# Qualité des données

**par Laure Berti-Équille**

*Maître de Conférences, Université de Rennes I*

## 1 La qualité des données: un état des lieux

Les problèmes de qualité des données stockées dans les bases et les entrepôts de données se propagent de façon endémique à tous les types de données (structurées ou non) et dans tous les domaines d'application : données gouvernementales, commerciales, industrielles ou scientifiques. Il s'agit en particulier d'erreurs sur les données, de doublons, d'incohérences, de valeurs manquantes, incomplètes, incertaines, obsolètes, aberrantes ou peu fiables. Les conséquences de la non qualité des données (ou de leur qualité médiocre) sur les prises de décision et les coûts financiers qu'elle engendre sont considérables[1]. Avec la multiplication des sources d'informations disponibles et l'accroissement des volumes de données potentiellement accessibles, la qualité des données et, plus largement, la qualité des informations ont pris une place de premier plan, d'abord, au sein des entreprises et, depuis ces dix dernières années, dans le monde académique [31][2][8][28][34]. Il n'est plus question de « laisser-faire[2] ». Il est urgent de proposer des solutions théoriques et pratiques aux multiples problèmes de qualité des données (voir Tableau 1 - ).

L'objet de ce dossier est de présenter une synthèse des solutions proposées et les perspectives de recherche actuelles pour le contrôle et la gestion de la qualité des données dans les bases et entrepôts de données.

## 2 La gestion de la qualité des données à la convergence de plusieurs disciplines

Dans la pratique, les premières stratégies d'amélioration de la qualité des données ont été mises en œuvre depuis une dizaine d'années par les entreprises soucieuses des pertes occasionnées par les décisions prises à partir d'informations erronées. Dans ce contexte, le contrôle et la gestion de la qualité des données reposent sur des techniques d'***audit*** et de ***suivi de données*** (incluant, par exemple, le recensement des différents types d'erreurs, l'élaboration de méthodes pour les détecter, l'estimation de leur fréquence d'occurrence dans la base, etc.). Ces deux techniques ainsi qu'un cas pratique vous seront présentées en détail dans la section suivante. Une première difficulté est l'absence de consensus sur la définition même de ce que représente la qualité des données. Si tout le monde s'accorde sur le fait que la qualité d'une donnée peut se décomposer en un certain nombre de dimensions, critères, facteurs, éléments ou attributs (les uns, subjectifs nécessitant un jugement et une expertise humaine et les autres, quantifiables et pouvant se mesurer par une grande variété de techniques et de métriques), aucune définition ne fait l'unanimité. Et plus de deux cents dimensions ont été recensées dans la littérature [34].

---

[1] de l'ordre de 611 milliards de dollars par an pour l'économie américaine selon un rapport du TDWI (*The Data Warehousing Institute*) en 2002.

[2] c'est-à-dire, utiliser aveuglément les données sans en connaître la qualité et les laisser se dégrader.

| ÉTAPES DE TRAITEMENT | SOURCES DE PROBLEMES DE QUALITE DES DONNEES |
|---|---|
| **Création des données** | Entrée manuelle : absence de vérifications systématiques des formulaires de saisie<br>Entrée automatique : problèmes de capture OCR, de reconnaissance de la parole<br>Incomplétude, absence de normalisation ou inadéquation de la modélisation conceptuelle des données : attributs peu structurés, absence de contraintes d'intégrité pour maintenir la cohérence des données<br>Entrée de doublons<br>Approximations<br>Contraintes matérielles ou logicielles<br>Erreurs de mesure<br>Corruption des données : faille de sécurité physique et logique des données |
| **Collecte / import des données** | Destruction ou mutilation d'information par des prétraitements inappropriés<br>Perte de données : *buffer overflows*, problèmes de transmission<br>Absence de vérification dans les procédures d'import massif<br>Introduction d'erreurs par les programmes de conversion de données |
| **Stockage des données** | Absence de méta-données<br>Absence de mise à jour et de rafraîchissement des données obsolètes ou répliquées<br>Modèles et structures de données inappropriés, spécifications incomplètes ou évolution des besoins dans l'analyse et conception du système<br>Modifications *ad hoc*<br>Contraintes matérielles ou logicielles |
| **Intégration des données** | Problèmes d'intégration de multiples sources de données ayant des niveaux de qualité et d'agrégation divers<br>Problèmes de synchronisation temporelle<br>Systèmes de données non conventionnels<br>Facteurs sociologiques conduisant à des problèmes d'interprétations et d'intégration des données<br>Jointures *ad hoc*<br>Appariements aléatoires<br>Heuristiques d'appariements des données inappropriées |
| **Recherche et analyse des données** | Erreur humaine<br>Contraintes liées à la complexité de calcul<br>Contraintes logicielles, incompatibilité<br>Problèmes de passage à l'échelle, de performances et de confiance dans les résultats<br>Approximations dues aux techniques de réduction des grandes dimensions<br>Utilisation de boîtes noires pour l'analyse<br>Attachement à une famille de modèles statistiques<br>Expertise insuffisante d'un domaine<br>Manque de familiarité avec les données |

**Tableau 1 - Des problèmes de qualité des données**

A titre indicatif, le Tableau 2 présente quelques-unes des principales dimensions considérées dans la plupart des applications. L'angle d'approche retenu consiste à aborder la qualité des données en considérant : *i)* la qualité de la représentation des données dans le système (au niveau du modèle conceptuel), *ii)* la qualité de la gestion des données par le système (au niveau des processus de traitement) et enfin, *iii)* la qualité des données (au niveau des instances et des valeurs). Dans la suite de ce dossier, nous ne nous intéresserons qu'aux dimensions mesurables par des procédures automatiques.

| Niveau | Dimensions | Descriptif |
|---|---|---|
| ***Qualité du modèle conceptuel des données*** | Lisibilité | Caractère qui confère au modèle conceptuel une facilité de lecture par sa clarté et sa minimalité (degré de factorisation) |
| | Complétude | Caractère qui confère au modèle conceptuel une couverture de l'ensemble des besoins |
| | Expressivité | Caractère qui confère au modèle conceptuel une richesse descriptive pour représenter naturellement les besoins et la réalité |
| | Correction | Caractère qui confère au modèle conceptuel une conformité par rapport aux spécifications |
| | Traçabilité | Documentation détaillée et historique de la conception et de l'évolution du modèle conceptuel des données |
| | Simplicité | Caractère qui restreint le modèle conceptuel a un ensemble minimal d'éléments nécessaires |
| ***Qualité des processus de traitement des données*** | Sécurité | Ensemble des facteurs portant sur l'aptitude du système à préserver les données de toute manipulation malveillante ou hasardeuse |
| | Fiabilité | Ensemble des facteurs portant sur l'aptitude du système à maintenir les données dans des conditions précises et pendant une période déterminée (tolérance aux pannes et récupération des données) |
| | Accessibilité | Ensemble des facteurs sur l'aptitude du système à rendre les données consultables et manipulables dans des temps adéquats |
| | Disponibilité | Ensemble des facteurs portant sur l'effort nécessaire pour l'utilisation des données et sur l'évaluation individuelle de cette utilisation par un ensemble défini ou implicite d'utilisateurs |
| | Maintenabilité | Ensemble des facteurs portant sur l'effort nécessaire pour faire des modifications sur les données et sur leur schéma. |
| | Interopérabilité | Ensemble des facteurs portant sur l'aptitude du système à permettre et faciliter l'échange des données |
| | Confidentialité | Ensemble des facteurs portant sur l'aptitude du système à assurer que les données ne soient accessibles que par ceux dont l'accès est autorisé |
| ***Qualité des instances ou valeurs des données*** | Complétude | Quantité de valeurs renseignées |
| | Cohérence | Quantité de valeurs satisfaisant l'ensemble des contraintes ou règles de gestion définies |
| | Exactitude | Quantité de valeurs correctes et sans erreur |
| | Fraîcheur | Ensemble des facteurs qui capturent le caractère récent et le caractère d'actualité d'une donnée entre l'instant où elle a été extraite ou créée dans la base et l'instant où elle est présentée à l'utilisateur |

**Tableau 2 - Principales dimensions de la qualité : qualité du modèle, des processus et des instances de données**

Comme nous le présentons ci-après, différentes communautés de recherche en Bases de Données, Statistiques, et Gestion de processus ont proposé plusieurs approches complémentaires pour évaluer chaque dimension par diverses métriques, et contrôler certains aspects de la qualité des données.

## 2.1 Les bases de données

Classiquement dans le domaine des bases de données, la gestion de la qualité des données consistait à assurer : *i)* l'exactitude syntaxique des données (par exemple, par la vérification de contraintes qui empêchent, en cas de violation, le stockage des données suspectées dans la base) et, *ii)* l'exactitude sémantique (c'est-à-dire, la

conformité du modèle et des données pour qu'ils reflètent véritablement le monde réel modélisé). Cette approche traditionnelle et restreinte qui reposait principalement sur les techniques telles que les contraintes d'intégrité et la gestion des conflits d'accès a été étendue pour permettre la gestion de la qualité dans les bases de données en particulier pour diriger l'intégration des données entre systèmes de gestion de bases de données hétérogènes [23][22]. D'autres travaux de recherche en Bases de Données se sont focalisés sur les données imparfaites, incomplètes, incohérentes, imprécises ou incertaines. L'incomplétude dans les bases de données est un domaine de recherche intensive où l'étude des différents types de valeurs nulles a occupé, très tôt, une place privilégiée, notamment pour généraliser l'algèbre relationnelle afin de prendre en compte les informations incomplètes [13] ou encore pour répondre au problème de la complétude du résultat d'une requête sur des relations partiellement complètes [20][17]. La théorie des probabilités, la théorie de Dempster-Schafer et la théorie des ensembles flous (ou théorie des possibilités) sont les principaux formalismes rencontrés pour modéliser l'incertain dans les bases de données [25][37]. Plus récemment, l'extension des langages de requêtes pour permettre le nettoyage des données a également suscité bon nombre de travaux de recherche dans le domaine [29][10][30][33][5] et en particulier, sur les techniques d'appariement d'enregistrements (*record linkage*) [12] et la détection de doublons par jointure approximative [16][7][24].

## 2.2 Les statistiques et la fouille de données

Depuis plus d'un siècle, un large éventail de méthodes d'analyses statistiques a été proposé pour tester des jeux de données selon des hypothèses sur des paramètres particuliers ou pour estimer la validité de certains modèles de probabilité. Les valeurs aberrantes ont toujours été des sources de contamination de l'information obtenue à partir des données brutes et elles ont suscité de nombreux travaux pour interpréter, caractériser et traiter ces valeurs anormales, soit en les rejetant systématiquement, soit en adoptant des méthodes dites robustes qui minimisent leur impact sur les analyses statistiques et qui, dans une certaine mesure, *s'adaptent* aux valeurs aberrantes ou les *accommodent* [1]. Nous présenterons dans ce dossier un panorama des méthodes de détection et de traitement des valeurs aberrantes. Les méthodes statistiques pour l'inférence sur les données manquantes [32][18] seront également présentées.

## 2.3 La gestion de processus

La gestion de processus a pour objectif de prévenir l'introduction de données erronées dans un système d'information en examinant chaque étape du traitement des données depuis leur acquisition jusqu'à leur usage. On entend par « processus » toute la chaîne de traitements et d'opérations de la création des données à leur destruction, en passant par des modifications de leurs valeurs. Les méthodes utilisées pour déceler les causes profondes des erreurs dans la chaîne de traitement des informations sont principalement l'audit et le suivi de données. A partir de l'étude d'un audit sur les données d'une base, des gabarits d'erreurs peuvent être déterminés et leurs sources profondes systématiquement éliminées. Les principales étapes de la gestion des processus sont les suivantes :

**Etapes de la gestion des processus pour améliorer la qualité des données**

1 - Désigner un propriétaire des processus (traitements sur les données et flux d'informations) et une équipe de responsables
2 - Décrire les processus et comprendre les besoins des utilisateurs et décideurs en terme de qualité des données
3 - Etablir un système de mesures pour l'audit et le suivi des données
4 - Etablir des contrôles statistiques et vérifier la conformité aux exigences
5 - Identifier les opportunités d'amélioration
6 - Sélectionner les opportunités
7 - Réaliser et maintenir les améliorations

Nous verrons dans la section suivante la mise en œuvre pratique des techniques de gestion de processus (par audit et suivi de données) afin d'améliorer la qualité des données d'un entrepôt de données.

# 3 Approches générales et cas pratique pour détecter et corriger les problèmes de qualité des données

Comme le représente la Figure 1, on peut classer la plupart des travaux abordant la problématique de la qualité des données selon quatre grands types d'approches complémentaires.

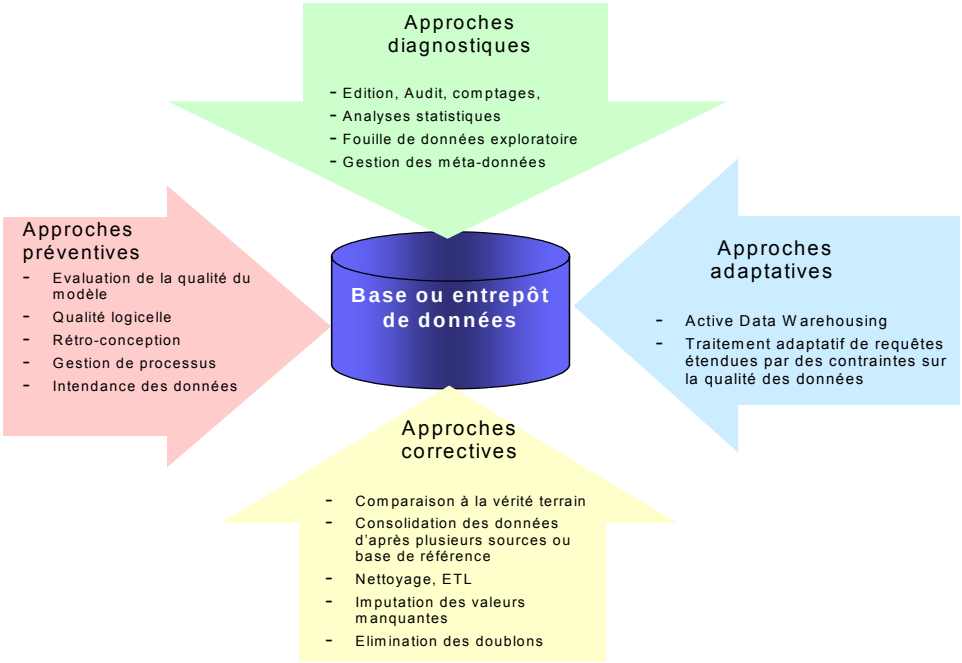



Figure 1 - Panorama des approches pour l'évaluation et le contrôle de la qualité des données

1- Les approches préventives centrée sur l'ingénierie des systèmes d'information et le contrôle des processus avec des techniques permettant

d'évaluer la qualité des modèles conceptuels, la qualité des développements logiciels et celle des processus employés pour le traitement des données,

2- Les approches diagnostiques centrées sur des méthodes statistiques, d'analyse et de fouille de données exploratoire permettant de détecter des anomalies sur les données,
3- Les approches correctives centrées sur des techniques de nettoyage et de consolidation de données et utilisant des langages de manipulation des données étendus et des outils d'extraction et de transformation de données (*ETL – Extraction-Transformation-Loading*)
4- Les approches adaptatives ou actives appliquées généralement lors de la médiation ou de l'intégration des données : elles sont centrées sur l'adaptation des traitements (requêtes ou opérations de nettoyage sur les données) de telle façon que ceux-ci incluent à l'exécution en temps-réel la vérification de contraintes sur la qualité des données.

Parmi les nombreuses techniques de détection et de correction des problèmes de qualité des données, nous présenterons, dans la suite de cette section, celles les plus communément employées dans la pratique et dont les coûts respectifs sont estimés en Figure 2 - : 1) la vérification d'après la vérité-terrain ou d'après une source de données de référence, 2) l'audit des données 3) le suivi de données et enfin, 4) le nettoyage des données.

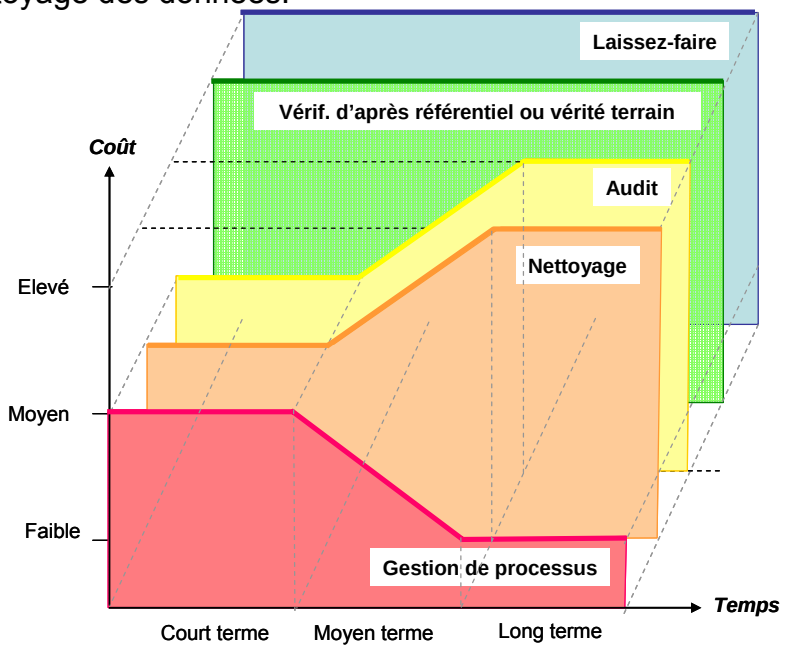



Figure 2 - Coût approché des approches incluant le coût induit par la non-correction des erreurs

1- La première technique consiste à comparer les valeurs de données avec leur contrepartie dans le monde réel (*vérification d'après la vérité-terrain*). Cette méthode est très coûteuse en temps et en moyen et, selon les domaines d'application, difficilement réalisable du fait que la contrepartie réelle est inaccessible ou trop complexe. Une seconde approche appelée *consolidation* met en oeuvre la comparaison de deux bases de données ou plus. Les données pertinentes de la base à inspecter sont comparées à leur contrepartie dans l'autre base : les données identiques sont considérées correctes, celles qui ne le sont pas sont signalées pour investigation et correction éventuelle. Dans ce dernier

cas, la difficulté réside dans la détermination de la valeur correcte (l'une et l'autre donnée pouvant être fausses). La méthode utilisée est alors l'imputation : si l'une des deux valeurs est incorrecte, on assumera que l'autre valeur est correcte, même si ce raisonnement reste risqué. Les inconvénients majeurs à ces deux premières approches pour la détection des erreurs et leur correction sont les suivants :

- il n'y a aucune garantie que les données identiques des différentes bases soient correctes. Les données utilisées par comparaison pour détecter les données erronées dans la base à inspecter peuvent être fausses, rendant la recherche d'erreurs difficile,
- cette méthode n'empêche en rien l'introduction de nouvelles erreurs sur les données.

Classiquement employé dans le domaine des systèmes d'informations géographiques (SIG), la comparaison par rapport à la vérité-terrain (appelé terrain nominal) et la consolidation permettent de réaliser des matrices de confusion entre les données de la base à inspecter et les jeux de données de contrôle.

2- *L'audit des données* met en oeuvre des programmes chargés de vérifier si les valeurs de données satisfont des contraintes de plusieurs types : distribution de probabilités, cohérence par rapport à des règles logiques, règles de calcul, règles spécifiques à l'application ou contraintes statistiques ou d'intégrité. Ces contraintes interviennent aux différents niveaux de la base (valeur, attribut, n-uplet, relation ou collections de chaque). L'avantage de l'audit des données est sa simplicité de mise en oeuvre par rapport aux deux méthodes précédentes de comparaison. Elle peut se concevoir en même temps que le modèle conceptuel des données et peut utiliser différents outils diagnostiques d'analyse de données. Cependant, elle ne permet pas d'améliorations prolongées de la qualité des données. L'édition de données vise l'intégrité, c'est-à-dire la conformité à des règles préalablement définies, mais elle ne garantie en rien l'exactitude des données.

**Audit des données**

- Définition du périmètre de l'audit dans la base de données, selon les dimensions de qualité à considérer et pour des utilisateurs-clés identifiés
- Identification des segments de données à analyser (par exemple, données client, Grand-compte, PME, etc.)
- Choix d'un ensemble représentatif de données (par exemple, par zone géographique)
- Analyse du dictionnaire de données (par exemple, le nom des attributs, type, domaine, taux de remplissage, etc.)
- Enumération des contraintes : par exemple, unicité de clés pour les n-uplets d'une table, respect des contraintes d'intégrité, respect de règles syntaxiques dans les valeurs de certains attributs (tel que le numéro de Sécurité Sociale), respect du zonage géographique (défini par exemple comme une règle de cohérence entre la ville et le code postal), etc.
- Multiples comptages : par exemple, taux d'informations non renseignées, taux d'anomalies de zonage, taux de données ne respectant pas chaque contrainte, détection de doublons, vérification de la cohérence entre la civilité et le prénom), normalisation des adresses, taux de NPAI (*i.e.*, n'habite pas à l'adresse indiquée),

vérification syntaxique du numéro de téléphone, taux de faux téléphones, taux de fax erronés, NPAI d'e-mailing, etc.
- Calculs croisés : par exemple, taux d'individus avec même email, même nom, même adresse, même téléphone, etc.
- Usage de référentiels : par exemple, un dictionnaire des prénoms, la base SIRET, etc. ou du référentiel RNVP (Restructuration, Normalisation et Validation postale). La normalisation des adresses a pour objectif principal de pouvoir bénéficier de tarifs postaux intéressants (TS3), de rectifier les adresses erronées (par exemple, code postal ou rue), et permettre par la suite la détection des doublons.

3- Le principe du *suivi de données* est d'échantillonner les enregistrements lorsqu'ils entrent dans un premier processus de traitement et de les suivre à travers chaque sous-processus jusqu'à leur entrée dans la base de données. Les modifications réalisées sur les enregistrements au fur et à mesure qu'ils poursuivent leur traitement, sont utilisées pour développer des standards de corrections en tirant partie de la redondance des données.

**Suivi de données**

- Echantillonnage aléatoire des enregistrements entrant dans le premier processus de traitement des données participant à la chaîne de traitements des informations et estampillage temporel des enregistrements suivis
- Suivi des échantillons au cours de leur progression dans chaque processus en lecture/écriture (transaction) dans la base de données
- Archivage des entrées sorties de processus avec les estampilles temporelles (début/fin de transactions)
- Identification des défauts et erreurs produits au cours de chaque processus et dans l'ensemble de la chaîne de traitement
- A intervalles de temps appropriés, synthèse de la progression des enregistrements échantillonnés et estampillés sous forme de graphes de Pareto, p-graphe, S-graphe et $\overline{\chi}$-graphe, graphe de décision, organigramme interprocessus, ou densité statistique par processus.

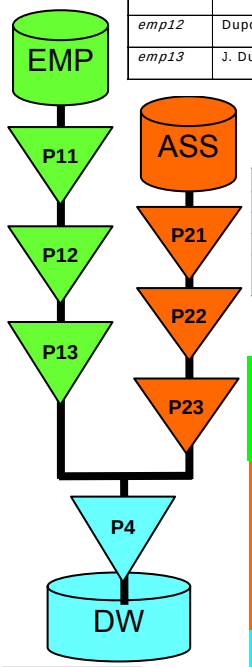
4- Le processus de *nettoyage des données* se compose d'un ensemble de transformations qui vise à normaliser les formats de données et à détecter les paires d'enregistrements qui se rapportent le plus probablement au même objet. Cette étape d'élimination des doublons est appliquée si des données approximativement redondantes sont trouvées et un appariement multi-table calcule des jointures approximatives entre des données distinctes mais similaires ce qui permet leur consolidation. Dans la section 6.1, nous décrirons plus en détail les opérations de transformations possibles.

Exemple : Cas pratique de l'intégration de données issues de deux bases au sein d'un entrepôt

Considérons deux bases de données relationnelles, notées EMP et ASS qui alimentent en données un entrepôt, noté DW, dont les schémas respectifs sont les

suivants :
***EMP***(*EMP-ID1, Nom_Emp, Dept, Salaire/An, Adresse, Tel, Email)*
***ASS***(*ASS_NUM, Nom, Prénom, Num_SS, Date_Naiss, Sexe, Num_Rue, Voie, Code_Postal, Ville)*
*DW(ID, Nom-Ep,Nom-JF, Prénom, Dept, Age, Revenu_Hebdo, Num_Rue, Voie, CP, Ville, Tel, mail)*

Un exemple de chaîne de traitements (processus) réalisés sur les données de EMP et ASS avant leur entrée dans l'entrepôt DW est présenté dans la Figure 3 - .




EMP

| EMP-ID | Nom_Emp | Dept | Salaire/An | Adresse | Tel | Email |
|---|---|---|---|---|---|---|
| emp11 | Dupont Laurent | SRH | 22 000 | 10 bd Malesherbes 75012 PARIS | 01-42-66-02-80 | L-dupont@hotmail.zb |
| emp12 | Dupont Linda | Informatique | NULL | 10 bd Malesherbes 75008 PARIS | 0142 66 02 80 | dupontl@freesirf.fr |
| emp13 | J. Durand | RH | 65 000 | 15, Avenue du Général de Gaulle | 999 999 9999 | NULL |

ASS

| ASS_NUM | Nom | Prénom | Num_SS | Date_Naiss | Sexe | Num_Rue | Voie | Code_Postal | Ville |
|---|---|---|---|---|---|---|---|---|---|
| ass21 | Durand | Linda | 273121309026-57 | 18/12/1973 | F | 10 | Malesherbes | NULL | Paris |
| ass22 | Dupont | L. | 273121309026-57 | 18/11/1973 | F | 10 | Malesbehres | NULL | Paris |
| ass23 | Dupont Laurent 17/05/1970 | NULL | 170050566045-23 | NULL | F | 10 | Malesherbes | NULL | 75008 |
| ass24 | Durand | Jean | 170050566045-25 | 16/13/1970 | M | NULL | Av Gal Gaulle, Paris | 750016 | Lyon |

**P11:** fractionner(Nom_Emp -> Nom-Ep, Prénom) **ET** formater(Tel)
**P12:** verifier(Tel) **ET** verifier(Email) **ET** formater(Dept) **ET** calcul (Revenu_Hebdo)
**P13:** fractionner(Adresse -> Num_Rue, Voie, Code_Postal, Ville)
**P21:** formater(Nom) et verifer(Num_SS, Sexe, Date_Naiss) **ET** verifier(Code_Postal, Ville)
**P22:** dédoublonner(Num_SS)
**P23:** rectifier(Code_Postal) **ET** rectifier(Voie)
**P4:** jointure_approximative(EMP, ASS)

DW

| ID | Nom-Ep | Nom-JF | Prénom | Dept | Age | Revenu_Hebdo | Num_Rue | Voie | CP | Ville | Tel | mail |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID1 | Dupont | NULL | Laurent | SRH | 36 | 393 | 10 | bd Malesherbes | 75008 | PARIS | 01 42 66 02 80 | L-dupont@hotmail.fr |
| ID2 | Durand | NULL | Jean | SRH | 36 | 1160 | 15 | av Gal de Gaulle | 75008 | PARIS | NULL | NULL |
| ID3 | Dupont | Durand | Linda | INF | 33 | NULL | 10 | bd Malesherbes | 75008 | PARIS | 01 42 66 02 80 | dupontl@freesurf.fr |

Figure 3 - Exemple d'intégration de données au sein d'un entrepôt

Un audit sur les sources de données EMP et ASS ainsi que sur l'entrepôt DW fournirait par exemple le tableau d'indicateurs présenté ci-après.

| BASE AUDITEE | Taux de remplissage | Unicité de clé | Taux d'anomalies de zonage | Taux d'incohérences de civilité | Taux d'incohérences syntaxiques | Taux de doublons | Taux de NPAI | Taux de conformité des adresses | Taux de NPAI e-mailing |
|---|---|---|---|---|---|---|---|---|---|
| EMP | 70% | 89% | 37% | NA | 56% | 30% | 26% | 23% | 85% |
| ASS | 43% | 73% | 57% | 25% | 42% | 28% | 12% | 75% | NA |
| DW | 78% | 92% | 3% | 15% | 27% | 13% | 8% | 95% | 85% |

**Tableau 3 - Exemple de résultat d'un audit des données sur EMP, ASS et DW**

Le suivi des données dans cet exemple sera réalisé avant l'import des données de EMP et ASS dans l'entrepôt en scrutant les entrées/sorties de chacun des processus (*P11, P12, P13, P21, P22, P23* et *P4*) sur un échantillon d'enregistrements étiquetés et estampillés. Si l'enregistrement *ID3* fait l'objet du suivi, les enregistrements emp12 de EMP et ass21 et ass22 de ASS seront également suivi tout au long de la chaîne de processus comme le présente le Tableau 4 qui renseigne la durée des processus et les valeurs obtenues en sortie.

| Attribut | EMP | | | ASS | | | | DW |
|---|---|---|---|---|---|---|---|---|
| | P11 | P12 | P13 | P21 | | P22 | P23 | P4 |
| Nom_Ep | Dupont | | | Dupont | Durand | Dupont Durand | | Dupont |
| Nom_JF | | | | | | | | Durand |
| Prénom | Linda | | | L. | Linda | Linda | | Linda |
| Dept | | INF | | | | | | INF |
| Age | | | | Verif(Num_SS, Date_Naiss) **NOK** 33 | Verif(Num_SS, Date_Naiss) **OK** 33 | 33 | | 33 |
| Revenu_Hebdo | | NULL | | | | | | NULL |
| Num_Rue | | | 10 | | | | 10 | 10 |
| Voie | | | bd Malesherbes | | | | bd Malesherbes | bd Malesherbes |
| CP | | | 75012 | Verif(Code_Postal, Ville) **NOK** | Verif(Code_Postal, Ville) **NOK** | | 75008 | 75008 |
| Ville | | | PARIS | PARIS | PARIS | PARIS | PARIS | PARIS |
| Tel | 01 42 66 02 80 | Verif(Tel) **OK** | | | | | | |
| Mail | | Verif(Email) **NOK** | | | | | | |
| Estampille en entrée du processus | Mar 28 15:11:05 MET DST 2006 | Mar 28 15:11:06 MET DST 2006 | Mar 28 15:11:05 MET DST 2006 | Mar 28 15:11:05 MET DST 2006 | | Mar 28 15:11:05 MET DST 2006 | Mar 28 15:11:05 MET DST 2006 | Mar 28 15:11:05 MET DST 2006 |
| Estampille en sortie du processus | Mar 28 15:12:07 MET DST 2006 | Mar 28 15:11:07 MET DST 2006 | Mar 28 15:11:07 MET DST 2006 | Mar 28 15:11:07 MET DST 2006 | | Mar 28 15:11:08 MET DST 2006 | Mar 28 15:13:07 MET DST 2006 | Mar 28 15:12:07 MET DST 2006 |

**Tableau 4 - Suivi d'un enregistrement avant import dans DW**

Le suivi d'échantillons de données tout au long des processus de traitement permettra en l'occurrence de tracer par exemple le nombre de modifications pour chaque type d'attribut ou encore des graphes de Pareto tel que celui présenté dans la Figure 4 qui montre les opportunités d'amélioration de la qualité des données sur les attributs les plus fréquemment modifiés en erreur (par exemple l'attribut Revenu_Hebdo). Par la suite une étude plus détaillée sera menée inter-processus précisément sur cet attribut au moyen de p-graphes afin de constater les taux d'erreurs sur l'attribut Revenu_Hebdo sur plusieurs échantillons.

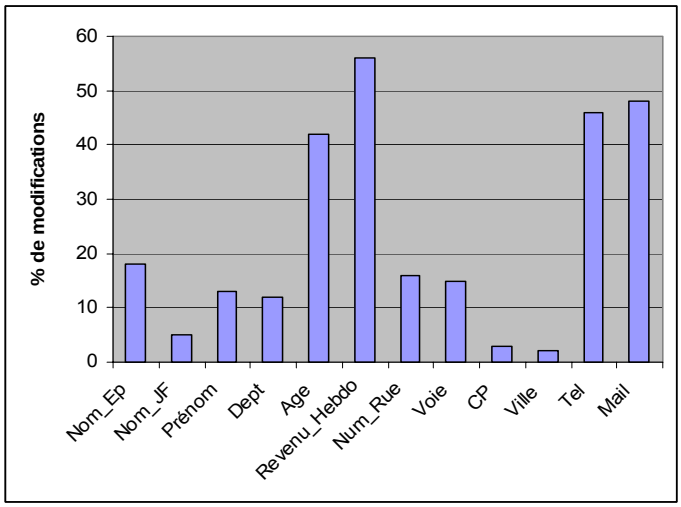



Figure 4 - Graphe de Pareto montrant la proportion de modifications en erreur réalisées sur chaque attribut avant import dans l'entrepôt DW

# 4 Mesurer la qualité du modèle et des données

Évaluer préalablement la qualité des données stockées dans les systèmes d'information, bases et entrepôts de données est essentiel afin de :

– proposer aux utilisateurs des mesures objectives et une expertise critique de la qualité des données stockées,

– permettre à ceux-ci de relativiser la confiance qu'ils pourraient accorder aux données, et leur permettre ainsi, de mieux en adapter leur usage,
– enfin évaluer la validité et l'intérêt des connaissances extraites en assurant les décisions prises à partir des données.

Si l'analyse des données et la prise de décision peuvent être réalisées sur des données inexactes, incomplètes, ambiguës et de qualité médiocre, on peut légitimement s'interroger sur le sens à donner aux résultats de ces analyses et, à juste titre, remettre en cause la qualité des connaissances découvertes à partir des données ainsi que le bien-fondé des décisions prises [4].

## 4.1 Evaluer la qualité d'un modèle conceptuel de données

Dès la conception d'un base de données, il est utile d'analyser la complétude, l'expressivité et la simplicité du modèle conceptuel à l'origine du schéma de la base de données, et ce, principalement pour en évaluer la maintenabilité et prévoir les dérives liées à son usage (par exemple, l'entrée systématique de données erronées (Tel=9999999999) lorsque l'information n'est pas connue au moment de la saisie). Quelques-unes des métriques proposées dans la littérature, notamment issues de l'ouvrage de Piattini *et al*. [28], sont présentées dans le Tableau 5. Ce sont des mesures objectives (basées sur des comptages d'entités/relations) et subjectives de la complexité et des facteurs de qualité d'un modèle conceptuel de type Entité-Association. La validation de ces métriques reste aujourd'hui un problème de recherche ouvert.

| AUTEURS | METRIQUES | VALIDATION | |
|---|---|---|---|
| | | THEORIQUE | EMPIRIQUE |
| Gray *et al.* [11] | Complexité structurelle du modèle : $E=\sum_{i=1}^{n} Ei$ avec $n$ entités $Ei$<br>Complexité pour l'entité $i$ : $Di=Ri*(a*FDA+b*NFDA)$ avec $0 < a <= b$, $Ri$ = nombre d'associations, $FDAi$ = nombre d'attributs en dépendances fonctionnelles, $NFDAi$ = nombre d'attributs sans dépendance fonctionnelle | Non | Non |
| Moody et al. [21] | **Complétude** : <nombre d'items du modèle ne correspondant pas aux spécifications des utilisateurs, nombre de spécifications non représentées dans le modèle, nombre d'items du modèle qui correspondent aux spécifications mais qui sont mal définis, nombre d'incohérences dans la modélisation><br>**Intégrité** : <nombre de contraintes non satisfaites par les données, nombre de contraintes incluses dans le modèle qui ne correspondent pas exactement à la réalité modélisée><br>**Flexibilité :** <nombre d'éléments dans le modèle sujets à modification, coût estimé des modifications, importance stratégique des modifications><br>**Compréhensibilité** : <estimation par l'utilisateur du caractère compréhensible et interprétable du modèle><br>**Correction** : <nombre de violations des conventions du modèle de données, nombre de violations des formes normales, nombre d'instances redondantes dans le modèle><br>**Simplicité** : <nombre d'entités et associations ; somme pondérée ($aN^E + bN^R + cN^A$), où $N^E$ est le nombre d'entités, $N^R$ le nombre d'associations et $N^A$ le nombre d'attributs><br>**Intégration** : <nombre de conflits avec le modèle de données commun, nombre de conflits avec les systèmes existants><br>**Implémentabilité** : <estimation du risque technique, estimation du risque de planification, , estimation du coût de développement, nombre d'éléments du niveau physique inclus dans le modèle> | Non | Non |
| Genero *et al.* in [27] | **NE** : nombre total d'entités dans le modèle ; **NA** : nombre total d'attributs d'entités et d'associations (simples ou composés); **DA** : nombre d'attributs dérivés (i.e., attributs dont la valeur peut être déduite ou calculée) ; **CA** : nombre total d'attributs composés ; **MVA** : nombre total d'attributs multi-valués ; **NR** : nombre total d'associations dans le modèle ; **M:NR** : nombre total d'associations M:N ; **1:NR** : nombre total d'associations 1:N et 1:1 ; **N-AryR** : nombre total d'associations N-aires ; **BinaryR** : nombre total d'associations binaires ; **NIS-AR** : nombre total d'associations IS_A (généralisation/spécialisation) ; **RefR** : nombre total d'associations cycliques ; **RR** : nombre total d'association redondantes | Oui | Partiellement |
| Piattini *et al.* [28] | **RvsE:** rapport entre le nombre d'associations $NR$ et le nombre d'entités $NE$ du modèle $RvsE=\left(\frac{NR}{NR+NE}\right)^2$ avec $NR + NE > 0$.<br>**DA:** rapport entre le nombre d'attributs dérivés $NDA$ et le nombre maximal d'attributs dérivés possibles $NA$ : $DA=\frac{NDA}{NA-1}$ avec $NA > 1$.<br>**CA:** rapport entre le nombre d'attributs composés $NCA$ et le nombre d'attributs total $NA$ : $CA=\frac{NCA}{NA}$ avec $NA > 0$.<br>**RR:** rapport entre le nombre d'associations redondantes $NRR$ et le nombre d'associations $NR$ : $RR=\frac{NRR}{NR-1}$ avec $NR > 1$.<br>**M:NRel:** rapport entre NM:NR, le nombre d'associations M:N sur le nombre d'associations $NR$ : $M:NRel=\frac{NM:NR}{NR}$ avec $NR > 1$.<br>**FLeaf :** mesure de la complexité des hiérarchies (IS_A) : $FLeaf=\frac{NLeaf}{NE}$ avec $NLeaf$ : nombre d'entités filles dans une hiérarchie généralisation/spécialisation et $NE$ nombre d'entités dans chaque hiérarchie, $NE > 0$.<br>**IS_Arel** : nombre moyen de supertypes directs et indirects par entité non racine $ALLSup$ : $IS\_ARel=FLeaf-\frac{RLeaf}{ALLSup}$ | Non | Partiellement |

**Tableau 5 - Métriques utilisées pour évaluer la complexité et la qualité d'un modèle conceptuel de type Entité-Association**

Ces mesures sont des supports quantitatifs permettant de comparer des alternatives de conception et l'identification des problèmes de conception qui ont nécessairement

un impact plus ou moins direct sur la qualité des données stockées.

### 4.2 Mesurer la qualité d'une base de données relationnelle

D'une façon générale, nous modélisons la qualité des données selon le formalisme UML comme le représente la Figure 5. La base de données est une classe abstraite pour laquelle les données sont représentées selon un modèle conceptuel et gérées par des processus. Dès lors que des données, processus ou tout ou partie du modèle de données sont considérés critiques, la qualité de ceux-ci peut être évaluée selon plusieurs dimensions (représentant les différentes facettes de la qualité). Chaque dimension peut être mesurée par une ou plusieurs métriques à un instant donné.

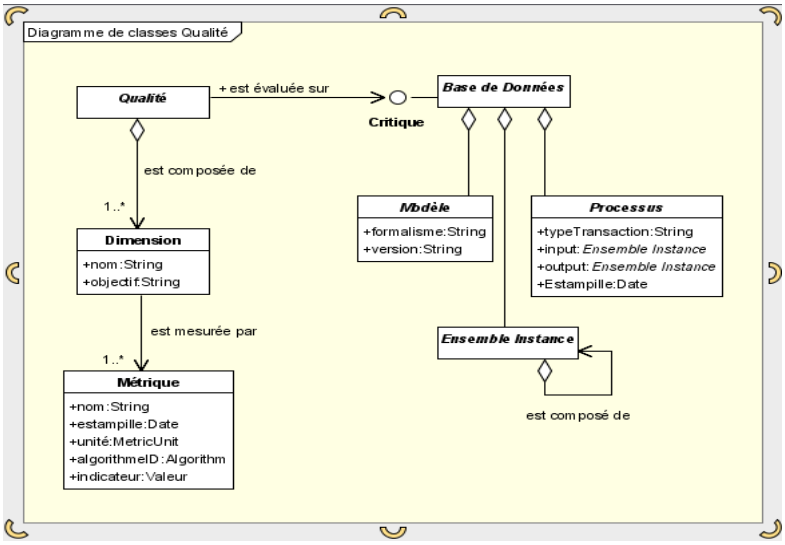



Figure 5 - Méta-modèle décrivant la qualité associée aux données, modèle et processus

## 5 Prévention et diagnostic : techniques de détection des anomalies

Comme nous l'avons évoqué précédemment, la prévention et le diagnostic reposent essentiellement sur la définition et la mise en œuvre préalables de contraintes et de procédures de vérifications automatiques de la cohérence des données.

### 5.1 Vérification de contraintes et gestion des méta-données associées à la qualité

De façon concrète, différents niveaux de contrôle (voir Figure 6) peuvent être implémentés au dessus d'une base ou d'un entrepôt de données : du plus simple comptage à des analyses statistiques sophistiquées. Tout d'abord, dans le contexte d'un base relationnelle, des contraintes peuvent être formulées par des assertions (SQL check) (A.), ensuite, selon les possibilités offertes par le système de gestion de base de données, des *triggers* peuvent être déclarés en SQL se déclenchant automatiquement dès qu'une contrainte n'est plus satisfaite sur les données de la base (B.). Lors d'un requêtage ciblé, des vues avec vérification de contraintes pourront être également déclarées (C.). Un peu plus évoluées car alliant la puissance d'un langage de programmation, des procédures stockées (*stored procedures*) (en PL/SQL par exemple) pourront automatiser des vérifications et analyses sur les données (D.)

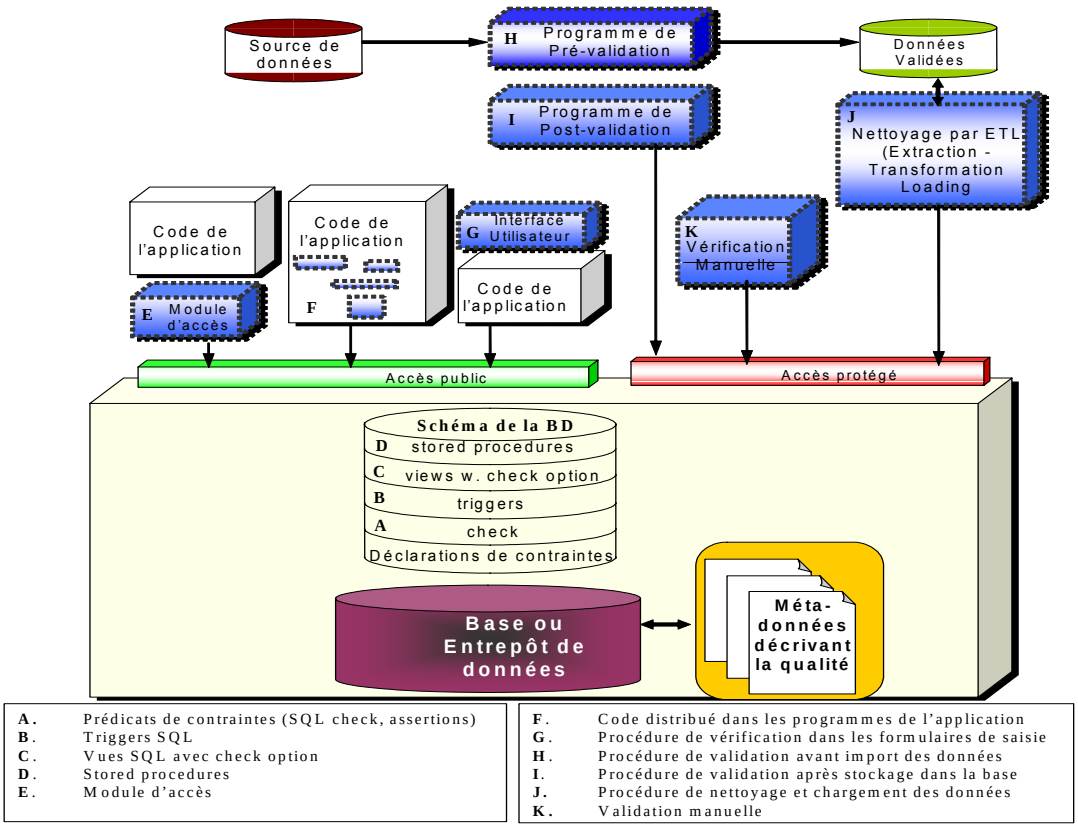



Figure 6 - Différents niveaux de contrôle et de mesure de la qualité des données

S'éloignant du « noyau » du système de gestion de la base de données, plusieurs programmes pourront assurer la vérification des données dès la saisie soit au niveau des modules d'accès (E.), du code de l'application qui permet d'accéder aux données (F.) ou de l'interface utilisateur (G.), la possibilité de valider manuellement les données restant préservée par un accès restreint à l'administrateur de la base ou à « l'intendant » des données (K.). Lors de l'import de données issues d'une source externe, des programmes de pré-validation (H.) et de nettoyage (J.) des données permettront de contrôler et corriger la qualité des données importées. Des programmes de validation *a posteriori* pourront par la suite mettre en œuvre des tests, analyses statistiques pour vérifier les données de la base et procéder éventuellement à des actions correctives (J.). Un historique des contrôles, vérifications de contraintes et résultats de mesure de chaque dimension de la qualité des données doit être stocké en tant que méta-donnée au sein de la base.

## 5.2 Analyse statistique et fouille de données exploratoire

La fouille de données exploratoire (*Exploratory Data Mining - EDM*) [8] est un ensemble de techniques statistiques proposant des résumés et des visualisations pour détecter des problèmes dans un jeu de données avant de réaliser des analyses plus coûteuses. Elle a pour objectifs d'être largement applicable, rapide en temps de réponse, simple à mettre en œuvre et ses résumés sont faciles à stocker, à mettre à jour et à interpréter. Elle révèle des informations qui permettent de faire des hypothèses, notamment sur la distribution des données ou sur les corrélations entre tel ou tel attribut (par exemple, une distribution gaussienne des valeurs d'un attribut *A* qui est lui-même lié de façon linéaire aux attributs *B* et *C* de la base). La fouille de données exploratoire peut être dirigée par les modèles et faciliter l'utilisation de

méthodes paramétriques (modèles log-linéaires, par exemple) ou encore être dirigée par les données. Les résumés seront typiquement des moyennes, écarts-types, médianes ou autres quantiles sur plusieurs échantillons de l'ensemble des données. Ils permettent notamment de caractériser le centre de la distribution des valeurs d'un attribut représentatif de la population, quantifier l'étendue de la dispersion des valeurs de l'attribut autour du centre ou, encore, décrire la dispersion (forme, densité, symétrie, etc.). Concernant les techniques d'analyse sur des données manquantes [32], la méthode d'imputation par régression logistique ou celle basée sur le maximum de vraisemblance décrite par Little et Rubin [18] sont très utilisées. D'autres méthodes telles que celle de Monte Carlo par chaînes de Markov (*MCMC*) permettent de simuler des données en leur supposant une distribution normale multivariée. D'autres références traitant des données manquantes sont décrites dans le tutorial de Pearson [26]. Dans la section 7.3, nous détaillerons le cas des données aberrantes ou valeurs isolées.

# 6 Correction : nettoyage des données

## 6.1 Extension des langages de manipulation des données pour l'extraction et la transformation

Le nettoyage de données par transformation ou ETL (*Extraction-Transformation-Loading*) fait partie des stratégies d'amélioration de la qualité des données [29][8]. Elle consiste à choisir et appliquer des transformations sur des jeux de données pour résoudre différents problèmes de format et d'incohérences, soit au sein d'une source, soit entre deux sources de données à intégrer. Comme a pu l'illustrer notre exemple précédent, les principales opérations de transformation sont énumérées dans la Figure 7 d'après [30].

| TRANSFORMATION | DÉFINITION FORMELLE |
|---|---|
| Formatage | $\phi(R, i, f) = \{(a_1,\dots, a_{i-1}, a_{i+1},\dots, a_n, f(a_i))\mid (a_1,\dots, a_n) \in R\}$ |
| Ajout | $\alpha(R, x) = \{(a_1,\dots, a_n, x)\mid (a_1,\dots, a_n) \in R\}$ |
| Suppression | $\pi(R, i) = \{(a_1,\dots, a_{i-1}, a_{i+1},\dots, a_n)\mid (a_1,\dots, a_n) \in R\}$ |
| Copie | $\kappa((a_1,\dots, a_n), i) = \{(a_1,\dots, a_n, a_i)\mid (a_1,\dots, a_n) \in R\}$ |
| Fusion | $\mu((a_1,\dots, a_n,), i, j, glue) = \{(a_1,\dots, a_{i-1}, a_{i+1}, \dots, a_{j-1}, a_{j+1},\dots, a_n, a_i \oplus \text{glue} \oplus a_j)\mid (a_1,\dots, a_n) \in R\}$ |
| Division | $\delta((a_1,\dots, a_n), i, \text{pred}) = \{(a_1,\dots, a_{i-1}, a_{i+1}, \dots, a_n, a_i, \text{null})\mid (a_1,\dots, a_n) \in R \wedge \text{pred}(a_i)\} \cup \{(a_1,\dots, a_{i-1}, a_{i+1}, \dots, a_n, \text{null}, a_i)\mid (a_1,\dots, a_n) \in R \wedge \neg\text{pred}(a_i)\}$ |
| Partage | $\omega((a_1,\dots, a_n,), i, splitter) = \{(a_1,\dots, a_{i-1}, a_{i+1},\dots, a_n, \text{left}(a_i, splitter), \text{right}(a_i, \text{splitter}))\mid (a_1,\dots, a_n) \in R\}$ |
| Repliement | $\lambda(R, i_1, i_2,\dots, i_k) = \{(a_1,\dots, a_{i1-1}, a_{i1+1},\dots, a_{i2-1}, a_{i2+1},\dots, a_{ik-1}, a_{ik+1},\dots, a_n, a_{il})\mid (a_1,\dots, a_n) \in R \wedge 1 <= l <= k\}$ |
| Sélection | $\sigma(R, \text{pred}) = \{(a_1,\dots, a_n)\mid (a_1,\dots, a_n) \in R \wedge \text{pred}((a_1,\dots, a_n))\}$ |

Notation : $R$ est une relation avec $n$ attributs, $i$ et $j$ sont les indices des attributs ; $a_i$ représente la valeur d'un attribut pour un tuple donné; $x$ et *glue* sont des valeurs, $f$ est une fonction de transformation d'une valeur en une autre ; $x \oplus y$ concatène $x$ et $y$ ; splitter est une position dans une chaîne de caractères ou une expression régulière ; left($x$, splitter) est la partie gauche de $x$ après avoir partagé la chaîne de caractères à la position indiquée par la variable splitter ; pred est une fonction retournant un booléen.

**Exemple de transformation des données d'une table:**

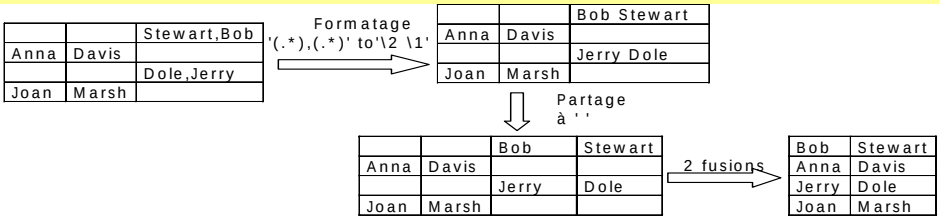



Figure 7 - Opérations de transformation utilisées pour le nettoyage des données

Les problèmes candidats au nettoyage peuvent être répartis en problèmes mono-sources et multi-sources, aux niveaux du schéma ou des instances. Nous verrons dans les tableaux d'exemples suivants que la classification d'une erreur dépend essentiellement des contraintes qui auront pu être définies. Les tableaux d'exemples reprennent les sources EMP et ASS de notre exemple et supposent un ensemble de contraintes implicites.

Le Tableau 6 présente des erreurs détectées au moyen de contrôles de cohérence.

| **Exemples détaillés des problèmes de violation de contraintes d'intégrité** | | | |
|---|---|---|---|
| **Impact/Problème** | | **Données** | **Descriptif** |
| **Attribut** | Valeurs improbables | ass24=(Date_Naiss=16/13/1970) | Valeurs non comprises dans l'intervalle admis |
| **Enregistrement** | Violation de contraintes de cohérence entre attributs | ass24=(Num_SS=1700505056045-23, Sexe=F) | La contrainte entre le 1er chiffre de Num_SS =1 et Sexe=M devrait être satisfaite |
| **Type d'enregistrement** | Violation de contraintes d'unicité | ass21=(Nom="Durand", Num_SS="2731213095026-57")<br>ass22=(Nom="Dupond", Num_SS ="2731213095026-57") | Unicité pour le n° de sécurité sociale non respectée |
| **Source** | Violation de contraintes d'intégrité référentielles | emp13=(Nom_Emp="J. Durand", Dept=RH) | Valeur référencée Dept (RH) non définie. |

**Tableau 6 - Exemples de violation de contraintes au niveau du schéma d'une source**

Le Tableau 7 présente des exemples d'erreurs au niveau instance. La détection de ces erreurs peut être effectuée avec : *i)* des contrôles de cohérence non référencés comme contraintes d'intégrité de la base, *ii)* des tests de vraisemblance ou encore, *iii)* au moyen de critères empiriques établis lors de la phase d'analyse des données.

| **Exemples de problèmes au niveau instance pour une source** | | | |
|---|---|---|---|
| **Impact/Problème** | | **Données** | **Descriptif** |
| **Attribut** | Valeurs manquantes | emp13=(Tel=9999999999) | Valeurs non disponibles au moment de la saisie |
| | Erreurs typographiques | ass22(Voie=" Malesbehres ") | typos, erreurs phonétiques |
| | abréviations | emp13=(Dept="RH")<br>ass24(Voie=" Av Gal Gaulle, Paris") | |
| | Valeurs imbriquées | ass23=(Nom=" Dupont Laurent 17/05/1970") | valeurs multiples saisies dans un attribut (au format texte libre) |
| | Erreur d'attributs | ass23=(Ville="75008") | |
| **Enregistrement** | Violation de dépendances entre attributs | ass24=(Ville="Lyon", Code_Postal=700016) | Les valeurs de Ville et Code_Postal doivent correspondre. |
| **Type d'enregistrement Enregistrement** | Transpositions | emp11=(Emp_Nom="J. Durand" ) et emp13=(Emp_Nom="Dupont Laurent") | Problème lié au format texte libre |
| | Doublons | ass21=(Nom="Durand Linda", ...);<br>ass22=( Nom ="Dupont L.", ...) | Même assurée entrée deux fois sous deux noms différents mais au même numéro de SS |
| | Enregistrements contradictoires | ass21=(Nom="Durand Linda", Date_Naiss=18/12/1973);<br>ass22=( Nom ="Dupont L.", Date_Naiss=18/11/1973); | Deux dates de naissances sont proposées pour Linda Durand |
| **Source** | Mauvais référencement | emp13=(Nom_Emp="Dupont Linda", Dept=Info); | Le département référencé existe mais il est faux pour Linda Dupont |

**Tableau 7 - Exemples de violation de contraintes au niveau instance d'une source**

Les problèmes multi-sources au niveau schéma peuvent se scinder en deux catégories : les conflits de noms et les conflits de structure telles que :

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

Les problèmes multi-sources au niveau instance peuvent être dus à des représentations différentes des données, à des différences d'agrégation, à l'évolution des usages de saisie et de description au cours du temps. La résolution de ces problèmes implique l'intégration des deux schémas, ainsi que le nettoyage de chaque source. Le Tableau 8 montre la solution adoptée et stockée dans l'entrepôt DW après un processus de nettoyage des données issues de EMP et ASS.

| *ID* | *Nom-Ep* | *Nom-JF* | *Prénom* | *Dept* | *Age* | *Revenu_Hebdo* | *Num_Rue* | *Voie* | *CP* | *Ville* | *Tel* | *mail* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *ID1* | Dupont | NULL | Laurent | SRH | 36 | 393 | 10 | bd Malesherbes | 75008 | PARIS | 0142 66 02 80 | L-dupont@hotmail.fr |
| *ID2* | Durand | NULL | Jean | SRH | 36 | 1160 | 15 | av Gal de Gaulle | 75008 | PARIS | 01 45 72 67 30 | NULL |
| *ID3* | Dupont | Durand | Linda | INF | 33 | NULL | 10 | bd Malesherbes | 75008 | PARIS | 0142 66 02 80 | dupontl@freesurf.fr |

**Tableau 8 - Résultat d'un nettoyage de données pour l'intégration dans l'entrepôt DW**

## 6.2 Standardisation des valeurs d'attributs avec les modèles de Markov

Parallèlement aux approches déclaratives de nettoyage employant des langages de manipulation des données, d'autres approches utilisent des modèles de Markov Cachés [7][6]. Un modèle de Markov caché (MMC) (*Hidden Markov Model* HMM) est un graphe probabilisé dans lequel chaque nœud (état) est stable ou transitoire. Un modèle de Markov caché est un double processus stochastique $(X_t,Y_t)$ $1 \leq t \leq T$. La chaîne interne $X_t$ non observable, et la chaîne externe $Y_t$ observable, s'allient pour générer le processus stochastique. La chaîne interne est supposée, pour chaque instant, être dans un état où la fonction correspondante génère une composante de l'observation. La chaîne interne change d'état en suivant une loi de transition. L'observateur ne peut voir que les sorties des fonctions aléatoires associées aux états et ne peut pas observer les états de la chaîne sous-jacente, d'où le terme de Modèles de Markov Cachés. A chaque changement d'état (transition) est associée une distribution de probabilité. Dans le cas de la standardisation d'attributs dont les types de valeurs sont composés des chaînes de caractères plus ou moins structurées et imbriquées (par exemple, les nom et adresse). Des modèles de Markov cachés peuvent être employés pour la standardisation après avoir été entraînés sur des jeux de données d'apprentissage pour initialiser les probabilités de transition. Dans la Figure 8 - , par exemple, la probabilité que le nom d'une personne commence par son prénom est de 35%, puis qu'il soit suivi du nom de famille a une probabilité conditionnelle de 40%. A partir de ce type de modèle, les données peuvent être étiquetées et réordonnées selon le format final requis.

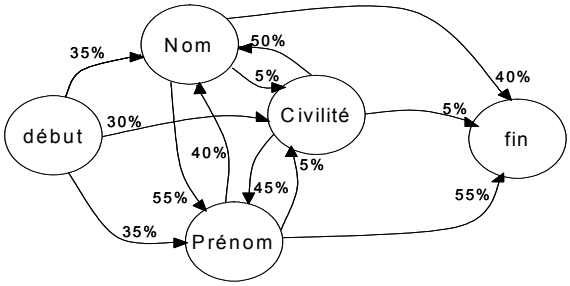



Figure 8 - Modèle de Markov caché pour standardiser les noms

## 6.3 Comparatif des outils actuels pour le nettoyage

Parmi les nombreux prototypes d'extraction et de transformation des données dont ceux figurant dans le Tableau 9 - , Potter's Wheel [30], ARKTOS [33], AJAX [10] et BELLMAN [9] permettent l'extraction de structures et expressions régulières, la transformation de valeurs de données (par application de fonctions de formatage), la transformation de l'ensemble des valeurs de n-uplets (lignes) et d'attributs (colonnes) d'une base de données relationnelle.

| ***Prototype*** | ***Descriptif*** | ***Particularités*** |
|---|---|---|
| Potter's Wheel [30] | Système ETL interactif permettant la détection et correction d'anomalies transformations des données (*add, drop, merge, split, divide, select, fold, format*) et la visualisation des effets des transformations sur les données. Par un mécanisme d'analyse sur les domaines, il permet également de trouver des incohérences dans les valeurs d'attributs. | Interactivité, inférence de la structure |
| Ajax [10] | Extension du langage SQL permettant la déclaration d'opérateurs logiques de transformations (*mapping, view, matching, clustering, merging*) néessaires au nettoyage des données | Séparation logique/physique, 3 algorithmes pour l'appariement |
| Arktos [33] | Méta-modèle unique permettant de couvrir toutes les activités ETL d'un entrepôt (architecture et gestion de la qualité) | Distinction entre le méta-modèle et les niveaux conceptuel, logique et physique |
| Intelliclean [19] | Détection et correction d'anomalies par utilisation d'une base de règles (*duplicate identification, merge, purge, stemming, soundex, stemming, abbreviation*) | Difficulté de passage à l'échelle |
| Telcordia's Tool [5] | Outil paramétrable pour l'appariement des enregistrements (*record linkage*) selon des fonctions de distance et d'appariement. Les règles d'appariement peuvent être générées, testées avec des techniques statistiques ou par apprentissage automatique | Pré-traitement avec élimination des « stop-words » |

| | | |
|---|---|---|
| Febrl [6] | Outil pour standardiser les noms et adresses basé sur les modèles de Markov cachés et pour apparier les données et éliminer les doublons | Dédié initialement au domaine biomédical mais extensible et paramétrable |
| BELLMAN [9] | Suite d'outils ETL et d'analyse statistique de la base de données. | |

**Tableau 9 - Prototypes de recherche dédiés au nettoyage des données**

Jusqu'à présent, le marché de la qualité des données concerne principalement les entreprises possédant de grands systèmes d'information (institutionnels, télécoms, transports, banques, énergéticiens, etc.). Les familles de produits commercialisés répondant à la problématique de la qualité des données sont présentées dans le Tableau 10 - .

| ***Outil (Société)*** | ***Type*** |
|---|---|
| Power Center (Informatica) | Outils de nettoyage par transformation ETL permettant de spécifier et gérer des migrations de données d'un système à un autre proposent des fonctions d'analyse et de transformation et d'alimentation des données. |
| Datastage (Ascential) | |
| SAS (SAS) | |
| ProfileStage/AuditStage (Ascential) | Outils d'audit des données (après une phase d'extraction des méta-données) ne permettant pas la vérification de contraintes métier sur les données |
| Discovery (Avellino) | |
| Evoke (Evoke Software) | |
| IQ8™ and Information Quality Suite® (FirstLogic) | Suite d'outils de nettoyage, normalisation, et consolidation des données avec plusieurs outils spécialisés |
| Normad (Normad1) | Suite progicielle de nettoyage et normalisation des adresses postales proposant également la standardisation et appariement d'enregistrements et la déduplication |

**Tableau 10 - Outils commerciaux pour la normalisation et le nettoyage des données**

La revue des logiciels existants nous montre qu'il n'existe pas encore sur le marché d'approche intégrée qui permette :

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

# 7 Gérer des problèmes spécifiques: les doublons, valeurs manquantes, incomplètes et exceptions

Dans cette section, nous nous intéressons plus particulièrement aux cas des doublons, des valeurs manquantes et des valeurs aberrantes ou isolées en présentant dans une certaine limite à l'exhaustivité, l'ensemble des méthodes et techniques issues des travaux de recherche dans le domaine.

## 7.1 Jointures approximatives et élimination des doublons

Dans le cas d'une intégration de plusieurs sources d'information (en l'occurrence

l'intégration de bases de données relationnelles), il est nécessaire d'associer plusieurs tables au moyen de jointures pour lesquelles souvent on ne dispose pas de clés communes exactes. Lors d'une recherche de doublons sur une seule table, il est nécessaire de procéder par auto-jointure : bien que les clés puissent identifier de façon unique chaque enregistrement de la table, plusieurs enregistrements peuvent pourtant décrire la même réalité : dans notre exemple précédent, les enregistrements *ass21* et *ass22* de la source ASS décrivent la même personne avec deux clés distinctes. Ainsi pour détecter les doublons, la technique de jointure approximative est recommandée [12][16]. D'après notre exemple, il est nécessaire d'apparier les données entre les tables EMP et ASS pour pouvoir renseigner tous les champs de l'entrepôt DW. Les noms et adresses sont décrits de différentes façons (par exemple, « Avenue du Général de Gaulle » ou « av. Gal Gaulle ») et il peut être difficile de faire l'appariement sur les noms ou adresses. Si, en revanche, le numéro de sécurité social ou de téléphone est le même, on pourra supposer qu'il s'agit bien de la même personne, c'est pourquoi il s'avère nécessaire d'abord de standardiser certains attributs (adresses, abréviations, etc.) puis, d'examiner les informations qui corroborent ou non une hypothèse d'appariement sur l'ensemble des attributs disponibles. Parfois très spécifique à l'application, la technique de jointure approximative consiste à regrouper et trier les enregistrements par « paquets » (ou groupes) selon une fonction de hachage sur les valeurs d'un ou plusieurs attributs (par exemple, utilisant les premières lettres ou les consonnes des noms propres). Les enregistrements qui se trouvent dans les mêmes groupes sont candidats à l'appariement et, pour chaque paire de candidats, une distance de similarité est calculée. Seules les paires de plus haut score sont effectivement appariées (ou assimilées à des doublons). La méthode classique de jointure approximative et de détection de doublons est présentée ci-après.

**Méthode générique pour la recherche de doublons**

1. Pré-traitement des données (standardisation des attributs, des abréviations, structuration des adresses, etc.)
2. Choix d'une fonction permettant de réduire l'espace de recherche par :
   - Tri ou hâchage selon une clé
   - Examen par fenêtrage (Multiple) (*Windowing*)
3. Choix d'une fonction de comparaison permettant d'exprimer la distance entre les paires telle que :
   - Identité stricte, distance simple ou complexe
   - Distance pondérée par la fréquence ou dirigée par des règles
   - Distance d'édition, distance de Jaro, Jaro-Winkler
   - Comparaison N-gram, Q-gram
   - Soundex
   - TF-IDF
   - Coefficient de Jaccard, etc.
4. Choix d'un modèle de décision
   - Méthodes probabilistes : avec/sans ensemble d'apprentissage
   - Méthodes basées sur des règles et connaissances du domaine
5. Vérification de l'efficacité de la méthode

Pour des domaines d'attributs textuels, l'appariement des chaînes de caractères (*string matching*) peut être calculée par une distance comptabilisant le nombre d'opérations d'édition (telles que l'ajout, la suppression d'un caractère ou le changement de lettre) nécessaires pour transformer une chaîne de caractères en une autre. Par exemple, « SRH » et « RH » ont une distance d'édition de 1. Les chaînes de caractères dont la distance d'édition est inférieure à un seuil fixé seront alors appariées. L'ensemble des algorithmes d'appariement de chaînes de caractères est détaillé dans [24]. Le Tableau 11 - présente les principales mesures de similarité pouvant être employées pour apparier les chaînes de caractères.

| **Calcul de similarité** | **Définition et principales caractéristiques** |
|---|---|
| Distance de Hamming | Applicable à des champs numériques fixes (N°Sécu, Code Postal) mais ne prend pas en compte les ajout/suppressions de caractères |
| Distance d'édition | Soient s1 et s2 deux chaînes de caractères à apparier, le calcul du coût minimal de conversion de s1 en s2 en cumulant le coût unitaire des opérations d'ajout (A), suppression (S) ou remplacement de caractères (R) est tel que :<br>$Edit(s1,s2) = \min\left(\sum A(s1,s2) + S(s1,s2) + R(s1,s2)\right)$<br>Calcul par programmation dynamique mais complexité quadratique |
| Distance de Jaro | Soient s1 et s2, deux chaînes de caractères de longueur respective L1 et L2, ayant C caractères communs et T transpositions de caractères :<br>Jaro(s1, s2)=(C/L1 +C/L2 +(2C-T)/2C)/3<br>Utilisé pour les chaînes de caractères courtes |
| Distance de Jaro-Winkler | Soit P la longueur du plus long préfixe commun entre s1 et s2<br>Jaro-Winkler(s1,s2)=Jaro(s1,s2)+ max(P,4).(1-Jaro(s1,s2))/10 |
| Distance N-grams | Somme du nombre de caractères communs sur toutes les sous-chaînes de caractères x de longueur N présents dans les chaînes a et b<br>$Ngram(a,b) = \sqrt{\sum_{\forall x} \lvert f_a(x) - f_b(x) \rvert}$ |
| Soundex | Première lettre du mot puis encodage des consommes sur 3 caractères tel que :<br>B,F,P,V -> 1 \| C,G,J,K,Q,S,X,Z ->2 \| D,T -> 3 \| L -> 4 \| M,N -> 5 \| R -> 6<br>Exemple : "John" et "Jan" sont encodé J500 ; "Dupontel" est encodé D134 |
| Indice de Jaccard | Soient deux ensembles de termes S et T<br>$Jaccard(S,T) = \lvert S \cap T \rvert / \lvert S \cup T \rvert$ |
| Mesure TF-IDF | Soit un terme s1 et un document d dans un ensemble de documents D, tf le nombre d'occurrences du terme s1 dans le document d et idf la fraction du nombre de documents dans D sur le nombre de documents contenant s1<br>Tfidf(s1, d, D)=log(tf(s1,d)+1)*log(idf(s1, D))<br>Usage traditionnel en recherche d'information, les termes rares sont rendus plus importants |
| Mesure probabiliste IDF de Fellegi-Sunter | Soient $P_{A\cap B}(s)$ la probabilité que la chaîne de caractère s se retrouve à la fois dans A et dans B (et soit donc identifiée comme doublon) et $P_A(s).P_B(s)$ la probabilité qu'elle ne le soit pas (avec $P_A(s) = P_B(s) = P_{A\cap B}(s)$ )<br>$Fellegi\text{-}Sunter\text{-}IDF(s) = \log(P_{A\cap B}(s)/(P_A(s).P_B(s)) = \log(1/P_A(s))$ |
| Mesure du cosinus | Soient a et b deux attributs, Da et Db les ensembles de termes de chaque attribut, et les scores Tfidf du terme s respectivement dans Da et dans Db :<br>$Cosinus(a,b) = \sum_{t \in Da \cap Db} Tfidf(t,D_a).Tfidf(t,D_b)$ |
| Deux autres distances hybrides | Soient $D_a=\{a_1,a_2,...,a_k\}$ et $D_b=\{b_1,b_2,...,b_p\}$ des ensembles de termes, et s1 et s2 deux chaînes de caractères à comparer, $Sim(a_i,b_j)$ une distance de similarité quelconque choisie parmi les précédentes :<br>$Hybrid1(D_a,D_b) = \frac{1}{k}\sum_{i=1}^{k} \max_{j=1}^{p}\left(Sim(a_i,b_j)\right)$<br>Soit l'ensemble des termes candidats tel que : $Candidats(t,D_a,D_b) = \{w \in D_a \exists v \in D_b, sim(w,v) > t\}$<br>$Hybrid2(w,D_b) = \sum_{w \in Candidats(t,Dd,Db)} Tfidf(w,D_a) * Tfidf(w,D_b) * \max_{v \in Db}(Sim(w,v))$ |
| Similarité floue | Soient Da et Db deux ensembles de termes, le coût de transformation de Da en Db est calculé en utilisant la distance d'édition et la mesure TF-IDF tel que :<br>$Cost(D_a,D_b) = \sum_{S_i \in Da} Tfidf(s_i,D_a)\Big\rvert_0^1 + Tfidf(s_i,D_a) + Edit(s_i,s_j) * Tfidf(s_i,D_a)$<br>$FuzzyMatchSim = 1 - \min\left(\left(Cost(D_a,D_b) / \sum_{S_i \in Da} Tfidf(s_i,D_a)\right),1\right)$ |

**Tableau 11 - Distances de similarité pour comparer les chaînes de caractères et identifier les doublons potentiels**

Le principe général d'une jointure approximative entre deux tables $T1(A_1, A_2,\ldots, A_n)$ $T2(B_1, B_2, \ldots, B_m)$ est le suivant : sur un sous-ensemble du produit cartésien de *T1* et *T2*, si le score de similarité entre les valeurs des ces attributs est supérieur à un seuil S fixé non nul, alors la jointure est réalisée entre les attributs $A_{i1}, A_{i2},\ldots, A_{ik}$ et $B_{i1},$

$B_{i2},\ldots, B_{ik}$. Une méthode naïve serait de calculer le score de similarité pour chaque paire d'enregistrements, mais celle-ci serait extrêmement coûteuse en CPU et non scalable pour des millions d'enregistrements. Par conséquent, l'objectif recherché par la plupart des méthodes proposées dans la littérature [16] est de réduire le coût de ce calcul $O(n^2)$ en O(n*w) avec w << n, c'est-à-dire de réduire le nombre de paires pour lesquelles la distance de similarité est calculée. Pour compléter, de nombreux modèles de décision ont été proposés pour confirmer ou informer les hypothèses d'appariement entre les enregistrements candidats. Le Tableau 12 - les classe en trois catégories selon leur appartenance à un type de modèle probabiliste, empirique ou basé sur des connaissances.

| ***Modèles décisionnels pour l'appariement (Auteurs) (Outil)*** | ***Type*** |
|---|---|
| Modèle basé erreur (Fellegi & Sunter, 1969) | ***Probabiliste*** |
| Méhode basée sur le maximum de vraisemblance (Dempster *et al.* 1977) | |
| Modèle par induction (Bilenko et Mooney, 2003)<br>Modèle basé sur le clustering (Elfeky *et al.* 2002) (Tailor) | |
| 1-1 matching et Bridging File [36] | |
| Tri des plus proches voisins et ses variantes [12] | ***Empirique*** |
| Appariement d'objets XML [35] | |
| Structure hiérarchique (Anantakrishna *et al.* 2002) (Delphi) | |
| Prédiction d'appariement basée sur des « indices » sur le domaine (Buechi *et al.* 2003) (Clue) | ***À base de connaissances*** |
| Fonctions de transformation (Tejada *et al.* 2001) (Active Atlas) | |
| Variante de tri des plus proches voisins utilisant des règles pour identifier et fusionner les doublons (Intelliclean) [19] | |

**Tableau 12 - Modèles décisionnels pour l'appariement des données**

## 7.2 Valeurs manquantes

Les données manquantes son classées en trois grandes catégories :

- les données manquantes complètement aléatoires (*Missing Completely At Random, MCAR*) : les enregistrements ayant une donnée manquante ne peuvent pas être distingués de ceux ayant une donnée renseignée. La probabilité qu'une donnée soit manquante ne dépend ni des valeurs des variables observées ni de la valeur non observée.
- les données manquantes aléatoires ou ignorables (*Missing At Random, MAR*) : le fait d'avoir une donnée manquante dépend d'autres caractéristiques observées, mais pas de la valeur manquante (qui aurait pu être renseignée). La probabilité qu'une donnée soit manquante dépend des valeurs des variables observées mais non de sa vraie valeur.
- les données manquantes informatives non aléatoires et non ignorables (*Missing Not At Random, MNAR*) : le fait d'avoir une donnée manquante n'est pas aléatoire, ne peut pas être déduit des autres variables et dépend de la valeur manquante (qui aurait pu être renseignée). La probabilité qu'une donnée soit manquante dépend de sa vraie valeur (non observée).

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

(a) en ne considérant que les données complètes : seuls les enregistrements ayant tous les attributs renseignés et complets sont analysés. Facile à mettre en oeuvre, cette approche n'est praticable que sur un faible nombre de données manquantes complètement aléatoires (MCAR) mais elle introduit un biais important,

(b) en n'analysant que les données disponibles : les effectifs seront alors différents selon l'attribut considéré. Elle est valable si les données sont manquantes complètement aléatoires (MCAR) et dans ce cas, elle fournira un estimateur non biaisé mais la variance sera incorrecte.

(c) par imputation : la valeur manquante est remplacée par une valeur observée dans un autre enregistrement ayant les mêmes caractéristiques. Cette dernière méthode assez simple nécessite une métrique pour choisir les variables d'appariement (nombre de classes pour les variables catégorielles) et le calcul d'une distance (permettant la combinaison de variables quantitatives et qualitatives). L'enregistrement le plus "proche" est alors retenu. Pour assurer une variance raisonnable, un nombre suffisant d'enregistrements doit être considéré. La méthode peut toutefois conduire à des estimateurs biaisés. Le Tableau 13 - présente trois méthodes d'imputation.

| ***Méthode*** | ***Principe*** | ***Calcul des valeurs imputées*** |
|---|---|---|
| Imputation par la moyenne | La "moyenne" de la variable observée dans les enregistrements remplace les données manquantes. Cette méthode fournit des estimations non biaisées si les données sont complètement aléatoires (MCAR) mais induit une distorsion de la distribution empirique : si la variable est regroupée en classes, toutes les données manquantes sont dans la même classe. Elle est à déconseillée sur des données ignorables (MAR). | Soit *U* la population fine de taille *N*, $z_i = 1 \forall i \in U, f(z_i) = \beta$ $\sigma_i^2 = \sigma^2$ on a alors la valeur imputée $y_i^*$ pour remplacer la valeur manquante $y_i$ telle qu'étant la moyenne pondérée des valeurs renseignées de la population *U* : $y_i^* = \bar{y}_r$ |
| Imputation par modèle de régression linéaire | La variable est modélisée par un modèle de régression (à partir des données observées) et la prédiction du modèle remplace la donnée manquante. Il est possible d'ajouter un aléa à la prédiction. La qualité de l'imputation dépend de la technique de modélisation. | La fonction d'imputation est telle que : $f(z_i) = \beta_0 + \beta_1 z_i$ $\sigma_i^2 = \sigma^2$ la valeur imputée est telle que : $y_i^* = \bar{y}_r - \hat{B}_{1r}\bar{z}_r + \hat{B}_{1r}\bar{z}_i$ avec la moyenne pondérée des valeurs renseignées pour les variables y et z telle que : $(\bar{y}_r, \bar{z}_r) = \frac{1}{\sum_{i \in s_r} w_i} \sum_{i \in s_r} w_i (y_i, z_i)$ |
| Imputation par le maximum de vraisemblance | Cette méthode (adaptée aux données MAR) utilise toutes les informations disponibles et calcule directement les variances des paramètres et des tests statistiques. La procédure itérative est menée en 2 étapes : 1) le calcul de l'espérance : identification de la distribution des données manquantes en fonction des données observées et les variables explicatives, 2) l'étape de maximisation qui remplace les données manquantes par les valeurs attendues. | En considérant un échantillon $x = (x_1, x_2, \ldots, x_n)$ d'individus suivant une loi $f(x_i;\theta)$ paramétrée par $\theta$, on cherche à déterminer le paramètre $\theta$ maximisant la log-vraisemblance donnée par $L(x;\theta) = \sum_{i=1}^{n} \log f(x_i, \theta)$. Cet algorithme est particulièrement utile lorsque la maximisation de $L$ est très complexe mais que, sous réserve de connaître certaines données judicieusement choisies, on peut très simplement déterminer $\theta$. Dans ce cas, on s'appuie sur des données complétées par un vecteur $z = (z_1, z_2, \ldots, z_n)$ inconnnu. En notant $f(z_i \mid x_i;\theta)$ la probabilité de $z_i$ sachant $x_i$ et le paramètre $\theta$, on peut définir la log-vraisemblance complétée comme la quantité : $L((x,z);\theta) = \sum_{i=1}^{n} \left(\log f(z_i \mid x_i, \theta) + \log f(x_i;\theta)\right)$ $L(x;\theta) = L((x,z);\theta) - \sum_{i=1}^{n} \left(\log f(z_i \mid x_i, \theta)\right)$ L'algorithme EM est une procédure itérative basée sur l'espérance des données complétées conditionnellement au paramètre courant. En notant $\theta^{(c)}$ ce paramètre, on peut écrire : $E\left[L(x;\theta) \mid \theta^{(c)}\right] = E\left[L((x,z);\theta) \mid \theta^{(c)}\right] - E\left[\sum_{i=1}^{n} \left(\log f(z_i \mid x_i, \theta)\right) \mid \theta^{(c)}\right]$ On montre que la suite définie par $\theta^{(c+1)} = \arg\max_{\theta}\left(Q\left(\theta, \theta^{(c)}\right)\right)$ fait tendre $L\left(x;\theta^{(c+1)}\right)$ vers un maximum local. |

**Tableau 13 - Quelques méthodes d'imputation**

On distingue généralement les méthodes d'imputation dites déterministes de celles dites aléatoires. Les méthodes déterministes sont celles qui fournissent une valeur fixe étant donné l'échantillon si le processus d'imputation est répété (par exemple, imputation par la moyenne, par le ratio, par régression et par plus proche voisin). Les méthodes aléatoires sont celles qui ont une composante aléatoire; par conséquent, ces méthodes ne fournissent pas nécessairement la même valeur étant donné l'échantillon si le processus d'imputation est répété (par exemple, imputation par *hot-deck aléatoire*).

La majorité des méthodes d'imputation peut être représentée de la façon suivante :

$$y_i = f(z_i) + \varepsilon_i$$

$$E(\varepsilon_i) = 0, E(\varepsilon_i \varepsilon_j) = 0, i \neq j, E(\varepsilon_i^2) = \sigma_i^2$$

où $z$ est un vecteur de variables auxiliaires disponible pour toutes les valeurs dans l'échantillon des données s.
La valeur imputée $y_i^*$ est obtenue en estimant la fonction $f(z_i)$ par $\hat{f}_r(z_i)$ au moyen des valeurs renseignées $i \in s_r$ c'est-à-dire $y_i^* = \hat{f}_r(z_i) + e_i^*$.
Avec le résidu $e_i^* = 0$ dans le cas des méthodes déterministes.

Dans le cas des méthodes aléatoires, un résidu aléatoire qui correspond aux résidus observés dans l'ensemble $s_r$ des valeurs renseignées est utilisé tel que :

$$e_i^* = \left[ y_j^* - \hat{f}_r(z_j) \right] \frac{\hat{\sigma}_i}{\hat{\sigma}_j}, j \in s_r$$

Exemple : Imputation de données manquantes

A partir des valeurs renseignées pour les attributs Age et Revenu_Hebdo de l'entrepôt DW de notre exemple précédent, nous calculons le revenu hebdomadaire moyen par tranche d'age. Le Tableau 14 - montre la moyenne de la variable *Revenu_Hebdo* par tranche d'âge pour une population de taille *N* = 11270 individus.

| ***Tranche âge*** | 15-19 | 20-24 | 25-29 | 30-34 | 35-39 | 40-44 | 45-49 | 50-54 | 55-59 | 60+ |
|---|---|---|---|---|---|---|---|---|---|---|
| ***Moyenne Revenu_Hebdo*** | 139.7 | 343.6 | 513.9 | 587.2 | 625.6 | 661.5 | 704.5 | 692.4 | 629.6 | 515.4 |

**Tableau 14 - Moyenne du *Revenu Hebdomadaire* par tranche d'âge**

La moyenne de la variable *Revenu_Hebdo* dans la population est 555 Euros. Un simple coup d'oeil au tableau révèle qu'il y a une relation entre les variables *Revenu_Hebdo* et *âge*. L'objectif est d'estimer la moyenne de la variable *Revenu_Hebdo* pour deux domaines d'intérêt : le groupe des 15-19 ans et celui des 30-34 ans. Notons que le premier domaine est celui pour lequel la moyenne est la plus éloignée de la moyenne de la population alors que le deuxième est celui pour lequel la moyenne est le plus près de la moyenne de la population. De cette population, *R* = 5000 échantillons aléatoires simples sans remise, de taille *n* = 500, ont été tirés. Dans chaque échantillon, de la non-réponse à la variable *Revenu_Hebdo* a été générée selon un mécanisme de non-réponse uniforme. Le taux de réponse a été fixé à 70%. Pour imputer les valeurs manquantes, nous avons utilisé l'imputation par moyenne:
- $y_i^* = \bar{y}_r$ la moyenne globale des valeurs renseignées qui ne tient pas compte des domaines d'intérêt.
Le Tableau 15 - présente le biais relatif de l'estimateur imputé. Les résultats montrent

le biais relatif des estimateurs imputés est négligeable (0.5% pour les 15-19 ans et 0.4% pour les 30-34 ans).

| Tranche Age | Biais pour $y_i^* = \bar{y}_r$ |
|---|---|
| 15-19 | 0.5 |
| 30-34 | 0.4 |

**Tableau 15 - Biais relatif (%) de l'estimateur imputé**

## 7.3 Détecter les valeurs aberrantes ou isolées

Concernant les données aberrantes ou isolées (*outliers*) : les techniques de détection employées sont des graphes de contrôle, des techniques basées respectivement *i)* sur un modèle, *ii)* sur une comparaison, *iii)* sur des méthodes géométriques de mesure de distance à l'ensemble des données [15], *iv)* sur la distribution (ou la densité) de la population des données avec la notion d'exception locales (*local outliers*) [3] ou, encore, *v)* basées sur la déviation dans une série temporelle de données. D'autres tests de « *goodness-of-fit* » tels que celui du $Chi^2$ permettent de vérifier l'indépendance des attributs, le test de Kolmogorov-Smirnov permet de mesurer la distance maximum entre la distribution supposée des données et la distribution empirique calculée à partir des données. La Figure 9 présente les graphes de quelques-uns de ces tests sur un échantillon de 1000 individus pour les valeurs (*Revenu_Hebdo, Age)* stockées dans l'entrepôt DW. Les tests univariés permettent de valider des techniques d'analyse et des hypothèses sur les modèles employés. D'autres tests plus complexes et pour les cas multivariés sont présentés dans [8] (par exemple, *DataSphere* : pyramides, hyper-pyramides et test de Mahalanobis pour des distances entre moyennes multivariées).

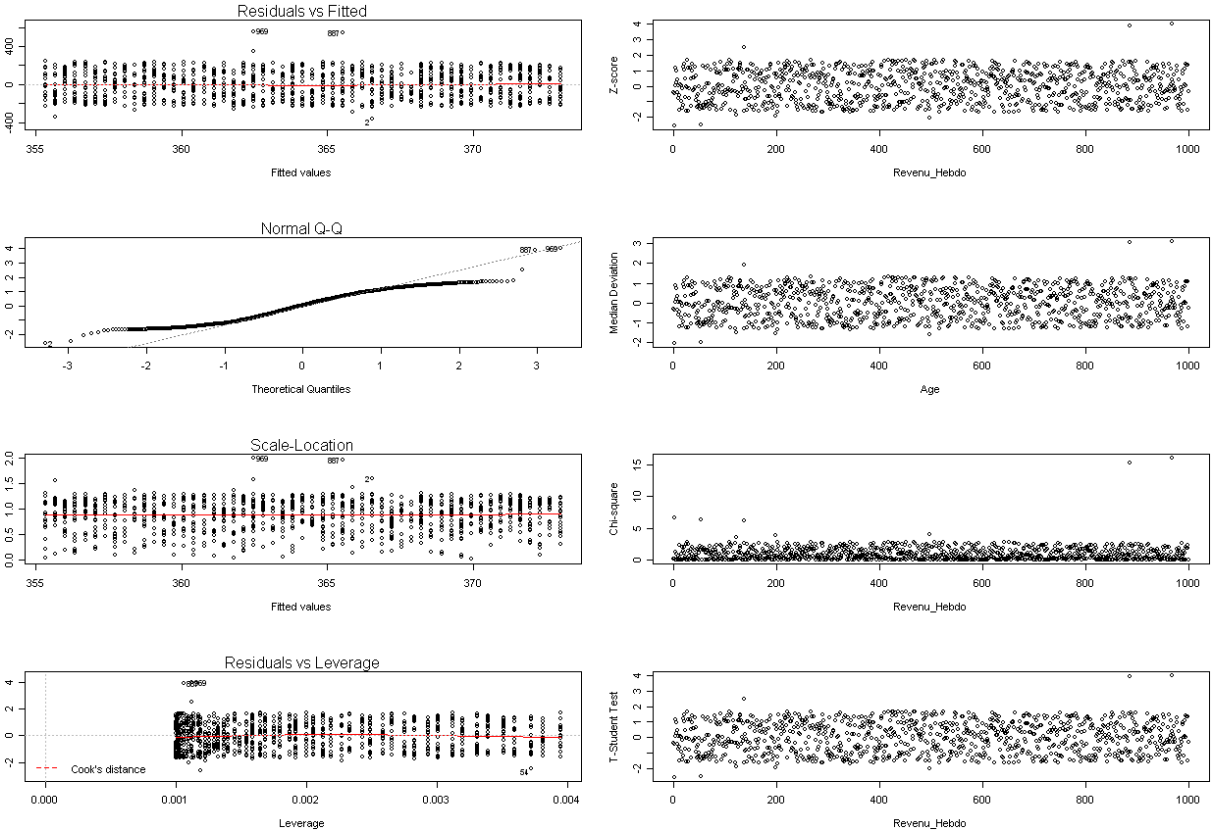



Figure 9 - Quelques tests pour la détection des valeurs aberrantes ou isolées

# 8 Conclusion

De nombreux cas décrits dans la littérature et dans la presse scientifique, révèlent de nombreuses situations alarmantes liées aux enjeux de la qualité des données dans les bases, entrepôts de données et systèmes d'information commerciaux, médicaux, du domaine public ou de l'industrie. Les approches jusqu'ici mises en œuvre sont le plus souvent *ad hoc*, fragmentées et spécifiques à des domaines d'application relativement cloisonnés. Des solutions théoriquement fondées et validées en pratique sont aujourd'hui très attendues pour évaluer et contrôler la qualité des données. Plusieurs verrous scientifiques dans ce domaine ont été identifiés [8][2] et ils expliquent d'ailleurs les limitations des outils actuellement disponibles. A titre indicatif, en voici les principaux :

- l'hétérogénéité et la diversité des données multi-sources à intégrer : les données sont extraites de différentes sources d'informations puis intégrées alors qu'elles possèdent différents niveaux d'abstraction (d'une donnée brute à un agrégat). Les données sont intégrées au sein d'un même jeu de données qui contient donc potentiellement une superposition de plusieurs traitements statistiques. Ceci nécessite une très grande précaution dans l'analyse de ces données. Les jointures entre les différents jeux de données sont difficiles (quand elles ne sont pas totalement faussées) par le problème de l'identification non ambiguë des données et l'impact des données manquantes,
- les volumes de données manipulées et le problème de passage à l'échelle des techniques de mesure et de détection : si certaines méthodes statistiques sont tout à fait adaptées pour fournir des résumés décrivant la qualité de grandes quantités de données numériques, elles s'avèrent inefficaces sur des données multidimensionnelles en grandes dimensions,
- la richesse et la complexité des données telles que les séries temporelles, les données extraites de pages *Web* (*web-scrapped*) et les données textuelles ou multimédias (combinant texte, audio, vidéo, image) pour lesquelles on ne dispose pas (ou très peu) de métriques de la qualité,
- la dynamique des données et de leur qualité : les bases de données modélisent une portion du monde réel en constante évolution. Or, dans le même temps, la qualité des données peut devenir obsolète. Le processus d'estimation et de contrôle doit être constamment reconduit en fonction de la dynamique du monde réel modélisé ainsi que des changements des centres d'intérêt ou de l'attention particulière portée à certaines données (car des données jugées critiques à un instant donné ne le restent pas indéfiniment).

Pour conclure, les multiples problèmes évoqués dans ce dossier offrent plus que jamais d'intéressantes perspectives de recherche pour les différentes communautés scientifiques travaillant autour de la qualité des données en Statistiques, Bases de Données, Ingénierie des Connaissances et Gestion de Processus. Pour les entreprises et industriels, détenteurs et garants de leurs masses de données, la qualité des données demeure un problème récurrent, les guidant ponctuellement vers des choix pragmatiques, souvent à court terme par manque de moyens et d'appuis hiérarchiques. Il est alors clair que pour apporter des solutions concrètes, opérationnelles sur le long terme et théoriquement fondées, des collaborations étroites entre le monde académique et les industriels sont nécessaires.

# Bibliographie


[1] V. Barnett and T. Lewis : *Outliers in Statistical Data*. John Wiley and Sons, 1994.

[2] C. Batini, T. Catarci et M. Scannapiceco : *A survey of data quality issues in cooperative information systems* ; tutorial présenté à International Conference on Conceptual Modeling (ER), 2004.

[3] M. Breunig, H. Kriegel, R. Ng et J. Sander : *LOF: Identifying density-based local outliers* ; International Conference ACM SIGMOD, pp. 93-104, 2000.

[4] L. Berti-Equille : Modelling and measuring data quality for quality-awareness in data mining, *Quality Measures in Data Mining*, Studies in Computational Intelligence, F. Guillet and H. Hamilton (eds), Springer, June 2006.

[5] F. Caruso, M. Cochinwala, U. Ganapathy, G. Lalk et P. Missier : *Telcordia's database reconciliation and data quality analysis tool* ; International Conference on Very Large databases (VLDB), pp. 615-618, 2000.

[6] P. Christen and T. Churches : Febrl - Freely extensible biomedical record linkage (Manual, release 0.3), http://datamining.anu.edu.au/software/febrl/

[7] P. Christen and K. Goiser : Quality and Complexity Measures for Data Linkage and Deduplication, Quality Measures in Data Mining, Studies in Computational Intelligence, F. Guillet and H. Hamilton (eds), Springer, June 2006.

[8] T. Dasu et T. Johnson : *Exploratory data mining and data cleaning* ; Wiley, 2003.

[9] T. Dasu, T. Johnson, S. Muthukrishnan et V. Shkapenyuk : *Mining database structure or, How to build a data quality browser* ; Proceedings of ACM SIGMOD Conference, 2002.

[10] H. Galhardas, D. Florescu, D. Shasha, E. Simon et C. Saita : *Declarative data cleaning: Language, model, and algorithms* ; International Conference on Very Large Databases (VLDB), pp. 371-380, 2001.

[11] R. Gray, B. Carey, N. McGlynn et A. Pengelly : *Design metrics for database systems* ; BT Technology Journal, 9(4), 1991, pp. 69-79.

[12] M. Hernandez et S. Stolfo : *Real-world data is dirty: Data cleansing and the merge/purge problem* ; Data Mining and Knowledge Discovery, 2(1), pp. 9-37, 1998.

[13] T. Imielinski, W. Lipski. Incomplete information in relational databases. Journal of the ACM, 31(4):761 - 791, 1984.

[14] T. Johnson et T. Dasu : *Comparing massive high-dimensional data sets* ; International Conference KDD, pp. 229-233, 1998.

[15] E. Knorr et R. Ng : *Algorithms for mining distance-based outliers in large datasets* ; International Conference on Very Large Databases (VLDB), pp. 392-403, 1998.

[16] N. Koudas et D. Srivastava : *Approximate joins: Concepts and techniques* ; tutorial donné à International Conference on Very Large Databases (VLDB), 1363, 2005.

[17] A. Levy: Obtaining complete answers from incomplete databases. In Proc. of the 22nd Intl. Conference on Very Large Data Bases (VLDB'96), 1996.

[18] R. J. A. Little et D. B. Rubin : *Statistical analysis with missing data* ;

Wiley, New-York, 1987.
[19] W. L. Low, M. L. Lee et T. W. Ling : *A knowledge-based approach for duplicate elimination in data cleaning* ; Information System, Vol. 26 (8), 2001.
[20] A. Motro. Integrity = validity + completeness. ACM Transactions on Database Systems, 14(4):480 - 502, 1989.
[21] L. Moody, G. Shanks et P. Darke : *Improving the quality of entity relationship models - Experience in research and practice* ; International Conference on Conceptual Modeling, pp. 255-276, 1998.
[22] F. Naumann : *Quality-driven query answering for integrated information systems* ; Lecture Notes in Computer Science, vol. 2261, Springer, 2002.
[23] F. Naumann, U. Leser et J. Freytag : *Quality-driven integration of heterogeneous information systems* ; International Conference on Very Large Databases (VLDB), 1999.
[24] G. Navarro : *A guided tour to approximate string matching* ; ACM Computer Surveys, 33(1), pp. 31-88, 2001.
[25] S. Parsons. Survey on methods for handling imperfect information. IEEE Transactions on Knowledge and Data Engineering, 8(3), 1996.
[26] R. K. Pearson : *Data mining in face of contaminated and incomplete records* ; SIAM International Conference on Data Mining, 2002.
[27] M. Piattini, M. Genero, C. Calero, C. Polo et F. Ruiz : *Database quality* ; in Chapitre 14, Advanced Database Technology and Design, Artech House, pp. 485-509, 2000.
[28] M. Piattini, C. Calero et M. Genero (eds.) : *Information and database quality* ; The Kluwer International Series on Advances in Database Systems, Vol. 25, 2002.
[29] E. Rahm et H. Do : *Data cleaning: Problems and current approaches* ; IEEE Data Engineering Bulletin 23(4), pp. 3-13, 2000.
[30] V. Raman et J. M. Hellerstein : *Potter's Wheel: an interactive data cleaning system* ; International Conference on Very Large Databases (VLDB), 2001.
[31] T. Redman : *Data quality: The field guide* ; Digital Press (Elsevier), 2001.
[32] J. L. Schafer : *Analysis of incomplete multivariate data* ; Chapman & Hall, 1997.
[33] P. Vassiliadis, Z. Vagena, S. Skiadopoulos et N. Karayannidis : *ARKTOS: A tool for data cleaning and transformation in data warehouse environments* ; IEEE Data Engineering Bulletin, 23(4), pp. 42-47, 2000.
[34] R. Wang, V. Storey et C. Firth : *A framework for analysis of data quality research* ; IEEE Transactions on Knowledge and Data Engineering, 7(4), pp. 670-677, 1995.
[35] M. Weiss et F. Naumann : *DogmatiX tracks down duplicates in XML* ; Proc. of the 2005 ACM SIGMOD International Conf. on Management of Data, Baltimore, MA, USA, juin, 2004.
[36] W. E. Winkler : *Methods for evaluating and creating data quality* ; Information Systems, Vol. 29, no. 7, p. 531-550, 2004.
[37] E. Zimanyi, A. Pirotte. Imperfect knowledge in databases. Kluwer Academic Publishers, 1996.