UMONS
Université de Mons

Faculté
des Sciences

Mémoire

Data Repairing

Project made by : Maxime Van Herzeele
Academic Year : 2017-2018
dissertation director : Jeff Wijsen
Section : 2nd Master Bloc in ComputerSciencesSciences

Faculté des Sciences • University of Mons • Place du Parc 20 • B-7000 Mons

Remerciements

Todo : remerciement

Table des matières

Chapitre 1

Introduction

Many institutions and companies collect, store and use a lot of data. These data could be dirty which means they contain erroneous information. An Erroneous information may mislead anyone who want to use the database. To prevent this problem, data should respect integrity constraints which are rules in database. Any information who doesn't fit these constraints is considered as a dirty data. But these constraints may be imprecise as well, failing to identify good data and dirty data. For this reason, some data aren't identify as violation as they should be and others data are identify as violent and they shouldn't be. Both mistakes on data and integrity constraints are a problem for anyone using the database.

During my training, i had to work on a project related to a database with such problems. It has a huge impact for a part of my project. The repair of these data is a project for 2018 and i feel it would be interesting to study the data repairing concept.

Data repairing means recover erroneous data but also repair bad integrity constraints. In this thesis we are going to analyse the *θ-tolerant repair model* as explained in a scientifique article[3]. First of all, we will to re-explain some notions and definition we need to understand the model. Then, we will present some database we used to illustrate data repairing models. Next we'll present some data repairing models with among them the θ-model. We'll theorically compare them and identify pros and cons for all them.
TO CONTINUE.

Chapitre 2

?(Find a good name)

In this chapter we'll remind some important notions that we are going to use to explain some data repairing models. We use database following the relational model which was introduced by E.F. Codd [**?**].

2.1 Relational Model

It's important to organize data and model the links between information. It can be done using the relational model with integrity constraints. It's one of the most used model for database management. The goal of this chapter is not to explain this model entirely, but remind some notions like the integrity constraints and others things we'll use to describe and explain data repairing models.

In this relationnal model [2]

- An alphabet A of predicates symbols. Each of them are unique(2 different symbol means two different things) and they are used to denote the relations, the objects and values in the database
- A set of *integrity constraint* which are rules. These rules define the consistency of data in the database. They are assertions of symbols of the alphabet A.

On the main paper which this thesis is based, they define a denial constraint as [3] :

Definition 1. *Consider a relation scheme R with attributes $attr(R)$. Let predicate space $\mathbb{P}$ be a set of predicate P in the form $v_1\phi V_2$ or $v1\phi c$ with $v_1, v_2 \in t_x.A$, $x \in \{\alpha, \beta\}$, $t_\alpha, t_\beta \in R$, $A \in attr(R)$, c is a constant and $\phi \in \{=, <, >, \leq, \geq, \neq\}$ is a build-in operator. A* ***Denial Constraint(DC)*** *:*

$$\varphi : t_\alpha, t_\beta, ... \in R, \neg(P_1 \wedge ... \wedge P_m)$$

states that for any tuples $t_\alpha, t_\beta, ...$ from R, all the predicates $P_i \in pred(\varphi)$,i = 1,...,m should not be true at the same time.

In other words, a denial constraint is a conjunction of predicates that shouldn't be true all in the same time. So if one of the predicates is false, the data is consider to be clean.

2.2 Database

In this section we'll present databasse we used as example in this thesis. These databases are used to ulistrate data repairing models and others notions we'll define.

The first database comes from the main article used as bibliography in this thesis[3].

	Name	BirthDay	Cellphone Number	Year	Income	Tax
t1	Ayres	8-8-1984	322-573	2007	21k	0
t2	Ayres	5-1-1960	***-389	2007	22k	0
t3	Ayres	5-1-1960	564-389	2007	22k	0
t4	Stanley	13-8-1987	868-701	2007	23k	3k
t5	Stanley	31-7-1983	***-198	2007	24k	0
t6	Stanley	31-7-1983	930-198	2008	24k	0
t7	Dustin	2-12-1985	179-924	2008	25k	0
t8	Dustin	5-9-1980	***-870	2008	100k	21k
t9	Dustin	5-9-1980	824-870	2009	100k	21k
t10	Dustin	9-4-1984	387-215	2009	150k	40k

TABLE 2.1 – Table de l'article de référence

.

La seconde base de données que nous allons utilisés est inventée de toute pièce. C'est un exemple de base de données que l'on peut retrouver dans un service public. Les attributs de la tables sont :

The second database we are going to use comes from a personal experience. In a training, i had to work on a project related to a database with some dirty data. These data can't be used outside the company but we'll try to get the main idea. It's a table named person, who got several basic information on people from Belgium[1].

- **NISS :** The national number of the person. A national number is unique. Usually, a NISS is formated like this[1]
  - It start with the birthdate of the person in a YY-MM-DD format. Exception are made for stranger(People without Belgian nationality), but for ease we won't consider these cases.
  - Number 7 to 9 is even for men and odd for female.
  - Remaining number are the modulo 97 of the 9 first number.
- **Name :** Person's name.

[1] Every data in our database are fictional person.

— **Firstname :** Person's firstname.

— **Birth_Date :** birthDate in DD-MM-YYYY format.

— **Decease_Date :** Person's date decease.

— **Civil_State :** Person's current civil state(example : single, married, divorced, decease, widow,...)

	Niss	Name	Firstname	Birth_date	Decease_date	civil_state
t1	14050250845	Dupont	Jean	14-05-1902	18-05-1962	decease

TABLE 2.2 – Table Person

.

2.3

Chapitre 3

Data Repairing

3.1 Integrity constraints variations

Insertion et délétion.

3.2 θ-tolerant model

3.3 Others

3.3.1 Holistic

et d'autres

Chapitre 4

Implementation and comparison with others models

Chapitre 5

Conclusion

Bibliographie

[1] Description des données du registre national et du registre bcss. https://www.ksz-bcss.fgov.be/sites/default/files/assets/services_et_support/cbss_manual_fr.pdf. accessed : 2018-02-15.

[2] ics relational database model. http://databasemanagement.wikia.com/wiki/Relational_Database_Model. Accessed : 2018-02-13.

[3] Giuseppe De Giacomo Andrea Cali, Diego Calvanese and Maurizio Lenzerini. Data integration under integrity constraints. Technical report, Departement of Computer Sciences and System, University of Roma "La Sapienza", •.

[4] Han Zhu Shaoxu Song and Jianmin Wang. Constraint-variance tolerant data repairing. Technical report, Tsinghua National Laboratory of Information Science and Technology, 2016.