Scientific Research Publishing

International Journal of Intelligence Science, 2023, 13, 48-62
https://www.scirp.org/journal/ijis
ISSN Online: 2163-0356
ISSN Print: 2163-0283

Dirty Data between Errors and Their Handling—A Firsthand Experience in Solving Dirty Data from Within

Faheem Bukhatwa[1], Ahmed Laarfi[2], Ismahan Salem[3]

[1]Faculty of Computing, Griffith College, Dublin, Ireland
[2]Brevard Public School, Viera, Florida, USA
[3]Retired, Benghazi, Libya.
Email: alaarfi2015@my.fit.edu

How to cite this paper: Bukhatwa, F., Laarfi, A. and Salem, I. (2023) Dirty Data between Errors and Their Handling—A Firsthand Experience in Solving Dirty Data from Within. *International Journal of Intelligence Science*, **13**, 48-62.
https://doi.org/10.4236/ijis.2023.132004

Received: February 21, 2023
Accepted: April 25, 2023
Published: April 28, 2023

Abstract

Managing large amounts of data is becoming part of everyday life in most organizations. Handling, analyzing, searching, and making predictions from big data is becoming the norm for many organizations of many interests. Big data provides the foundations for more benefits and higher values to be extracted from big data. As big data comes with countless benefits, it also comes with many challenges to fulfilling its expectations. Some of those problems haunting big data banks are being termed dirty data. This paper focuses on dirty data while working on an organization's natural live information system. The author was responsible for studying and analyzing a faltering information system and planning and carrying out the required solutions and fixes. The importance of the work carried out lies in the high level of dirty data observed in the system. Therefore, this paper is based on the part of dirty data—the paper focuses on how the team suffered from dirty data and how it was dealt with.

Keywords

Data Science, Database, Artificial Intelligence, System Analysis, Big Data

1. Background

This paper discusses significant work previously carried out by the author at some organizations. At that time, the primary goal of the team who was assigned was developing, maintaining, and solving problems experienced in the organization's computerized information system. The work was implemented and completed. Some of the essential steps and methods followed in carrying out the work will be listed in this paper. The required job was implemented, documented results, and recommendations were issued. This paper is more concerned with the aspect of the discovery of the widespread dirty data within the system. Therefore, the paper outlines the firsthand experience of facing, identifying, and dealing with dirty data in a particular system.

The Information system involved contained a large number of files and records. The data in the entire system exceeded two million records. Therefore, this system can be classified among Big Data.

If a database system is not designed and not developed correctly, it will, over time, develop problems and issues that need to be corrected. It is just a matter of time before dirty data creep into systems or its effects multiply in magnitude. It reduces efficiency, and performance increases cost and maintenance and potentially renders a system unusable, inaccurate, or too costly to continue. The system in this organization was showing many of those symptoms. Our team's job was dissecting this organization's faltering information system, finding its problems, and designing and implementing solutions. Unfortunately, this information system's original system designers and developers were no longer available, nor were the system documentation or code. The task was enormous. This paper is concerned only with the solution model applied to solve the dirty data problem in the system. In this case, some data was missing as it was never included in the original design, which made the database incomplete. This caused many queries to fail or return unexpected results. The paper will review methods applied by the team to handle some of the issues with the relational database involved. Since the source code was unavailable, some new code was created in separate files and placed where the entire database could be easily accessed. New tables were created and added to the existing tables in the database. Some data in the system is found to have a certain level of sensitivity and must be treated accordingly.

2. Introduction

"*In societies before the invention of fire, fighting was the means of resolving disputes. After thousands of years, the two opponents negotiated for days, and if they did not reach a solution, they started battles. Currently, we would have liked to write that the era of war was over, but it seems that we are back to the starting point*!"

Figure 1 represents my paragraph in italic.

We reviewed a method to maintain systems with relational databases programmed in Fox Pro. We did not get information that can be stored in databases of any type or size. Life, in general, when it is fragmented, turns into data. Name, date of birth, residential address, height, weight, license and passport numbers, and others are data. Financial, administrative, commercial, and military information is stored in scalable databases. According to our programming, markets, stock exchanges, banks, and money transfers start as small data that gives information. Data has become the essential commodity that can be sold, bought, and

Figure 1. Retrieved from: https://factsanddetails.com/world/cat56/sub362/item1498.html on 07/01/2020 at 7:23am.

bartered in this era. Various mechanisms have been developed to treat data depending on the type, quantity, and requirements. [1] [2]

Since the beginning of history, information has been the most expensive or valuable commodity and has remained the same today. Our whole life is a collection of information whose worth is determined according to its value and importance. In ancient societies, most of their activities, in addition to daily activities, were expansion, pushing them to clashes and wars. Thus, information in wartime differs in value and importance from information at other times. The lives of primitive societies mainly lacked stability and frequently moved. That led to continuous collision; hence, information was necessary. The various sources of what those societies possessed were allotted to information. We are not surprised when we see all or most inventions were either for the sake of war or preparing for it. Later the story is viewed from a civil point of view. It is used in civilian life. Spying has become a profession. Intelligence became the primary source of information for conflicting or warring parties. The last century was particularly famous for its espionage work after two opposing entities emerged as a product of the Second World War. Information is not limited to use in wars but is essential in all competitive activities. Information remains the most critical commodity in financial markets and commercial exchanges, and whoever has the enormous amount of information controls the market. In the last seventy years since the advent of the computer, we have switched to digitization systems. The form and content of information have changed. Information becomes organized in data form, for which its databases are designed to suit the shape and size of this information. Data is managed according to several factors, the most important of which are size and importance. With the significant growth in computer science and engineering, these databases have become feared and targeted by hackers and spies using more modern means. As a consequence, information security and protection systems emerged.

On the other hand, scientists and researchers started theorizing about the new sciences concerned with data, developing it, and adapting it for exploitation in different areas of life. This adaptation is made by linking it to various other sciences, such as machine learning and artificial intelligence, and creating standard databases based on a new logic through which the machine is adapted and trained to work remotely. That development led to less interference of the human mind in all the details. [2] [3] [4] [5] [6]

What was discussed in this research is a set of separate tasks for maintaining and upgrading different computer systems in an oil company. It was implemented independently, as the source code for the systems was not obtained. Only databases were available.

With reference to the ability data found in the presence of the computer. The highlight of the interface is the networking operations of large and heterogeneous databases. Finally, after collecting data in various forms, dirty data became part of any data. It is expected that with time and the succession of technical developments, dirty data will become a part of the past.

2.1. Network Age

Since its inception, the computer has adopted numerical systems as being digitally based. Later, attempts began to collect data on separate computers. After the success of networking, two related topics started evolving. First is accessing and dealing with information, which is done digitally. The second relates to communications media and data transfers. The then-existing analog technologies managed this part. Subsequently, the press was developed and converted to fully digital in the last two decades. The way data was formed and how it was transferred resulted in the different mechanisms and techniques adopted by the databases. How data is transferred also controls the databases' size and imposes speed limitations on them. Reaching or providing access to all devices was difficult when networks were launched first. This is contrary to the present situation. At that time, the number of devices that were not connected to the networks was a lot more than those which were connected.

The emergence of Distributed Systems in communication networks did not solve the problem either, and a significant percentage of devices remained isolated and inaccessible. However, such isolated devices may not be of paramount importance, as they may be used for printing purposes. Nevertheless, they were still considered information devices.

There were no thoughts of Data Science or linking it to Machine Learning and Deep Learning. However, data outside of databases or textual data has become a primary catalyst for interest in this data and its quality. The science that deals with such textual data experienced rapid development. Furthermore, data not present or stored on computers has also given rise to the science that addressed these issues. This included such data, whether it was before the laptop or was typed by hand.

2.2. The Computer Began as a Single Science, and Today It Is a Group of Disciplines

"*A scientific branch develops into several branches, and every branch has branches today.*"

The computer began with hardware and a set of programs in a machine language to run. A few years later, applications appeared within the computer, becoming independent disciplines later. Computer systems were one of the first applications where programmers tried to employ computers. Systems have emerged to control databases of businesses. The database was as close to how text files were managed in the Basic language. For example, all data was accumulated in one file that might exceed a hundred fields. Developers using programming languages could deal with a group of files with difficulties. These files were text files consisting of Symbols. System Analysis and Software Engineering overlapped to serve databases before they evolved and became independent disciplines. Programming languages were developed to work with databases. Some companies still use some of them, although some programming languages, such as PASCAL, have stopped growing.

The main reason some businesses continue to use such legacy systems is to save money. Instead of redesigning the system in a new programming language, some companies preferred to keep their old systems and link the outputs with modern designs. The reason is that redesigning new languages involved high costs, predominantly when prices were determined by the number of lines of source code written. Linking the output of older systems with modern designs was when the issue of unclean data started to appear. An example of dirty data is data stored textually on remote computers. Sometimes the transmission and maintenance of this data into databases constitute some incompatibilities. The development of computer science and the introduction of images and videos, for example, as an essential part of databases, especially in radio library systems or medical systems, have led to many more problems. Another example is that reports written by physicians were treated as images. Such issues in the past were represented by slowness and perhaps inconsistencies with the nature of databases that initially originated as textual information.

3. Dirty Data Caused by Analysis and Designs

There are two main reasons for dirty data.

3.1. Reason #1: System Analysis

System analysis has become a discipline that is taught and has multiple methodologies. If the same system is given to many analysts to analyze/design, then it will be found that they generally agree in their analysis and designs. However, their designs will not be seen as carbon copies. There are differences in the structure of entities, relationships, selection of master and child keys, and thus in the arrangement of database tables. The most crucial step is for a system analyst to understand how it works and not leave out any details to chance. Understanding the system is the basis for its design and construction. An excellent analyst chooses the right people in the system's working chain to question. Understanding requirements and outputs are also necessary. The depth of understanding of the analyst makes him/her innovate and develop solutions and additions to the system. Some analysts fall into the redundancy trap without even realizing it by building their schedules. Thus, the data is duplicated, and this is a negative thing. Alternatively, they do not choose suitable keys, or records contain a large amount of data at a time that can be hashed into files containing fewer fields.

3.2. Issues Caused by the System Analyst/Team Analysis

3.2.1. Errors by the System Analysts

Solving problems begins by breaking them down into groups of parts to make them easier to deal with, whether big or small. For example, the system to be automated is segmented into processors and subsystems. Moreover, each can be divided into secondary processors according to certain levels, as the SSADM methodology follows. The problems arise more in such cases if each processor is a single indivisible unit prepared by a homogeneous team and takes into account all solutions, from databases to networks, meaning when the work is not carried out as a whole but rather as parts.

In companies in general, system analysis depends on the resources and priorities of the company. Some companies require large systems to run in parallel. There are private companies that predate the database era. In such companies, the manual methods are automated successively, according to the requirements of one stage at a time.

These business requirements define the next step in succession systems. For example, an analyst builds an employee file system, which leads to a request to set up a payroll system. After completing the payroll system, we may need to design one that monitors costs. The costing system needs several sources of information, starting with the stores and what has been spent and ending with the library and its contents. Unfortunately, some system builders do not consider other systems related to the work they are analyzing. For example, a system analyzer sets up an employee file system, and a table like this acts as the master file and is the most important in the database. After other subsequent computer systems were created, some data was known to become sloppy.

3.2.2. Issues Underestimation of Analysis and Assignment of Non-Specialists

Once the system design is prepared by someone no longer available, management hires an excellent and reliable programmer as an outside contractor to add what is needed. System documentation may not be available. Sometimes it becomes Effortless to dedicate an extensive and expert operator to do this through other applications such as Excel or Access.

See Table 1 below.

Indeed, reliability is considered when conducting the initial study of the system. One of the reliability criteria is to ensure that the system's developers are close to the workplace and have the complete documentation of the system in place. However, the desire to build the system quickly may be the reason we mentioned in the previous paragraph as a problem that operators with great experience end up doing the work.

3.2.3. Problems Are Due to Frequent System Maintenance and Patching

In some businesses, there are outdated systems that decision-makers feel comfortable with, and such systems become difficult to abandon. Despite being outdated, those systems will eventually need to be developed. As more demands for updating increase, new requirements are included and must be part of this system. E.g., preparing new inputs and obtaining new outputs. To make matters worse, more often than not, developers of these systems have already left the company. Developers of the original system may be former employees of the company. They may have moved on to a new job. Or it was developed by an external private trading company that may or may not be around. Often, there is no working documentation and no source code files. There are only encrypted executable files that cannot be modified. Personnel updating the system must create a parallel program that links the available databases. This is done in specific files that function like any standalone computer system except for their data sources. If fields can be added, then it is done. This may cause Redundancy in the data and thus lead to dirty data.

4. Dirty Data Caused by Issues with the Data

Many types of dirty data can occur in such cases. Examples:

4.1. Incomplete, Inaccurate, and Outdated Data

Data lacks access to a key that connects it to the rest of the data. Preliminary data, either because it is unavailable to the operator at entry or data obtained from another system with a link to the keys. As for this system under study, the data

Table 1. Indicates inaccurate or incomplete data.

Employee No.	Employee Name	DoB
1003		1970/6/6
1005		62
1006		
1007		Fifty five
1290		1990/11/29

was incomplete because there was no key identifier. In addition, there was outdated data that may be unusable. Examples are: When an employee leaves work, his/her information remains available. This type of old data is archived and then deleted. Another example: There may be data related to competitors or suppliers that no longer exist. [7] [8]

These are a couple of fields from within a large database table. We note in the date field that there is no condition (or no format) setting for the date in the field; see **Table 1**. The defect may be due to the inaccuracy of birth dates for some, as we sometimes see them not mention the month and year. The programmer is expected to accept any date form. We find great difficulty calculating the average ages for predictions that benefit the employer's work. This difficulty causes a false mean to be returned if there is Null Data. Writing the age in different formats can be solved by complex manipulations.

The programmer can make the program restrict the entry to the date of birth in the form of a date of 8 digits. For example, the user should be notified using "January 1" for those who did not specify the month or day. For a person who wants the month of birth on his driver's license, the beginning of the month can be considered the day of birth. Here the error rate is almost non-existent.

An outdated data example is the records left in the system for an employee who has left the job for any reason. Such records can cause problems, the least of which is the continuation of his salary or failure to migrate to the archive. All records of terminated employees or similar cases must be transferred to the relevant archive in the computer system. For example, if a customer or supplier is no longer in the market for any reason, if his record is still in the system, this can cause problems (as mentioned in paragraph 4.1 above).

4.2. Duplicate Data

Several reasons cause data to be duplicated. First, wrong analysis of the system can cause this. For example, if the same field is found to exist in two different tables, it can lead to duplication. Another example is if it was determined that there was no need for a field to exist, but later on, it is recognized that it was necessary to include it in the system. The field then has to be added sometime in the future. Second, transferring data from different machines often leads to data redundancy. Finally, backup and data Recovery could lead to Redundancy. Regarding inaccurate data, the most critical example is that a numeric number is stored in a text field. Other examples: include partial phone numbers or incorrect zip codes.

In **Table 2**, it is noticed that there are two types of repetition. The first is a duplicate cost center (111) with the same Title. This is more dangerous than if the Title were the same. Such problems usually occur when using distributed computer systems. Each system has domains that are unrelated to other systems and are not repeated. Some schedules are the same across all distributed systems, such as the cost centers' schedules. When a new partition or unit is created, it is entered. This entry can be made in each device, causing Redundancy. For example,

Table 2. Redundancy in cost centers table.

Cost Center	Title
15	Computer Department
17	Computer Maintaince
23	Payroll Section
40	Marketing Unit
111	Internet Unit
125	Payroll Unit
126	Payroll Review Unit
111	Internet Unit

the most dangerous thing is going to the existing partitions and dividing them into modules. The department still exists, and its new units are also on the cost center's schedule. Here redundancy occurs that is not easily recognizable. So that the request sometimes holds the unit and the Section to which the unit belongs simultaneously.

It is noted in Table 2 that the cost center. No. 23 has been divided into two cost centers, No. 125 and 126, with the original center remaining, which is considered a repetition. To solve this problem, we canceled the original "Payroll Section" center with cost center 23.

4.3. Irrational Data

There is also irrational data, which takes different forms, including data placed in Excel tables but stacked in the same field. Whoever designed it might have been able to use it. If this data is combined with other tables, it will not be easy to deal with the problem. The second type can be seen when many fields are represented by non-relational data in texts, as in written reports, see Table 3 and Figure 2.

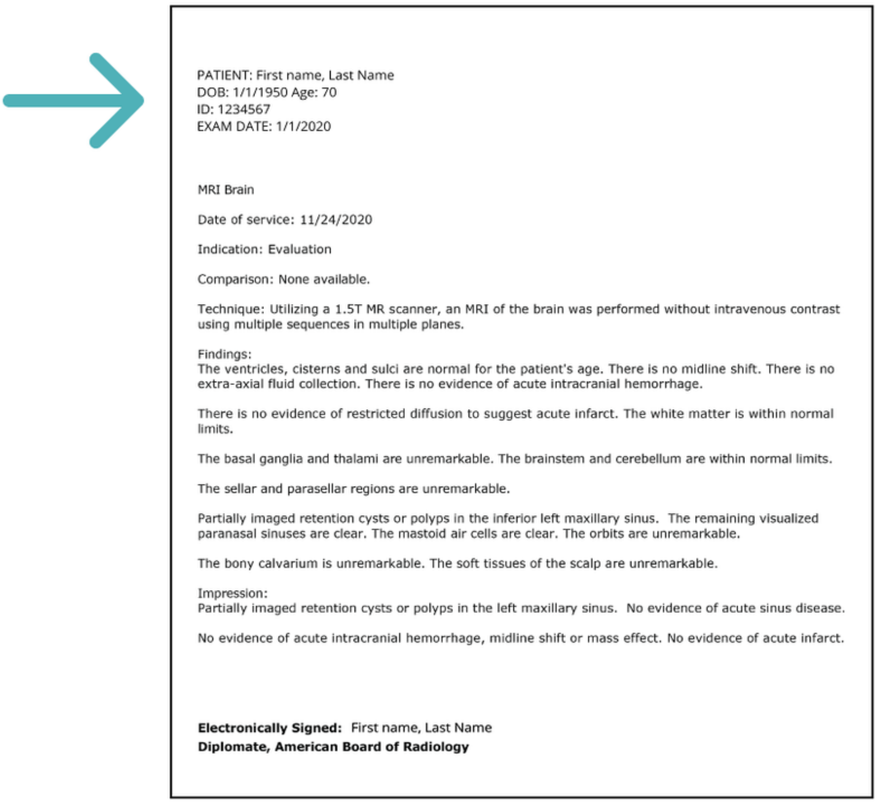
4.4. Wrong Data Format Data by the System Analyst or Programmer

In some databases, the field definition causes an error when requested. The user will be surprised by an unexpected outcome. In such databases, the programmer has to put all the required possibilities that give the same result. Fortunately, everything is defined in other databases, so the system rejects the request and sends an alert message to the user.

An example of this is shown in Table 4. If the cost center field is formatted as "Text" or "String" instead of "Numeric", then ordering based on the field will provide an ordering which is not expected by a user or sometimes even by a programmer. The reason is ordering numbers in string format yields different ordering of numbers in numerical format. See the original data in Table 4(a) and the ordered data based on the string field in Table 4(b), while Table 4(c) shows ordering based on a numerical field.

Table 3. Mixed Table between irrational data and relational data.

Purchases	Sales
70,000	2000



PATIENT: First name, Last Name
DOB: 1/1/1950 Age: 70
ID: 1234567
EXAM DATE: 1/1/2020

MRI Brain

Date of service: 11/24/2020

Indication: Evaluation

Comparison: None available.

Technique: Utilizing a 1.5T MR scanner, an MRI of the brain was performed without intravenous contrast using multiple sequences in multiple planes.

Findings:
The ventricles, cisterns and sulci are normal for the patient's age. There is no midline shift. There is no extra-axial fluid collection. There is no evidence of acute intracranial hemorrhage.

There is no evidence of restricted diffusion to suggest acute infarct. The white matter is within normal limits.

The basal ganglia and thalami are unremarkable. The brainstem and cerebellum are within normal limits.

The sellar and parasellar regions are unremarkable.

Partially imaged retention cysts or polyps in the inferior left maxillary sinus. The remaining visualized paranasal sinuses are clear. The mastoid air cells are clear. The orbits are unremarkable.

The bony calvarium is unremarkable. The soft tissues of the scalp are unremarkable.

Impression:
Partially imaged retention cysts or polyps in the left maxillary sinus. No evidence of acute sinus disease.

No evidence of acute intracranial hemorrhage, midline shift or mass effect. No evidence of acute infarct.

Electronically Signed: First name, Last Name
Diplomate, American Board of Radiology

Figure 2. Shows textual data, which is irrational. Retrieved from https://www.brainkey.ai/blog/how-read-brain-mri-radiology-report on 6/11/2022 at 12:30.

Table 4. (a) Original table; (b) Text/string ordering; (c) Numerical ordering.

(a)

Cost Center	Title
15	Computer Department
17	Computer Maintaince
23	Payroll Section
40	Marketing Unit
111	Internet Unit
125	Payroll Unit
126	Payroll Review Unit
111	Internet Unit

(b)

Cost Center	Title
111	Internet Unit
111	Internet Unit

Continued

125	Payroll Unit
126	Payroll Review Unit
15	Computer Department
17	Computer Maintenance
23	Payroll Section
40	Marketing Unit

(c)

Cost Center	Title
15	Computer Department
17	Computer Maintenance
23	Payroll Section
40	Marketing Unit
111	Internet Unit
125	Payroll Unit
126	Payroll Review Unit
111	Internet Unit

4.5. Other Types of Data Problems

There are other types of dirty data. Many researchers and authors give them different names, but most are similar to those mentioned above.

5. Previous Experience in Dirty Databases

System designer(s) can control who accesses the databases based on the implemented design. The designer defines specific access levels. For example, access to all database tables is reserved for higher levels of operators. Some users have limited access to data and tables, while others have different access levels. Such access levels depend on job title, position, or responsibility. These access levels are done within programming processes, provided that the databases are stored in a device that is not accessible to operators. This is also done for information security, including confidentiality, integrity, theft, and destruction. Database administrators can access all database tables. Everything that has been talked about regarding restricting access is on the programming side.

However, it becomes accessible if you get the database tables. There is one author's experience coping with databases managed by an executable file and not by source code. Several modifications to the system have been requested. The database is available, but the programs required to manipulate it are unavailable. In this case, the programmer can create solutions, although these solutions will eventually result in much dirty data. In such a case, the programmer creates his new system in which input, modification, deletion, and backup are made. When the rules are based on the system user query, these programs process the data and produce results in printed reports or displayed on the screen.

For example, when the Human Resources department requests a specific report, such a report may still have to do by the salaries section. It may also have to do with the Department of Transportation. A portion of the data may be in tabular form in a database. Another piece of the required data may be in a document in word tables and other Microsoft Office formats. There are many requests from the Human Resources department, for example. Some applications have data in the database available. In this case, the programmer has to design new tables and include that data in the database.

Senior management requested that the Human Resources department provide a report on allocating company vehicles or cars. When a vehicle is allocated, a printed paper and an electronic copy are also submitted to the Payroll Office. The purpose is to maintain an inventory of cars due to new jobs requiring obtaining a vehicle, yet no vehicle was allocated to that employee. Therefore, a directive was issued to grant a financial allowance to those eligible employees. The allowance is not of a fixed value. Instead, it is determined based on some factors and conditions. Another requirement was to forward this report to the costing department to determine the vehicle's depreciation premium. This process is carried out by a specialized employee who calculates the costs according to specific and transparent rules. For example, if the depreciation premium reaches a value of $1000, the vehicle is scrapped and sold in the employer's first public auction.

There was no vehicle management system at that time. On the other hand, some data in some tables could be used. After examining all the database tables, no fields related to vehicles in the company were found. This made it necessary to create a unique table for cars from which eligible users were allocated these vehicles. The key to this Table was the "employee's number", and a single field has been added. That is a unique vehicle number assigned to each vehicle. This field is used to link this Table with a vehicle information table. The "vehicle information table" Table contained several fields, including the Make, Model, Year, and other data like who used this vehicle and when. This vehicle number is set as a master key.

This allocation of company car reports requested by senior management led to the creation another small system for the company cars. Such requests and modifications are a fundamental reason for the dirty data that needed to be dealt with at later stages, especially when all these systems were later to be connected.

5.1. Database Example

In Table 5, the first row contains an entry: Table 1 is from the Human resource system, and the second is from the salaries system, which is repeated in the costing system. The same applies to the third Table, as it is like the second Table, while the fourth is from the warehouse control system.

At a glance, it was found that there is a need to add a field for the Cost Center in the Human Resource affairs system. When the data is migrated to the same

Table 5. Model of four tables from different systems collected after connecting these systems.

Table 1	Employee No	Employee Name	DoB	Department
Table 2	Employee No	Cost Center		
Table 3	Cost Center	Title		
Table 4	Cost Center	Employee No	Date	

database, most of the problems discussed earlier in this paper occur in both parts. Issues that appeared included a lot of Redundancy—the problem of Redundancy in the data. Moreover, there is also the problem of missing some data. The absence of which caused the poor quality of the data. This leads to the fact that it was not possible to get the results of some required queries. For example, how the fields were formulated was complicated, such that the first Table had a field for Full Name. The code made it possible to perform a text search and specify the first and last names. Nevertheless, this complex process consumed the processing time, increased the written code, and was time-consuming in execution. If it was wanted to go directly to the point, it was better to separate the first name from the last; instead, it was preferable to insert a middle name or other names field.

5.2. Prediction

The examples noticed while doing this work varied and overlapped with various data sources. Hence, this diversity complicates matters further to dirty data. Similarly, prediction can be difficult. In the company's vehicles example, valuable facts were known to predict the car's value by following the vehicle's life cycle. We understood if the problem was with the make and model of these vehicles or with specific drivers. Through routine maintenance operations, we knew which departments were the most car consumers/users. Since the nature of the company's work and the distance from its headquarters in the city dictated the presence of 10% of women, it is difficult to predict that a driver was a man or a woman, according to the rules of probability that stipulated that the sample should be random. Taking a random sample would have a higher tendency towards a male and may only show Males.

In the abstract, it was mentioned that such dirty data is primarily found in medical data. However, these data are diverse in form, text, relational databases, or images. Also, the methods of dealing with it and predicting it are different. Let us look at an example here:

For all inpatients/outpatients, general data such as height or weight, blood pressure, and other data are collected. In this case, if a random sample is taken, it may not give predictions close to reality. For example, male physical measurements differ from female measurements. Furthermore, the random sample does not provide a gender balance. So first, a selection was taken, for example, stratified sampling. Second, complex conditions were set for the prediction process.

For instance, following a patient's response to a particular drug in a hospital department was observed. Then the effect of this drug on many patients was followed. Finally, the requirement that patients' admission and discharge dates be in a comparable period and not in intervals was considered. We believe in observing factors such as pressure, pulse, age, and weight for each gender, others, and each value metric in the prediction process.

5.3. Dirty Data

Several models of dirty data were presented in the previous Section. A simple example is if an organization has stores, it is customary to find approximately 50,000 items, considering that some of the items may be the exact item (duplicated) but with different versions. Over a year, it is possible to find 10,000 records for supplying or withdrawing items when the movement is monitored to draw them. Depending on the clean necessary fields from any redundancy, you may have to build tables totaling up to 100 fields. With simple arithmetic, we have a billion records and fields. This is an alarming number. If the database relates to some hospital stores, it will be found that there is a group of stores in addition to drug stores in each hospital. One store may reach the drug according to its size and quality: oral, liquid, syringe, gas, and others, in addition to the companies from where the medicine was supplied. It may reach 100 types. Warehouse systems remain easy because they are programmed once with upgrades later, not many, and their data is homogeneous. However, the matter becomes more complicated if it concerns patients and their data. Many data types are found here: some are text and different images. The case details family doctors or the general clinic and its connection to distributed systems.

6. Conclusions

1) Unless the data is of high quality, the efforts employed during the analysis, design, and compiling may be wasted. The data also must be free of impurities and additions.

2) Some companies and employers keep old data. Rapid computer developments have led to the emergence of more up-to-date designed and programmed systems in keeping with the times. However, businesses insist on using their old data. The reason is that these parties do not want to waste time and money creating new systems for them and reintroducing or migrating them. As a result, it takes any required data from this old data, which is used in more recent designs. This causes problems like the ones we touched on in this paper. However, in many cases, cleaning data through the system is much better than using external programs.

3) There were some parties in the past when computers were expensive, and programmers were paid by the number of lines they wrote; they paid costly wages. As a result, some parties resorted to prioritizing the systems automation. It starts with programming one system. Then, they find it necessary to make

second and third systems. In the end, necessity compels them to connect these systems. Here problems arise with dirty data, incomplete data, or out-of-date data. Also, electronic archives are from a group of different or hybrid databases where the same problem appears. The irrational data often appearing in medical systems is also challenging. These problems can be dealt with by traditional methods followed in data science, manually, or by preparing algorithms for processing.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Laarfi, A. (2020) Life: A Huge Archive Electronic Archive Has Become an Urgent Necessity in the Face of Enormous Technological Advances. *Journal of Computer and Communications*, **8**, 1-10. https://doi.org/10.4236/jcc.2020.84001

[2] Laarfi, A. (2022) Biometrics as a Matrix: The Short Distance between Crime and Security Systems, Prompting an Artificial Intelligence to Invent Electronic Biometrics ID! *International Journal of Intelligence Science*, **12**, 1-8. https://doi.org/10.4236/ijis.2022.121001

[3] Laarfi, A. and Kabuska, V. (2022) Framework for Reasoning with Speech Processing. Services for Science and Education, Cheshire.

[4] Laarfi, A. and Kepuska, V. (2020) Implementation of a Verbal Compiler: The Need to Develop Audio Language to Keep Pace with Rapid Development Becomes a Necessity. *Global Journal of Human-Social Science*: (*G*) *Linguistics & Education*, **20**, 1-12. https://doi.org/10.34257/GJHSSGVOL20IS4PG1

[5] Laarfi, A. and Kepuska, V. (2020) Constructing a Simple Verbal Compiler. September. *International Journal of Intelligence Science*, **10**, 83-91. https://doi.org/10.4236/ijis.2020.104006

[6] Weber, H. Big Data: A Complete Guide to the Basic Concepts in Data Science, Cyber Security, Analytics and Metrics (Big Data and Artificial Intelligence). Book 1, a Kindle Version.

[7] Sud, K., Erdogmus, P. and Kadry, S. (2020) Introduction to Data Science and Machine Learning. https://www.google.com/books/edition/Introduction_to_Data_Science_and_Machine/BXqozQEACAAJ?hl=en&gbpv=1&printsec=frontcover https://doi.org/10.5772/intechopen.77469

[8] Stanton, J. M. and Saltz, J. S. (2017) An Introduction to Data Science. SAGE Publications, Inc., Los Angeles, CA.