, F. Boufarès, and S.Correia. Semantic data profiling for bigdata. In *Proceedings of the 28th International Conference on Computer Applications in Industry and Engineering (CAINE 2015), San Diego, California, USA*, pages 139–146, 2015.

O. Benjalloun, H. Garcia-Molina, D. Menestria, S.E Whang, Q.Su, and J.Widom. Swooch : A generic approach to entity resolution. *The 35th International Journal on Very Large Data Bases (VLDB)*, pages 255–276, 2009.

A. BenSalem. *Qualité contextuelle des données : Détection et nettoyage guidés par la sémantique des données.* PhD thesis, Université Sorbonne Paris cité, Paris, France, 2015.

L. Berti-équille. Assurer la qualité des données : un défi permanent pour les systèmes d'information, bases et entrepôts de données. *Revue Génie logiciel*, Volume 1265-1397 : 13–20, 2005.

L. Berti-Equille. Panorama des méthodes de détection et de traitement des anomalies. *5 ème journée thématiques Apprentissage Artificiel et Fouille de Données, Université Paris 13, Villetaneuse, France, Présentation*, 2012.

P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In *In Proceedings of the 23rd International Conference on Data Engineering (ICDE 2007), Istanbul, Turkey*, page 746755, 2007.

F. Boufarès. *Des Bases de Données aux Entrepôts de Données, Contribution au développement de nouveaux outils de gestion de la Qualité des Données.* HDR, Université Sorbonne Paris cité Paris 13, Villetaneuse, France, 2012.

F. Boufarès, A. Ben Salem, and S. Correia. Qualité de données dans les entrepôts de données : élimination des similaires. *8ème Journées francophones sur les Entrepôts de Données et l'Analyse en ligne, Revue des Nouvelles Technologies de l'Information, Bordeaux, France*, RNTI-B-8 :32–41, 2012.

F. Boufarès, A. Ben Salem, M.Rehab, and S. Correia. Similar elimination data : Mfb algorithm. *IEEE-2013 International Conference on Control, Decision and Information Technologies, Hammamet, Tunisie*, pages 289–293, 2013.

F. Chiang and R. J. Millert. Discovering data quality rules. *Journal on Very Large Data Bases (VLDB)*, pages 1166–1177, 2008.

X. Chu1and, J. Morcos, I.F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. Katara : A data cleaning system powered by knowledge bases and crowdsourcing. *Proceedings of the 2015 ACM International Conference on Management of Data (SIGMOD 2015)*, pages 1247–1261, 2015.

W.W. Cohen. Eliminating fuzzy duplicates in data warehouses. In *In the 28th international conference on Very Large Data Bases (VLDB2002), Hong Kong, China*, page 586597, 2002.

M. Dallachiesay, A. Ebaid, A. Eldawy, A. Elmagarmid, I.F. Ilyas, M. Ouzzani, and N. Tang. Nadeef : a commodity data cleaning system. *ACM SIGMOD International Conference on Management of Data*, pages 541–552, 2013.

J. Dean and S. Ghemawat. Mapreduce : simplified data processing on large clusters. In *The 6th Conference On Symposium On Operating Systems Design And Implementation (OSDI2004), California, USA*, page 137150, 2004.

D. Deng, Y. Jiang, G. Li, J. Li, and C. Yu. Scalable column concept determination for web tables using large knowledge bases. *Journal on Very Large Data Bases (VLDB)*, Volume 6 Issue 13 :1606–1617, 2013.

T. Diallo and N. Novelli. Découverte des dépendances fonctionnelles conditionnelles. *Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, Revue des Nouvelle Technologies de l'information*, 2010.

T.M Diallo. *Discovring data quality rules in a master data management context.* PhD thesis, Institut National des Sciences Appliquées de Lyon, Lyon, France, 2013.

C. Eaton, D. Deroos, T. Deutsch, G. Lapis, and P. Zikopoulos. *Understanding Big Data.* McGraw-Hill Companies, 2012.

A. Elbyed. *Une approche dalignement dontologies à bases dinstances.* PhD thesis, Institut National des Télécommunication, Evry, France, 2009.

A.K. Elmagarmid, P.G. Ipeirotis, and V.S. Verykios. Duplicate record detection : A survey. *In IEEE Transactions on Knowledge and Data Engineering*, Volume 19 :116, 2007.

R. Erhard. *Towards Large-Scale Schema and Ontology Matching.* Data-Centric Systems and Application, 2011.

R. Erhard and A.B Philip. A survey of approaches to automatic schema matching. *Journal on Very Large Data Bases (VLDB)*, Volume 10 :334–350, 2011.

A. Ben-Salem F. Boufarès and S.Correia. Un algorithme de déduplication pour les bases et entrepôts de données. In *In Congrès INFormatique des ORganisations et Systèmes dInformation et de Décision (INFORSID 2012), Montpellier, France*, page 497506, 2012.

E. Garnaud. *Dépendances fonctionnelles : extraction et exploitation.* PhD thesis, Université de Bordeaux, Bordeaux, France, 2013.

E. Garnaud, N. Hanusse, S. Maabout, and N. Novelli. Calcul parallèle de dépendances. *9ème journées Bases de Données Avancées*, 2013.

S. Hamdoun and F. Boufarès. Un formalisme pour l'intégration de données hétérogènes. *In Revue des Nouvelles Technologies de lInformation, (RNTI)*, Volume B-6 :107119, 2010.

M.A. Hernandez and S.J. Stolfo. Data cleansing and the mergepurge problem. *Data Mining and Knowledge Discovery (DMKD 1998), New York, USA*, pages 9–37, 1998.

Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. Tane : An efficient algorithm for discovering functional and approximate dependencies. *Computer Journal*, Volume 42, No. 2 :100–111, 1999.

IBM. Fichier csv. URL `http://www.ibm.com/support/knowledgecenter/fr/SVU13_7.2.1/com.ibm.ismsaas.doc/import/r_sample_csv_files.html`.

K. Berberich J. Hoffart, F. M. Suchanek and G. Weikumt. Yago2 : A spatially and temporally enhanced knowledge base from wikipedia. In *The 6th Conference On Symposium On Operating Systems Design And Implementation (OSDI 2004), California, USA*, 2013.

S. Hamdoun khalfallah. *Construction d'entrepôts de données par integration de sources hétérogènes*. PhD thesis, Université Paris 13, Villtaneuse, France, 2006.

J. Kivinen and H. Mannila. Approximate inference of functional dependencies from relations. *Theoretical Computer Science*, Volume 149, No. 1 :129–149, 1995.

N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage : Similarity measures and algorithms. *In the ACM International Conference on Management of Data (SIGMOD 2006)*, page 802803, 2006.

J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, Volume 6(2) :167195, 2015.

V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *In Doklady Akademii Nauk SSSR*, Volume 163 no. 4 :845848, 2007.

G. Limayeand, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Journal on Very Large Data Bases (VLDB)*, Volume 3 Issue 1-2 :1338–1347, 2010.

J. Liu, J. Li, C. Liu, and Y. Chen. Discover dependencies from dataa review. *IEEE Transactions on Knowledge and Data Engineering*, Volume 24, no. 2 :251264, 2012.

D. Menard. Schémas de bases de données, 2008. URL `http://www.ascodocpsy.org/santepsy/AutoDoc/?filename=fab.schemas#fab.schemas.introduction`.

N. Novelli and R. Cicchetti. Fun : An efficient algorithm for mining functional and embedded dependencies. In Springer, editor, *In Proceedings of the 8th International Conference on Database-Theory (ICDT 2001), London, United Kingdom*, volume Volume 1973, pages 189–203, 2001.

P. Oliveira, F. Rodrigues, and P. Henriques. A formal definition of data quality problems. *10th International Conference on Information Quality, Cambridge, Massachusetts, USA*, pages 181–184, 2005a.

P. Oliveira, F. Rodrigues, P. Henriques, and H. Galhardas. A taxonomy of data quality problems. *2nd International Workshop on Data and Information Quality, Portugal*, pages 219–233, 2005b.

R.K. Pearson. The problem of disguised missing data. *SIGKDD Explorations*, Volume 8, Issue 1 :83–92, 2006.

P.Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics IV*, Volume 3730 :146–171, 2005.

E. Rahm and H.H Do. Data cleaning : Problems and current approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*, Volume 23 :03–11, 2000.

F. Saais. *Intégration sémantique de données guidée par une ontologie.* PhD thesis, Université Paris-Sud, Orsay, France, 2007.

C. Samitsch. *Data Quality and its Impacts on Decision Making.* Springer Gabler, 2015.

S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *The 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2008), Alberta, Canada*, page 269278, 2002.

E. Simonenko and N. Novelli. Extraction de dépendances fonctionnelles approximatives : une approche incrémentale. *12ième Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, Revue des Nouvelle Technologies de l'information*, Volume RNTI-E-23 :95–100, 2012a.

E. Simonenko and N. Novelli. Extraction de dépendances fonctionnelles approximatives : une approche incrémentale. *12ième Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, Bordeaux, France*, pages 95–100, 2012b.

H. Tanghe, S. Ait Daoud, P.O Gibert, S. Sahri, S. Benbernou, J.Totel, L. Bertin, C.Diaconu, I. Ghorbel, D. Geldwerth-Feniger, L. Abidi, C. Cérin, P. Werle, and M. Lafaille. *Approches Contemporaines en hébergement et gestion de données.* 2016.

C. Toulemonde. *JEMM research_ Informatica : Le capital de votre organisation.* Un livre blanc de JEMM research, 2008.

E. Ukkonen. Approximate string matching with q-grams and maximal matches. *Theoretical Computer Science*, Volume 92, no. 1 :191211, 1992.

P. Venetis, A. Y. Halevy, J. Madhavan, W. Shen M. Pasca, F. Wu, G. Miao, and C. Wu. Recovering semantics of tables on the web. *Journal on Very Large Data Bases (VLDB)*, Volume 4 Issue 9 :528–538, 2011.

W.E. Winkler. The double metaphone search algorithm. *C/C++ Users Journal*, Volume 18, no. 5 :3843, 2000.

W.E. Winkler. Overview of record linkage and current research directions. *Research Report Series, RRS*, page 144, 2006.

H. Zaidi, Y. Pollet, F. Boufarès, and N. Kraiem. Semantic of data dependencies to improve the data quality. *5th International Conference on Model & Data Engenieering (MEDI 2015)*, LNCS 9344 :53–61, 2015.

H. Zaidi, F. Boufarès, and Y. Pollet. Nettoyage de données guidé par la sémantique inter-colonnes. *16th Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, Bordeaux, France*, Volume RNTI-E-30 :18–22, 2016a.

H. Zaidi, F. Boufarès, and Y. Pollet. Improve data quality by processing null values and semantic dependencies. *8th International Conference on Computational Intelligence and Software Engineering (CiSE 2016), Journal of Computer and Communications*, Volume 4 No.5 :78–85, 2016b.

M.A Zoghlami, M. Sassi Hidri, and R. Ben Ayed. Sampling-based consensus fuzzy clustering on big data. In IEEE, editor, *International Conference on Fuzzy Systems, FUZZ-IEEE 2016, Vancouver, BC, Canada*, pages 1501–1508, 2016.

Chapitre 7

Annexe

Procédures PL/SQL

```
CREATE OR REPLACE PROCEDURE DuplicSource (VDATASOURCENAME CHAR) AS
VREQUETE varchar2(500);
BEGIN
VREQUETE :='DROP TABLE DSPrim';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :='CREATE TABLE DSPrim AS (SELECT * FROM '|| VDATASOURCENAME ||')';
EXECUTE IMMEDIATE VREQUETE; END;
```

```
CREATE OR REPLACE PROCEDURE VERIFCATSUBCATDATA
(VDATASOURCENAME CHAR) AS
TYPE CurTyp IS REF CURSOR;
Cur CurTyp;
Vn Number (5);
Vcol Varchar2(10);
Vcategory Varchar2(20);
Vrequete Varchar2(2000);
VDATATYPEE VARCHAR(10);
VCOLUMNIDENTIFIER NUMBER;
VNBRCOL NUMBER;
i NUMBER;
BEGIN
VREQUETE :='delete from DATAREPORTCATCOL';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :='delete from DATAREPORTSUBCATCOL';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :='Select COUNT(COLUMN_NAME) from USER_TAB_COLUMNS where TABLE_NAME =
"' || VDATASOURCENAME || "";
EXECUTE IMMEDIATE VREQUETE INTO VNBRCOL;
FOR i IN 1..VNBRCOL LOOP
VCOLUMNIDENTIFIER :=i;
Open Cur For 'Select CATEGORY, Count(*) From
(Select Distinct COL' || to_char(i) || ', CATEGORY From
'|| VDATASOURCENAME || ', DDVSTOT WHERE '|| VDATASOURCENAME || '.COL' ||
to_char(i) || '=DDVSTOT.VALIDSTRING) Group By CATEGORY';
```

```
LOOP
FETCH Cur INTO Vcategory, Vn;
Exit When Cur%Notfound;
VCol := 'COL' || to_char(i) || '';
VDATATYPEE := 'String';
Vrequete :='INSERT INTO DATAREPORTCATCOL VALUES
('''|| Vcol ||''','''|| VDATATYPEE ||''','''|| Vcategory ||''', '''|| Vn ||''',
'''||VCOLUMNIDENTIFIER||''', '''||VDATASOURCENAME||''', SYSDATE)';
Dbms_Output.Put_Line(Vrequete);
EXECUTE IMMEDIATE VREQUETE;
END LOOP;
Close Cur;
EXECUTE IMMEDIATE 'COMMIT';
END LOOP;
FOR i IN 1..VNBRCOL LOOP
VCOLUMNIDENTIFIER :=i;
Open Cur For 'Select SUBCATEGORY, Count(*) From
(Select Distinct COL' || to_char(i) || ', SUBCATEGORY From '|| VDATASOURCENAME || ',
DDVSTOT WHERE '|| VDATASOURCENAME || '.COL' || to_char(i) || '=DDVSTOT.VALIDSTRING)
Group By SUBCATEGORY';
LOOP
FETCH Cur INTO Vcategory, Vn;
Exit When Cur%Notfound;
VCol := 'COL' || to_char(i) || '';
VDATATYPEE := 'String';
Vrequete :='INSERT INTO DATAREPORTSUBCATCOL VALUES
('''|| Vcol ||''','''|| VDATATYPEE ||''','''|| Vcategory ||''', '''|| Vn ||''',
'''||VCOLUMNIDENTIFIER||''','''||VDATASOURCENAME||''', SYSDATE)';
EXECUTE IMMEDIATE VREQUETE;
END LOOP;
Close Cur;
EXECUTE IMMEDIATE 'COMMIT';
END LOOP;
END;
```

```
CREATE OR REPLACE PROCEDURE VERIFCATRE
(VDATASOURCENAME CHAR) AS
TYPE CurTyp IS REF CURSOR;
Cur CurTyp;
Vn Number (5);
Vcol Varchar2(10);
Vcategory Varchar2(20);
Vrequete Varchar2(2000);
VDATATYPEE VARCHAR(10);
Vreq Varchar2(500);
VCOLUMNIDENTIFIER NUMBER;
VNBRCOL NUMBER;
BEGIN
VREQUETE :='delete from CATDATAREPORTCOL';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :='Select COUNT(COLUMN_NAME) from USER_TAB_COLUMNS where TABLE_NAME =
"' || VDATASOURCENAME || "";
EXECUTE IMMEDIATE VREQUETE INTO VNBRCOL;
FOR i IN 1..VNBRCOL LOOP
VCOLUMNIDENTIFIER :=i;
Open Cur For 'Select CATEGORY, Count(*) From
(Select COL' || to_char(i) || ', CATEGORY From '|| VDATASOURCENAME || ', DDRE WHERE RE-
GEXP_LIKE('|| VDATASOURCENAME || '.COL' || to_char(i) || ',DDRE.REGEXPR)) Group By CATEGO-
RY';
Vreq := 'Select CATEGORY, Count(*) From (Select DISTINCT COL' || to_char(i) || ', CATEGORY
From '|| VDATASOURCENAME || ', DDRE WHERE REGEXP_LIKE('|| VDATASOURCENAME || '.COL' ||
to_char(i) || ',DDRE.REGEXPR )) Group By CATEGORY';
Dbms_Output.Put_Line(Vreq);
LOOP
FETCH Cur INTO Vcategory, Vn;
Exit When Cur%Notfound;
VCol := 'COL' || to_char(i) || ";
VDATATYPEE := 'String';
Vrequete :='INSERT INTO CATDATAREPORTCOL VALUES ( "'|| Vcol ||"',"'|| VDATATYPEE ||"',
"'|| Vcategory ||"', "'|| Vn ||"',"'|| VCOLUMNIDENTIFIER ||"',
"'|| VDATASOURCENAME ||"',SYSDATE)';
EXECUTE IMMEDIATE VREQUETE;
END LOOP;
Close Cur;
EXECUTE IMMEDIATE 'COMMIT';
END LOOP;
End;
```

```
CREATE OR REPLACE PROCEDURE DIAGNOSTICDS (VDATASOURCENAME CHAR) AS
VM000 NUMBER;
VM001 NUMBER;
VM002 NUMBER;
VM003 NUMBER;
VM004 NUMBER;
VM005 NUMBER;
VM006 NUMBER;
VMsubcatfr NUMBER;
VMsubcateng NUMBER;
VM011 VARCHAR(30);
VREQUETE varchar2(2000);
BEGIN
VREQUETE :='DROP TABLE DRDIAGNOGLOBAL';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :='
CREATE TABLE DRDIAGNOGLOBAL
(DATASOURCE VARCHAR(10),
M000 NUMBER,
M001 NUMBER,
M002 NUMBER,
M003 NUMBER,
M004 NUMBER,
M005 NUMBER,
M006 NUMBER,
M011 VARCHAR(30),
DATEANALYSIS DATE )';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :='SELECT COUNT(*)FROM ' || VDATASOURCENAME;
EXECUTE IMMEDIATE VREQUETE INTO VM000;
VREQUETE :='Select COUNT(COLUMN_NAME) from USER_TAB_COLUMNS
where TABLE_NAME = "' || VDATASOURCENAME || "";
EXECUTE IMMEDIATE VREQUETE INTO VM001;
VM002 :=VM000*VM001;
```

```
VREQUETE :='Select SUM(M101) from datareportcol';
EXECUTE IMMEDIATE VREQUETE INTO VM003;
VREQUETE :='Select SUM(M102) from datareportcol';
EXECUTE IMMEDIATE VREQUETE INTO VM004;
VREQUETE :='Select SUM(M104) from datareportcol';
EXECUTE IMMEDIATE VREQUETE INTO VM005;
VREQUETE :='Select SUM(M105) from datareportcol';
EXECUTE IMMEDIATE VREQUETE INTO VM006;
VREQUETE :='Select SUM(NBRVALSUBCAT ) from DATAREPORTSUBCATCOL
WHERE SUBCATEGORY="FRENCH"';
EXECUTE IMMEDIATE VREQUETE INTO VMsubcatfr;
VREQUETE :='Select SUM(NBRVALSUBCAT ) from DATAREPORTSUBCATCOL
WHERE SUBCATEGORY="ENGLISH"';
EXECUTE IMMEDIATE VREQUETE INTO VMsubcateng;
If VMsubcatfr > VMsubcateng
then
VM011 :='FRENCH';
else
VM011 :='ENGLISH';
END IF;
VREQUETE :='INSERT INTO DRDIAGNOGLOBAL VALUES (
"'||VDATASOURCENAME||"',
"'||VM000||"',
"'||VM001||"',
"'||VM002||"',
"'||VM003||"',
"'||VM004||"',
"'||VM005||"',
"'||VM006||"',
"'||VM011||"',
SYSDATE)';
EXECUTE IMMEDIATE VREQUETE;
END;
```

```
CREATE OR REPLACE PROCEDURE SPLIT_COLUMN(nom_tab VARCHAR ,
nom_column VARCHAR) AS
query VARCHAR(500);
BEGIN
query :='create table TA (col1, nbremots, col1A, col1B) as (select
'||nom_column||',length('||nom_column||')-length(replace('||nom_column||',
" ","""))+1,substr('||nom_column||',0,instr('||nom_column||',
" ")-1),substr('||nom_column||',instr('||nom_column||'," ")+1) from '||nom_tab||')';
execute immediate query;
END;
```

```
CREATE OR REPLACE PROCEDURE DiscoverSEMANTICDFLinks AS
VREQUETE varchar2(2000);
BEGIN
VREQUETE :='DROP TABLE DRDF';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :=' Create table DRDF (COLUMNNAMEL, CATEGORYL, CATEGORYR)
AS (
select DISTINCT DRDIAGNOINTRACOL.COLUMNNAME,
DRDIAGNOINTRACOL.CATEGORYENG ,
DDCATLINKS3.CATEGORYR FROM DRDIAGNOINTRACOL, DDCATLINKS3
WHERE DRDIAGNOINTRACOL.CATEGORYENG=DDCATLINKS3.CATEGORYL)';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :='DROP TABLE DRDIAGNOINTERCOLDF';
EXECUTE IMMEDIATE VREQUETE;
VREQUETE :=' Create table DRDIAGNOINTERCOLDF
(COLUMNNAMEL, CATEGORYL,COLUMNNAMER, CATEGORYR)
AS ( select DISTINCT COLUMNNAMEL, CATEGORYL, COLUMNNAME, CATEGORYR
FROM DRDF T1, DRDIAGNOINTRACOL T2
WHERE T1.CATEGORYR =T2.CATEGORYENG)';
EXECUTE IMMEDIATE VREQUETE;
END;
```

le cnam

Houda ZAIDI

Amélioration de la qualité des données

Correction sémantique des anomalies inter-colonnes

Résumé :

La qualité des données présente un grand enjeu au sein d'une organisation et influe énormément sur la qualité de ses services et sur sa rentabilité. La présence de données erronées engendre donc des préoccupations importantes autour de cette qualité. Ce rapport traite la problématique de l'amélioration de la qualité des données dans les grosses masses de données. Notre approche consiste à aider l'utilisateur afin de mieux comprendre les schémas des données manipulées, mais aussi définir les actions à réaliser sur celles-ci. Nous abordons plusieurs concepts tels que les anomalies des données au sein d'une même colonne, et les anomalies entre les colonnes relatives aux dépendances fonctionnelles. Nous proposons dans ce contexte plusieurs moyens de pallier ces défauts en nous intéressons à la performance des traitements ainsi opérés.

Mots clés :
Qualité des données, Dépendances fonctionnelles, Dépendances sémantiques, traitement des valeurs nulles, nettoyage de données.

Abstract :

Data quality represents a major challenge because the cost of anomalies can be very high especially for large databases in enterprises that need to exchange information between systems and integrate large amounts of data. Decision making using erroneous data has a bad influence on the activities of organizations. Quantity of data continues to increase as well as the risks of anomalies. The automatic correction of these anomalies is a topic that is becoming more important both in business and in the academic world.
In this report, we propose an approach to better understand the semantics and the structure of the data. Our approach helps to correct automatically the intra-column anomalies and the inter-columns ones. We aim to improve the quality of data by processing the null values and the semantic dependencies between columns.

Keywords :
Data Quality, Functional Dependencies, Semantic Dependencies, Null Values, Data Cleaning, Big Data.