Paper 162-28

The Horror of Bad Data Quality

ir. Henri Theuwissen, SOLID Partners, Belgium
Nancy Croonen, SOLID Partners, Belgium

ABSTRACT

Operational databases contain 'core' data and 'reporting' data. 'Core' data is vital for the operational applications, whereas 'reporting' data is 'nice-to-have'. Usually the quality of this 'reporting' data is very low. Once this data is used in a Data Warehouse or a CRM application, users suffer from the poor quality.

For example: what happens if a customer is defined twice in the customer database, and you approach this customer in a cross-selling campaign? What happens if you send a birthday card to a customer on a wrong date?

Poor quality data creates a short-term direct cost and has a long-term impact on the image of your company.

INTRODUCTION

The paper discusses different aspects of bad data quality:

- Several types of quality problems.
- Reasons of bad quality data.
- Detection mechanisms for bad quality data.
- Solutions to improve data quality.

All these routines are integrated in the processes, managed by SAS/Warehouse Administrator®.

All topics are illustrated with examples from real life projects.

Attendees should have a basic knowledge of SAS®. SAS modules covered include Base SAS® and SAS/Warehouse Administrator®.

REASONS FOR BAD QUALITY DATA

HISTORICAL CHANGES

The importance of a data item might change over time. Example: the date of birth of the customers of an insurance company.

- For most insurance policies, the date of birth or the age of the customer were not important in the past, except for life insurance contracts.
- Later, insurance companies started marketing campaigns and they were interested in the age of the customer. From that moment on, they collected the year in which the customer was born.
- Recently insurance companies analyzed car accidents and they found out that most accidents were caused by young drivers and they created contracts where the premium due is related to the age of the driver. So, the date of birth became very important.

As a result, the operational databases contain the date of birth, with the following values:

- For old contracts / customers the field is missing.
- For recent contracts the field is filled in correctly.
- For 'intermediate' contracts the year of birth was available. A date was created from this year value, taking either January 1st, July 1st or December 31st.

DATA USAGE

Assume that you go to your bank office to buy some shares. The broker will have an application to fill in some information, like:

- The amount of shares that you buy and their value: these fields must be correct.
- Your name and address: this information must be more or less correct, i.e. there may be some errors in these fields, like using 'str.' in the address instead of 'street'.
- Your profession: this is information that might be used later by the marketing department of the bank. Since filling in this information takes some time, the broker often skips it, or fills in some default values.

The quality of the data is sufficient for it's operational goal: giving you the correct shares and sending a correct invoice to you. Once the bank starts using this data for other purposes (Business Intelligence) or when they start combining this data with other internal or external sources, problems show up.

COMPANY MERGERS

During the last few years several bank and insurance companies merged together into larger holdings, to be more competitive in a larger market place. As a result several databases must be merged together into a single corporate database. Usually the source databases reside in different DBMSs, on different platforms, using different business rules and physical structures. Powerful integration rules must be applied to create a good quality result.

PRIVACY REGULATIONS

Within several countries laws exist to protect the privacy of individuals. Hence, for individuals you may not exchange, sell or buy personal information. This means that when you want to build or maintain a consumer database for a B2C project, you have to collect, code and maintain the data yourself.

Some of these records are 'dormant', i.e. you do not have any contact or business with these people. Knowing that approximately 10 % of the population moves to another address every year, your data will be outdated very soon.

DATA ENRICHMENT

To build a high ROI on your data warehouse, you often enrich your internal data with external sources. The added value of external data is:

- Checking the correctness of your own data.
- Adding fields to your data that you do not have or that you do not collect.
- Getting information about companies or individuals that are not (yet) your customers but they are potential prospects.

Two procedures exist to enrich your data:

- Buying external data and integrating this data into your own data warehouse.
- Providing your data to an external company. This external company will enrich the data and return an updated file.

Often these external companies are experts in data gathering, but have limited expertise in IT, resulting in poor routines to combine your data with their data. As a result they introduce errors in your data warehouse.

HORROR STORIES

HOW BAD DATA QUALITY DESTROYS THE IMAGE OF YOUR COMPANY AND COSTS A LOT OF MONEY

CASE 1: CHECKING FOR TERRORISTS IN YOUR CUSTOMER DATABASES

The Treasury's Office of Foreign Assets Control (OFAC) maintains and distributes (via Internet) a listing of specially designated nationals and blocked persons. Since September 11th, 2001 many companies use this listing to check whether any of these names appear in their customer database.

A part of this listing (two blocks) is shown below. Notice this is a typical text listing, not designed for IT usage: data are very unstructured, which makes it difficult to have an automated lookup process in the customer database.

```
ABU-MARZUQ, Sa'id (a.k.a. ABU MARZOOK, Mousa
Mohammed; a.k.a. ABU-MARZUQ, Dr. Musa; a.k.a.
ABU-'UMAR; a.k.a. MARZOOK, Mousa Mohamed
Abou; a.k.a. MARZUK, Musa Abu), Political
Leader in Amman, Jordan and Damascus, Syria
for HAMAS; DOB 09 Feb 1951; POB Gaza, Egypt;
SSN 523-33-8386 (U.S.A.); Passport No. 92/664
(Egypt) (individual) [SDT]

ABU NIDAL ORGANIZATION (a.k.a. ANO; a.k.a.
BLACK SEPTEMBER; a.k.a. FATAH REVOLUTIONARY
COUNCIL; a.k.a. ARAB REVOLUTIONARY COUNCIL;
a.k.a. ARAB REVOLUTIONARY BRIGADES; a.k.a.
REVOLUTIONARY ORG. OF SOCIALIST MUSLIMS)
[SDT][FTO]
```

Analyzing the records in the listing shows:

- The listing contains information about individuals and about organizations.
- A block can contain several alias names for a person or an organization, these alias names are preceded by the indication a.k.a., f.k.a. or n.k.a.
- The last name is always presented in uppercase, and it is the first item in the string.
- The country information is always shown at the end, between brackets.
- Records sometimes contain unbalanced brackets. This complicates the cleaning process.
- Each record in the listing can contain descriptive information like the address, passport number etc.
- Coding is not at all structured.

A Belgian bank company spent 2 man-weeks to do a manual checking of their customer database.

After a cleaning process, the two sample records create the following result:

NAME	FIRST NAME
ABU MARZOOK	Mousa Mohammed
ABU-MARZUQ	Musa
ABU-MARZUQ	Said
ABU-UMAR	
MARZOOK	Mousa Mohamed Abou
MARZUK	Musa Abu

NAME
ABU NIDAL ORGANIZATION
ANO
ARAB REVOLUTIONARY BRIGADES
ARAB REVOLUTIONARY COUNCIL
BLACK SEPTEMBER
FATAH REVOLUTIONARY COUNCIL
REVOLUTIONARY ORG. OF SOCIALIST MUSLIMS

CASE 2: SENDING BIRTHDAY VOUCHERS

A restaurant collects information about it's customers: when eating in the restaurant, the customers will find a card on their tables to be filled in with some personal information like name, address, phone number, age.

After a while the restaurant owner decides to start a campaign, where he offers customers a free dinner for their birthday if three paying guests accompany them. He decides to use the same information cards, but he now asks the birth date instead of the age. For the old information in their customers database (age and not date of birth) he decides to 'create' the date of birth as January 1st or July 1st or December 31st.

Imagine the reaction of the customers when they receive a happy birthday card in December if they are born on for example the 4th of July.

CASE 3: A CROSS-SELLING CAMPAIGN

For an insurance contract, there are at least three important dimensions:

- The policy itself, with the covered items, premium due, covered amount, start date, etc.
- The customer (who must pay the invoice and who is protected by the contract).
- The broker (who will close the deal and will get commission for it).

Usually the information about the policy is accurate, since it is vital for the insurance contract. Also, broker information is correct and unique: every broker has a unique identification in the insurance company through a Broker ID. The customer information though might be stored in duplicate records, caused by:

- The customer signed two insurance policies (car and life) with two different brokers, because brokers are specialized in different areas.
- Customer information is not accurate: e.g. for a first contract the customer is specified as Theuwissen Robert and for the second contract he is specified as Theuwissen Bob. As a result a new customer is entered in the database and two different customer IDs are created.
- Although lookup tables exist to specify for the correct street and location, based on point-and-click selections, users still can type themselves a value because “new streets were created and these new values do not exist yet in the application.”

Consider the following subset of the insurance company’s data:

Customer ID	Last Name	First Name	Policy Type	Other Data
B1004378	Theuwissen	Robert	Car	…
B1075988	Theuwissen	Bob	Life	…
B2113956	Noten	Richard	Life	…
B3392261	Croonen	Peggy	Car	…
B5439876	Neyens	Bart	Car	…
B5439876	Neyens	Bart	Life	…
B9992154	Simpson	Kelly	Life	…
B8274651	Mullen	Ben	Car	…

- The first two records concern the same person.
- The 3rd record and the 4th record contain information for two different persons, being husband and wife.
- The 5th record and the 6th record concern the same person, identified by one Customer ID.
- The customer in the 7th record only has a life insurance contract.
- The customer in the last record only has a car insurance.

The insurance company starts a cross-selling campaign: they want to bring in customers by a mailing to their existing customers with a life insurance contract. They offer special conditions to these potential new customers: a 20% discount on the premium due. Based on the information in the database a mail is sent to:

Customer ID	Last Name	First Name	Policy Type	Other Data
B1075988	Theuwissen	Bob	Life	…
B2113956	Noten	Richard	Life	…
B9992154	Simpson	Kelly	Life	…

Only the last customer should receive this mailing.

The first customer already has a car insurance contract. Sending the mailing to this customer has a short-term negative cost impact:

- It is a waste of money (letter creation, stamp), because he will not require / sign another car insurance contract.
- The customer will be unhappy, receiving an offer 20% cheaper than his actual payment. He will call the insurance company requesting the same discount for his existing contract.
- The customer will have certain doubts about the accuracy of the insurance company.

For the second customer, his wife signed an existing car insurance contract. Also in this case there is a double negative impact:

- It is a waste of money (letter creation, stamp), because they will not require / sign another car insurance contract.
- The customer expects that the (IT system of the) insurance company knows the concept ‘family’ or ‘married people’.

So, sending mailings to the wrong people will imply a financial cost, without return and will create a negative image about your company.

CASE 4: DATA ENRICHMENT

A large hardware supplier has data from different companies in a B2B application. They use this database for prospecting purposes and they want to enrich this data with information about the creditworthiness of these companies. This information is bought from an external company: the hardware supplier provides a file with the company data and receives the file back, enriched with the creditworthiness information.

A VAT number uniquely defines a company. The hardware supplier does not have the VAT number for all the companies in his database. Hence matching must be done on company name and address.

The external company did not use powerful matching routines and ‘matched’ the data of a large company (ARTESIA) with external data from a very small company (HORTENSIA). As a result the hardware supplier decided that ARTESIA was not important for them as a prospect.

CASE 5: BAD RISK ANALYSIS

Within an insurance company a monthly report shows customers with multiple car accidents and high claim costs. To improve the profitability, the management of the insurance company decides to cancel the contracts of ‘bad’ drivers.

The company CityBird VA has a contract for 12 company cars. They had some accidents during the last year. The company also has a pension plan for their employees. This pension plan is covered by an insurance policy with an annual premium of 370.000 Euro. Within the policy database, these two contracts are defined for two ‘different’ customers.

Customer	Policy	Premium	Claims	Other
CityB VA	Car	20.567	107.098	…
CityBird VA	Pension	370.000		…

Purely based on the bad risk report, the insurance company decides to cancel the contract with CityB VA. The CEO of CityBird VA immediately decides to stop the payment for the pension plan too.

By cleaning and integrating the customers database, the insurance company would be able to calculate the customer value for each customer, and based on the full picture of CityBird VA, they would decide not to cancel the car insurance contract.

CASE 6: FUTURE VALUE ANALYSIS

A bank company wants to create a report showing the potential future value of their customers. To calculate this forecast, two axes are important:

- The general profile of a customer in the past: for example a customer first opens a bank account and afterwards a savings account. Later on the customer gets a loan or starts working with shares on the stock market, using the bank as intermediate broker.
- The customer profile: age, current products, etc.

If the customers database contains duplicate records it is impossible to

- find the complete portfolio of a customer, and build the profile of that customer.
- build the profile of a general customer.
- assign a value or weight to each individual customer.
- perform customer segmentation.

Once all duplicates are removed, a customer value index can be created, allowing defining the correct treatment for each customer:

- Customers with a negative value might be 'pushed' to switch to another company.
- Low value customers are immediately routed to a call center.
- Important customers will receive a personal treatment, by a dedicated account manager.

DATA QUALITY IMPROVEMENT

The following steps are required to improve the quality of your data:

1. Detect the problems.
2. Clean individual data items. Standardize and normalize data.
3. Clean quality problems, related to a combination of data items.
4. Integrate the programs in the full ETL process of your data warehouse.

DETECTING QUALITY PROBLEMS

Standardized Lookup Tables

A technique to detect errors in the data is the usage of lookup tables. These lookup tables might be created in house for company specific data, or can be bought externally. For example: you can buy a database with the official street names.

These databases can be used to

- check your own data
- correct the data in your databases with the standardized information
- add a unique key field to your database, corresponding to the value in the lookup database.

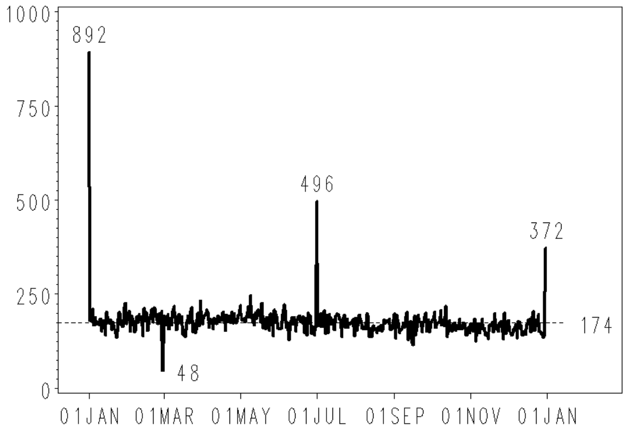
Error Reports

Simple techniques can be sufficient to detect data errors or doubtful data. These techniques include creating reports with

- frequency tables
- scatter plots

Reconsider the case of the company using date of birth for a CRM application. By simply creating a line plot showing the number of people born per day, potential problems are visualized: you expect approximately the same number for every day in the year, except for February 29th. The plot below shows immediately doubtful data for the 1st of January, July 1st and December 31st.

Distribution Birth Days Clients



Although each individual value is a correct date, you are sure that there are problems when you show them all together.

The only possibility to correct these errors is to contact your customers, for example by adding a reply card with the next mailing.

Incompatible Combinations

Finally you can create reports to detect incompatible combinations. Each individual value might be a correct possible value, but by combining them they are incompatible.

Consider the following example: 2 customers have a loan since 1999. For one of them the contract ended. The operational applications store the data in 2 tables: one table defining the contract, with start date and end date, the second table containing payment information. In the data warehouse these two tables are combined, resulting in:

Customer ID	Start Date	End Date	Payment Year	Amount Due
HL13803	31/07/1999	31/12/2001	2000	900
HL13803	31/07/1999	31/12/2001	2001	900
HL13803	31/07/1999	31/12/2001	2002	900
HL73008	12/05/1999		2000	1200
HL73008	12/05/1999		2001	1100
HL73008	12/05/1999		2003	1000

By creating a report combining the end date and the amount due, the following errors are detected:

- Although the contract of customer HL13803 was finished in 2001, there is still an amount due for 2002.
- The contract for the customer HL73008 is still active, but there is no information for payments due in 2002.

CLEANING INDIVIDUAL COLUMNS

Within the operational applications you usually find different fields for last name and first name and for street and number. Users sometimes put this data together in one field.

Examples:

Last Name	First Name	Title
Mr. Smith	Larry	
Smith	Larry	
Smith, Larry		Mr.
Smith	L.	Mr.

Street	Number	Box
Sterrenlaan	12	202
Sterrenlaan	12/202	
Sterrenlaan, 12, 202		
George IV Boulevard	18	
George IV Blvd	18	
George IV Blvd 18		
George IV Boulevard, 18		

You can easily build routines to correct these data items. These routines contain specific business rules per data item.

For example:

- You will work with a lookup table with all possible titles (Mr., Dr., Mrs., etc.) to remove the title from the name field to a separate field.
- You will use a lookup table with synonyms for the street field: STR, STR., STREET are the same, BOULEVARD, BLVARD and BLVD are the same.
- The street name should not contain a number, and certainly not at the end. If a number is used in a street name, it must be presented in Roman.

The cleaning process is an evolutionary process: not all problems will be solved automatically. Based on exception reports you might extend the tables with synonyms to be matched with standardized / normalized values in the lookup tables.

Instead of building yourself these cleaning routines, you might use dfPower Studio from DataFlux®.

REMOVING DUPLICATES

Detecting duplicate records is a difficult task. You need to distinguish between business customers (B2B applications) and consumer customers (B2C applications).
For business customers, unique key identifications are available and not protected by privacy laws. It is quite easy to collect / find these unique identifiers, like for example the VAT number.
For the consumer market customers you usually must fall back on text fields to find and to remove duplicates.

The process of detecting duplicates is illustrated with the following example:

Fieldname	Ex. 1	Ex. 2	Ex. 3
ID	13578902	15589187	17963285
Last Name	Vantheuwissen	Theuwissen	Theuwis, Rob
First Name	Robert	R.	
Street	Minervastraat	Minervastr.	Minervaln 14
Number	14	14/2	
Box			2
Postal Code	1930	1930	1930
Location	Zaventem	Diegem	Diegem
Sex	M		M
Birth Date	02/05/1968	01/01/1968	

The cleaning process is executed in several steps. For each input record, a verification process is executed to find potential duplicates in the other records. To avoid a double process for each record, potential duplicate records are searched only in the higher observation numbers (Customer ID), after all if Customer ID 123 is a potential duplicate for Customer ID 63, then Customer ID 63 is also a potential duplicate for Customer ID 123.

1. Record Cleaning

First, each individual record is cleaned: a cleaning is executed on each individual field, following the business rules defined for each field:

- split first name and last name
- split street, number and box
- adding the correct location for postal code.

The result is shown below:

Fieldname	Ex. 1	Ex. 2	Ex. 3
Last Name	Vantheuwissen	Theuwissen	Theuwis
First Name	Robert	R.	Rob
Street	Minervastraat	Minervastraat	Minervastraat
Number	14	14	14
Box		2	2
Postal Code	1930	1930	1930
Location	Zaventem	Zaventem	Zaventem
Sex	M		M
Birth Date	02/05/1968	01/01/1968	

2. Defining Search Keys

Search keys are created for the last name, the first name and the street name. Such a search key can be created using the SOUNDEX function or by using a more advanced user written routine. The result is shown below: NameKey 1 is the result of the SOUNDEX function; NameKey 2 is the result of another user written routine, creating phonetic keys.

Fieldname	Ex. 1	Ex. 2	Ex. 3
Last Name	Vantheuwissen	Theuwissen	Theuwis
NameKey 1	2DUWS	DUWS	DUWS
NameKey 2	V5325	T25	T2
First Name	Robert	R.	Rob
FNameKey	R163	R	R1
Street	Minervastraat	Minervastraat	Minervastraat

StreetKey	MNRV	MNRV	MNRV
Number	14	14	14
Box		2	2
Postal Code	1930	1930	1930
Location	Zaventem	Zaventem	Zaventem
Sex	M		M
Birth Date	02/05/1968	01/01/1968	

3. Searching for Duplicate Records

Depending on the different fields, a specific test is executed:

- For some fields like sex, the values are compared on equality.
- For date of birth, knowing the possible wrong values, checking results in
  - a. equal
  - b. same year, but different date
  - c. missing
  - d. not equal.
- For fields like last name, first name and street name, a checking is executed using the search keys that were created, instead of the fields itself. This can result in
  - a. equal
  - b. the search key of one record contains the search key of the other record
  - c. missing
  - d. not equal.

  For example, since the value 2DUWS contains the value DUWS the name checking returns these two as potential duplicates.

The process of detecting potential duplicate records is CPU intense: all possibilities must be analyzed. Avoid using SQL queries to solve this problem: SQL queries will consume too much CPU. Consider DATA step processing instead. The following code illustrates a possible DATA step, searching for potential duplicates according to the following business rule: two records are considered as potential duplicates if:

- postal code has the same value, and
- the search key on first name is part of the search key on first name in the other record, and
- the search keys on name are the same in both records or the name in one record is part of the name in the other record.

```
DATA POT_DUPS_1 (DROP=START);
   SET CUSTOMERS;
   BY POSTAL_CODE SRCH_KEY_STREET;
   RETAIN START 0;
   IF FIRST.SRCH_KEY_STREET THEN START = _N_;
   OBSNR = START;
   DO WHILE (OBSNR LT _N_ );
      SET CUSTOMERS (RENAME = (
          CUSTNR          = _CUSTNR
          KEY_NAME        = _KEY_NAME
          KEY_FNAME       = _KEY_FNAME
          BIRTH           = _BIRTH
          POSTAL_CODE     = _POSTAL_CODE
          KEY_STREET      = _KEY_STREET
          STREET_NBR      = _STREET_NBR
          POLBOX          = _POBOX
          SEX             = _SEX
          SRCH_KEY_STREET = _SRCH_KEY_STREET
          SRCH_KEY_NAME   = _SRCH_KEY_NAME
          SRCH_KEY_FNAME  = _SRCH_KEY_FNAME))
              POINT = OBSNR;
      IF INDEX (_SRCH_KEY_FNAME,
                TRIM(SRCH_KEY_FNAME)) > 0 AND
         (INDEX (_KEY_NAME,
                 TRIM(KEY_NAME)) > 0 OR
          INDEX (KEY_NAME,
                 TRIM(_KEY_NAME)) > 0 OR
          SRCH_KEY_NAME = _ SRCH_KEY_NAME)
      THEN OUTPUT;
      OBSNR = OBSNR + 1 ;
   END;
RUN;
```

Comparable DATA steps are executed, testing other combinations, like for example people with the same date of birth and similar names.

All tables with potential duplicate records are combined too find all common or almost common fields in each record. Within the following table, the result is shown: E indicates ‘equal’, C indicates ‘contains’ and M indicates ‘missing’.

Fieldname	Ex. 1	Ex. 2	Ex. 3
Last Name	Vantheuwissen		
NameKey 1	2DUWS	C	C
NameKey 2	V5325		
First Name	Robert		
FName Key	R163	C	C
Street	Minervastraat		
StreetKey	MNRV	E	E
Number	14	E	E
Box		M	M
Postal Code	1930	E	E
Location	Zaventem	E	E
Sex	M	M	E
Birth Date	02/05/1968	C	M

4. Identifying Records as Potential Duplicates

Each field gets a weight and each matching value (equal, contains, missing) also receives a weight. Based on the total value two records are considered as duplicates. For example: you need to have at least 6 E’s or C’s out of 8 possible fields to accept a record as a potential duplicate.

You can change the different weights to give more importance to one or another field, depending on your knowledge of the correctness of a specific field.

5. Using Mathematics

Consider the following example:

Last Name	First Name	Other Data
Greenspan	Jenny	…
Greenspan – Cole	Jenny	…
Cole	Jenny	…

According to the mechanism defined in step 3, Greenspan-Cole is a possible duplicate for Greenspan and Cole is a possible duplicate for Greenspan-Cole, but Cole is not a possible duplicate for Greenspan. A rule in logics in mathematics indicates:

```
IF A = B AND B = C ⇒ A = C
```

Applying this rule in your routines indicates that Cole is a possible duplicate for Greenspan.

Be careful though with this rule. Do not apply the rule in B2B databases, checking for duplicate company names. Consider the following companies, located on the same address:

Name	Other Data
Cohen, Reuters and Ferrero	…
Cohen	…
Ferrero	…

Cohen and Ferrero have their own, private company and they created, together with Reuters another company. You should not merge them together into one single customer identification.

6. Integrating Cleaning Routines in the ETL process

Data quality problems must be solved as early as possible in the full process. You may not allow these problems to enter the staging process of the data warehouse, since that data might be used to create consolidated tables, derived reports etc. The quality problem will be propagated as a virus throughout your data warehouse.

The cleaning process is part of the complete ETL chain that is created and documented through SAS/Warehouse Administrator®. You will need to balance between automatic metadata creation and CPU efficiency: the cleaning steps are defined as Post Processing tasks after creating staging tables or will be defined as User Exits in the creation process of the staging tables. The metadata automatically created by SAS/Warehouse Administrator® for these – mainly DATA step – programs is very limited: they are black boxes in the full process.

7. Acting on Quality Problems.

In the ideal, ultimate world all data quality problems that were detected should be resolved in the operational applications. In reality though this is just wishful thinking.

For the problems that you solved in step 1 by standardizing the data, you might create source code automatically to update the operational tables. This code can be executed by the IT department managing the operational applications.

A possible source code for the 3 sample records is shown below:

```
UPDATE data_base_table
   SET STREET   = 'Minervastraat',
       NUMBER   = '14',
       BOX      = '2',
       LOCATION= 'Zaventem'
   WHERE ID = 15589187;

UPDATE data_base_table
   SET STREET   = 'Minervastraat',
       NUMBER   = '14',
       LOCATION= 'Zaventem'
   WHERE ID = 17963285;
```

Removing duplicates must be executed carefully: you must avoid ‘merging’ different people together into one single customer in the operational applications.

CONCLUSION

Within the past few years, companies were concentrating on making more money, gathering more customers, selling more. With the current negative economic situation cost reduction becomes more and more important. Poor data quality, caused by duplicate records, non-standardized data or errors in the data, create a high cost for companies with zero return.

Usually the poor data quality shows up when bringing data from different sources together, like in a data warehouse.

The short-term ROI on the effort on improving the quality of the data can be measured immediately on the reduced cost in for example mailing campaigns. On a long-term, companies will have better targeted campaigns, resulting in a higher ROI on their marketing efforts.

Data quality checking and improving is a never-ending process, which must be put in place as early as possible in the complete ETL process.

SAS provides several tools for detecting and solving data quality problems.

CONTACT INFORMATION

Contact the authors at:

ir. Henri THEUWISSEN
SOLID Partners NV
Minervastraat 14 Bis
B-1930 ZAVENTEM
Belgium
Work Phone: +32 495 54 52 53
Fax: +32 2 706 03 09
Email: Henri.Theuwissen@SOLIDPartners.be
Web: www.solidpartners.be

Nancy CROONEN
CC Training Services BVBA
Kesseldallaan 12/202
B-3010 KESSEL-LO
Belgium
Work Phone: +32 496 28 45 28
Fax: +32 2 706 03 09
Email: Nancy.Croonen@SOLIDPartners.be
Web: www.solidpartners.be

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.