



NOVA

IMS

Information
Management
School

MAAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Data Profiling in Cloud Migration

Data Quality Measures while Migrating Data from
a Data Warehouse to the Google Cloud Platform

Andreia Filipa Gonçalves Cabral

Internship Report presented as the partial requirement for
obtaining a Master's degree in Data Science and Advanced
Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

DATA PROFILING IN CLOUD MIGRATION

Data Quality Measures while Migrating Data from a Data
Warehouse to the Google Cloud Platform

by

Andreia Filipa Gonçalves Cabral

Internship Report presented as the partial requirement for obtaining a Master's degree in
Data Science and Advanced Analytics

Advisor: *Prof Doutor* Flávio Luis Portas Pinheiro

External Advisor: Pedro Santos Figueira

DEDICATION

Para o Fernando e a Teresa, o pai e a mãe que me ensinaram o certo e o errado, o bom e o mau do que a vida tem para oferecer. Tudo o que fiz e farei será sempre por e para eles.

Para o João, o irmão que me faz querer ser o exemplo e ser sempre melhor.

ACKNOWLEDGEMENTS

Quero agradecer ao meu namorado Vitor pelo amor, apoio e motivação ao longo de todo este tempo. Obrigada por me ajudares a ver o copo sempre meio cheio.

Agradeço, também, à Inês, Vanessa e Carolina pela sua amizade incondicional, e à Marta por-me ter acompanhado ao longo deste caminho.

Finalmente, agradeço ao meu orientador Flávio Pinheiro pelos ensinamentos e compreensão, e ao Pedro Santos Figueira pelos conselhos e atenção desde o meu primeiro dia na empresa.

ABSTRACT

In today times, corporations have gained a vast interest in data. More and more, companies realized that the key to improving their efficiency and effectiveness and understanding their customers' needs and preferences better was reachable by mining data. However, as the amount of data grow, so must the companies necessities for storage capacity and ensuring data quality for more accurate insights. As such, new data storage methods must be considered, evolving from old ones, still keeping data integrity. Migrating a company's data from an old method like a Data Warehouse to a new one, Google Cloud Platform is an elaborate task. Even more so when data quality needs to be assured and sensible data, like Personal Identifiable Information, needs to be anonymized in a Cloud computing environment. To ensure these points, profiling data, before or after it migrated, has a significant value by design a profile for the data available in each data source (e.g., Databases, files, and others) based on statistics, metadata information, and pattern rules. Thus, ensuring data quality is within reasonable standards through statistics metrics, and all Personal Identifiable Information is identified and anonymized accordingly. This work will reflect the required process of how profiling Data Warehouse data can improve data quality to better migrate to the Cloud.

KEYWORDS

Data Quality; Data Profile; Database; Data Warehouse; Cloud; Data Migration; Pandas Profiling; Personal Identifiable Information;

INDEX

1. Introduction	1
1.1. Company overview.....	2
1.2. The team and Activities.....	2
1.3. Internship Goals.....	3
2. Theoretical Framework.....	4
2.1. Data Storage	4
2.1.1. Database	4
2.1.2. Data Warehouse	5
2.2. Cloud Migration	6
2.2.1. The Migration Process.....	6
2.2.2. The Advantages to a Cloud Migration	8
2.2.3. The Challenges to a Cloud Migration	9
2.3. Data Quality.....	10
2.3.1. Bad Data	10
2.3.2. The Quality in Data	11
2.3.3. Data Types.....	14
2.3.4. Metrics for Data Quality.....	16
2.4. Data Profile.....	18
2.4.1. Structure discovery or Column Profile.....	19
2.4.2. Relationship discovery or Multi-column Profile.....	20
2.5. Data anonymization.....	21
3. Tools and Technology	24
4. Data Profile: The Project	29
4.1. The Planning.....	30
4.1.1. Tool Assessment.....	30
4.1.2. The strategy	33
4.2. The Implementation.....	34
4.2.1. Shell.....	34
4.2.2. Python.....	36
4.3. The Results and Conclusion	39
4.3.1. HTML Report.....	39
4.3.2. Excel Report	44
4.3.3. Main Conclusions.....	46
4.4. Useful Applications	48

5. Conclusions	50
6. Bibliography	51

LIST OF FIGURES

Figure 1. WS3 Teams Schema	3
Figure 2. Data Warehouse Inputs & Outputs	6
Figure 3. Project example of an XML file - A semi-structured data	16
Figure 4. Talend Profiling Modules	26
Figure 5. Functional Architecture	34
Figure 6. Dictionary format of Regex Patterns Examples	37
Figure 7. HTML Report Overview - Case Study	39
Figure 8. HTML Report - Warnings Tab	40
Figure 9. HTML Report - Information for Categorical Variables	40
Figure 10. HTML Report - Common Values for Col_3	41
Figure 11. HTML Report - Chart for Col_3.....	41
Figure 12. HTML Report - Information for Numeric Variables.....	42
Figure 13. HTML Report - Statistics for Col_4.....	43
Figure 14. HTML Report - Correlations Feature	44

LIST OF TABLES

Table 1. Accuracy Example in Date Birth	11
Table 2. Completeness Example in Address	12
Table 3. Consistency Example in Address	12
Table 4. Timeliness Example by Updating Address	13
Table 5. Uniqueness Example by Having Duplicate Records	13
Table 6. Validity Example in Date_Birth.....	13
Table 7. RDBMS Table Example - Structure Data Type.....	15
Table 8. DQ Dimensions - Definition and Metrics	17
Table 9. Illustrative Example of Client Information – Entry Date is a Control Field.....	17
Table 10. Columns Properties to be Verified	19
Table 11. Randomization Technique Example in Full_Name	22
Table 12. Pseudoanonymization Technique Example using Hash function SHA256 in Fiscal Number.....	23
Table 13. Suppression Technique Example in Address and Fiscal Number	23
Table 14. Generalization Technique Example in Date Birth	23
Table 15. Tools Assessment Pros and Cons.....	31
Table 16. Tools Functional Requirements Estimation	32
Table 17. Computing time of a 30 GB Feed	38
Table 18. Excel Report – Overview	45
Table 19. Excel Report - Features Summary	45
Table 20. Excel Report - Regex Validation Results.....	46
Table 21. DQ Dimensions - Definition, Metrics and Results	48
Table 22. PII Examples & Anonymizations and Normalizations Functions	49

LIST OF ABBREVIATIONS AND ACRONYMS

DWH	Data Warehouse
ON	On-Premises
GCP	Google Cloud Platform
PII	Personal Identifiable Information
DQ	Data Quality
DP	Data Profile
CSP	Communication Service Providers
OSS	Operation Support Systems
BSS	Business Support Systems
WS	Work Streams
BAU	Business as Usual
DB	Database
Dmaap	Data Movement as a Platform
EIM	Enterprise Information Model
ML	Machine Learning
DBMS	Database Management System
CRM	Customer Relationship Management
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
AWS	Amazon Web Services
VMs	Virtual Machines
UIs	User Interfaces

APIs	Application Programming Interface
NIF	Fiscal Identification Number
RBDMS	Relational Database Management System
BI	Business Intelligence
ETL	Extract, Transform and Load
DA	Data Anonymization
NER	Named Entity Recognition
EDQ	Oracle Enterprise Data Quality
PP	Pandas Profiling
CPU	Central Processing Unit
POC	Proof of Concept
VE	Virtual Environment
RDD	Resilient Distributed Dataset
DC	Data Custodian
Q1	First Quantile
Q3	Third Quantile
IQR	Interquartile Range
CV	Coefficient of Variation
MAD	Median Absolute Deviation

1. INTRODUCTION

Over the years, technology has taken a massive place in our lives and the world. A single cellphone can generate an enormous amount of data (MobiThinking, 2012), and as the usage of technological devices has grown exponentially, so has the information produced by each one. Nowadays, more than ever, information is power. Thus, companies realized that the key to improving their efficiency, effectiveness and understanding their customers' needs and preferences was reachable by mining data. (Atkins et al., 2003). A large amount of data requires special care and, above all, storage. Most companies build their storage solution in Data Warehouses (DWH) (Imhoff et al., 2003), a centralized data repository with integrated data from several sources and a variety of workloads, including customer satisfaction, financial, logistical, and historical info, with the primary intent of storage and a processing platform for reporting, analytics, and many other features. Traditionally, due to the necessity of physical hardware, and consequently, purchases, deployment, and maintenance, it is considered an On-Premises (OP) solution. However, what happens when the storage capacity reaches its maximum capacity?

As companies grow, so does the data. To increase the DWH storage space, the company must purchase more hardware infrastructure, leading to higher maintenance and possible upgrade costs. If storage requirement space is nothing but a seasonal high, then the newly added infrastructures are of no use, making the DWH solution unscalable. Therefore, there is now a more efficient and good price/quality ratio alternative for large or fast-growing companies – Cloud DWH Platforms, for example, Google Cloud Platform (GCP). In the software as a service approach, with no need for physical hardware, companies can increase or decrease their storage space and computing power on-demand, making it a scalable solution. The ability to take seasonality into account leads to a faster and higher performance at a lower cost. Because most corporations have their systems based on a DWH solution, to migrate all the different types of data to the GCP (or other Cloud Platforms) requires a combination of sub-processes. Each sub-process deals with a certain level of complexity and oversees handling a crucial part for the migration itself can be successful. One of the many challenges faced in the migration process in a massive company is ensuring the migrated data on the Cloud has no Personal Identifiable Information (PII) exposed, as required by law. To avoid so, data must be classified as a PII and later anonymized before migrated into the Cloud. Because we are dealing with corporate data, it is very often common to exist incoherence among the data, making the PII classification task harder (Zhang et al., 2003). To better analyze the types of information stored, Data Quality (DQ) (R. Y. Wang, 1996) analysis is of significant importance, not only for accurate data anonymization but for futures analysis and modelling for when the data is already available in the Cloud, thus leading to better results for decision making. To help ensure DQ, most often, a profile is designed for the data available in each data source (e.g., Databases, files, and others) based on statistics, metadata information, and pattern rules -this

process is called Data Profiling (DP). This work will describe how profiling DWH data can improve DQ to better migrate to the Cloud.

1.1. COMPANY OVERVIEW

Company_A, whose anonymity will remain throughout this work, is one of the largest Portuguese Information Technology companies, focusing on developing and implementing solutions that aid Communication Service Providers (CSP) undergoing the digital revolution. It was founded in 2000 and specialized in IT Operation Support Systems (OSS) and Business Support System (BSS), connecting the business world with technical answers, thus minimizing IT projects' risk, meeting the client-centric architecture with the technology available in the market or created on-premises. So, Company_B is one of the clients from Company_A who required an alternative solution for their DWH infrastructures, which led to the migration from DWH to the GCP project.

1.2. THE TEAM AND ACTIVITIES

The team responsible for designing this project is quite numerous and divides itself into seven Work Streams (WS), each responsible for a specific part of the project. These are the following:

- WS1: Functional documents and data model's design.
- WS2: Platform Engineering.
- WS3: Data Sourcing to GCP (Local Data Extraction/Provisioning).
- WS4: Modelling & Pipeline (Data Engineering, Data Migration, Test).
- WS5: Visualization (Insight Products and Machine Learning Models).
- WS6: Target Operational Model.
- WS7: DWH Decommission.

Data in the DWH is being consolidated over the last 20 years, creating a high data volume from different Company B business areas; as such, all data is organized into tables, also referred to as feeds, or files (positional or delimited) with information from several business areas and customers. To improve performance and operational manageability, feeds are partitioned into nine releases, in which release divides itself into Migration and Business as Usual (BAU) feeds. Migration feeds are associated with historical data; thus, Migration feeds are only processed once migrated to the GCP for each release. However, BAU feeds are daily load processes, referring to all reliable data from previous alterations (inserts or updates) taking place on the operational systems, thus often requiring configuration to ensure data updates in the DWH on a daily basis and needing to be processed in multiple occasions into the GCP.

On WS3, where this report was currently based, the main goal is to move Migration and BAU feeds from separate Database (DB) from DWH to GCP. It is divided into three significant teams: Extraction,

Data Movement As A Platform (Dmaap), and Google Storage. The Extraction team is responsible for replicating BAU feeds with extra control fields through a DB procedure, adding the NGBI prefix to distinguish the originals from the copies. Once the copy is completed in the Dmaap team, those feeds are processed through an ETL pipeline created in Talend to generate an XML file with the feeds metadata. As follows, the Cloud team will use the generated XML file to process it in the GCP Data Fusion pipeline to be deployed to the GCP so that it can be transformed into the new Enterprise Information Model (EIM) (Figure 1). Additionally, WS3 performs Data Profile reports for a better understanding of what underlines the data.

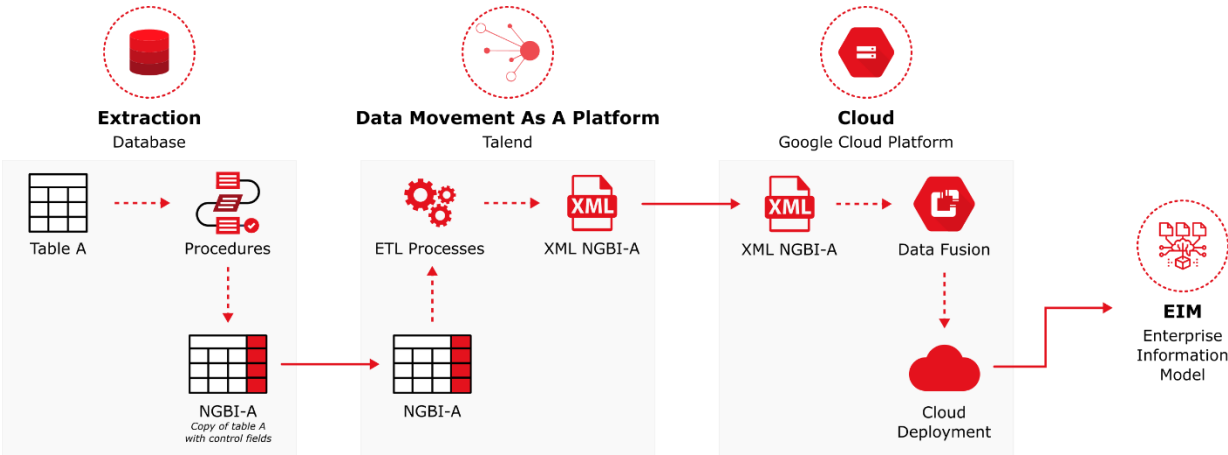


Figure 1. WS3 Teams Schema

1.3. INTERNSHIP GOALS

Company_A’s primary goal is to ensure the migration process, focusing on improving and optimizing every step along the way. Often, the existing sub-processes need adjustments and improvements, or new processes need to be created to answer new challenges that may appear along with the project. Therefore, the assigned role was firstly evaluating the viability of creating a DP process according to the resources Company_A could provide and what was expected. Secondly, to build the processing pipeline, using Python and shell scripting, with the teammates’ help. Lastly, perform evaluation metrics and criteria to ensure the DP analysis was accurate enough and some improvements based on the assessment. This sub-process is transversal to the many teams referred to earlier, using the functional documentation and data model’s design, PII’s anonymization, testing, and upcoming Machine Learning (ML) modelling and reporting.

2. THEORETICAL FRAMEWORK

Data is present in our day-to-day; it is encountered in every corner of our lives now. Ignoring its importance is thoughtless, which means a decrease in profits or even unwanted corporate organizations' costs. In a more competitive environment, the proper use of data - how fast it can be accessed, how well it can be mined, how accurate it is - can make a company thrive in its achievements with valuable insights. In this new wave of technology, those that place their organizations under fast and correct data management will more likely lead the way. This chapter will describe the underlying processes of a DWH migration to the Cloud; additionally, it will detail DQ's relevance and techniques to assess DQ – Data Profile.

2.1. DATA STORAGE

2.1.1. Database

Over the years, companies stored their data in various ways. It started with the basic approach, pen and paper and went from excel sheets to what is the most used method nowadays – a Database. A DB is an organized data collection stored in electronic format in a computer system. Accessing this DB requires an application that allows the users to retrieve and manage data efficiently – Database Management System (DBMS) (Elmasri & Navathe, 2016). Some capable applications are Oracle DB, MySQL, Sybase or MongoDB. As Elmasri and Navathe express in their work, “*The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.*”. To define a DB, one must designate the data types, structures, and constraints of the database's data, consequently generating a DB catalogue or dictionary with all descriptive information - the meta-data file. After, data needs to be stored in a physical device capable of being managed by the DBMS, thus constructing it. The DB can then be manipulated by retrieving, updating, or generating explicit data through application commands - querying. Finally, sharing a database allows for simultaneous access to the database by several users and systems (Elmasri & Navathe, 2016).

Various types of DB models exist, such as the Hierarchical, Network, Entity-Relationship or NoSQL model. However, the Relational DB model is the most used type, storing data points linked to each other (Elmasri & Navathe, 2016). The relational model, an intuitive, transparent way of viewing data in a tabular format, is based on relational databases. Each row in the table in a relational database is a record that has a unique ID called the key. Table columns contain data attributes, and each record typically has a value for each attribute, making the relationships between data points simple to define. Organizations of all types and sizes use the basic but efficient relational model for a wide range of information needs. Relational databases are used to monitor inventories, manage e-commerce orders and control vast volumes of mission-critical customer data. For any knowledge necessity in which data points refer to

each other, a relational database may be considered and maintained in a stable, rules-based, coherent manner. Since the 1970s, relational databases have been around, and their benefits continue to make it the most commonly recognized database model.

Although the NoSQL model is starting to gain attention in the corporate world, due to technology advances and Big Data applications, Relational DB was less suited to manage the rapidly rising volumes of data and the increasing complexity of data structures. NoSQL databases are non-tabular and store information differently from relational tables in a different manner. Based on their data model, NoSQL databases come with several forms: files, JSON, key-value, broad-column, and graph. With massive volumes of data and heavy usage loads, their schemas are quickly scalable. The complex schema of NoSQL databases enables the use of what is known as “unstructured data.” Enabling to construct the framework without needing to specify the schema firsthand. In a relational database, before applying data to the database, the schema must be specified. Non-relational NoSQL databases have become more prevalent in the last decade to provide a more modular, scalable, cost-effective alternative to the conventional relational model (Elmasri & Navathe, 2016).

2.1.2. Data Warehouse

A database was defined as an organized data collection stored in electronic format in a computer system in chapter 2.1.1. A Data Warehouse is also a data collection, an electronic method of organizing information. However, unlike a DB, a DWH is designed exclusively for decision-support applications, retrieving data and not conventional transaction processing. Additionally, a DWH consolidates an immense amount of information from several sources into one extensive system. It can include databases from different data models and even files obtained from individual networks and platforms. For example, it can gather information about various business areas, like employee records, customer information, and website data (Figure 2). An organization will evaluate its clients more comprehensively by integrating all of this data in one location, guaranteeing that it has considered all available information. DWH also enables data mining analysis by searching for data trends that can increase revenues.

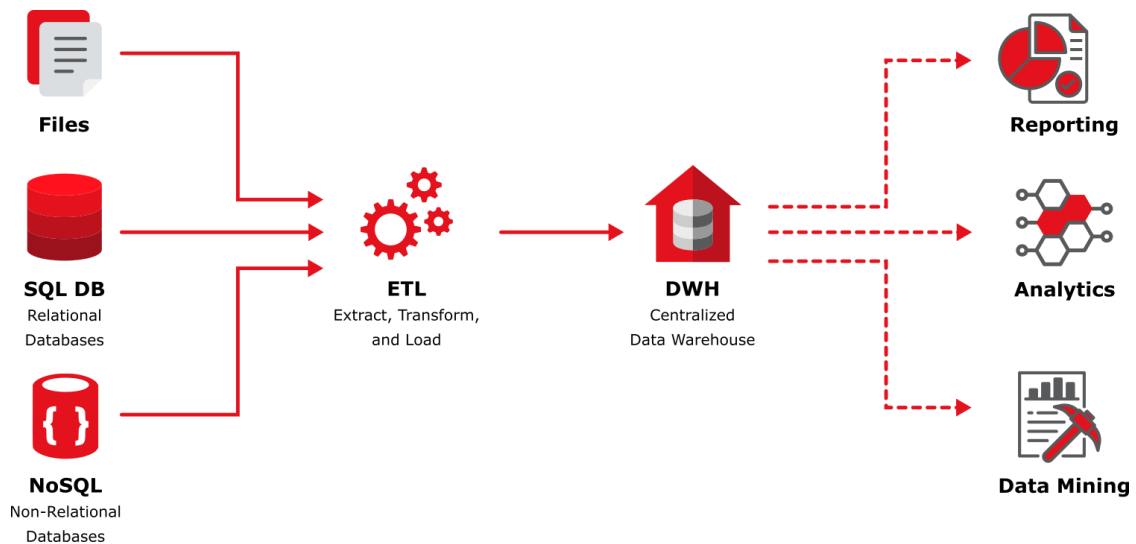


Figure 2. Data Warehouse Inputs & Outputs

2.2. CLOUD MIGRATION

Nowadays, businesses are answering complex business problems that are data-driven in high volume. However, they are often held back due to their data platform infrastructure. Data platform architectures designed in the 1990s are not ready to solve business problems for the twenty-first century, much regarding the explosive data growth for businesses worldwide. Ever faster and more extensive data streams, global business needs and requirements for their IT teams to address the issue quicker and with more agility. Most legacy DWH cannot keep up anymore, so companies tend to turn to a more elegant and efficient solution – the Cloud. Cloud Migration is the mechanism by which data, application code, and other business processes related to technology are transitioned from an on-premise or legacy system to the cloud environment. As a new and modern way for computing, the Cloud has both scalable and virtualized resources dynamically provided as a service. With cloud computing technologies, customers can access programs, storage, and application development tools over the internet through their available technology such as laptops, tablets, and smartphones. Being a new trend on the market, it is believed that cloud computing will reshape the IT business and the world (“Handb. Cloud Comput.,” 2010).

2.2.1. The Migration Process

Migrating data is the process of moving data, applications, and other business elements from on-premises computers to the Cloud - a virtual environment of on-demand, shared resources that provide computing, storage, and network services at scale. The correct data migration approach is an essential part of the cloud migration planning process, and it should be considered from the outset (Z. Wang et al., 2020). The following steps will describe in more depth the basic steps to take into account when considering migrating data to the Cloud.

PLANNING. The first thing to be considered is the motive and explicit purpose of migrating a company's data to a cloud computing environment to outline the best possible strategy. Starting by assessing the current environment, it must be selected what should or not be migrated, the necessary data to execute this migration's primary purpose (Fahmideh et al., 2019). When moving an application from an on-premise data centre to the Cloud, it can be made as shallow cloud integration, also known as a "lift-and-shift", for the data is migrated as is, with none or minor alterations. Additionally, contrary to a shallow cloud integration, a deep cloud integration can be considered, on which the application is modified to benefit the Cloud capabilities.

Furthermore, it is possible to perform an Online migration, in which data is moved across the internet or a private connection (e.g. XML file), or an Offline migration, in which data is conveyed via a storage device that is physically dispatched between its data centre of origin and the destination cloud storage location.

CLOUD ENVIRONMENT CHOICE. The next step would be to decide what sort of cloud model will be more suitable for business needs. It may either be single or multi-cloud. A single cloud environment is created by having a single cloud provider serve all of the applications or services that the company expects to move to the cloud. Single cloud environments can use either private or public clouds, depending on which best meets their current and future requirements. This solution allows companies to migrate workloads to the cloud as their needs grow, with the ability to scale up the number of virtualized servers if their needs surpass a single cloud server's capacity. Typically, businesses with a single cloud model use the cloud for a single service or program, e.g. Customer Relationship Management (CRM).

However, a multi-cloud environment is when a business uses several public cloud services, often from different vendors. To achieve best-of-breed results or minimize vendor lock-in, various cloud solutions can be used for various tasks, since not all clouds are designed equal. For instance, Marketing and Revenue likely have different needs than Software Development or Finances, and diverse cloud solutions can satisfy those requirements more effectively. Multiple clouds also reassure companies by reducing dependency on a single vendor, lowering costs, and increasing versatility.

Some of the possible choices of cloud computing services are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) (Zhao & Zhou, 2014). Infrastructure as a Service (IaaS) - refers to cloud infrastructure systems composed of highly scalable and automated computation resources. IaaS allows companies to buy resources on-demand and as-needed rather than buying hardware outright. It allows them to access and track items like compute, storage, networking, and other infrastructure-related services. The leading vendors for these services are Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, and IBM Cloud.

Platform as a Service or cloud platform systems offer cloud components to specific apps and are primarily used for applications. PaaS provides developers with a platform on which they can design and customize applications. The enterprise or a third-party vendor can handle all servers, storage, and networking, while the developers can manage the applications. A few leading vendors for these services are AWS Elastic Beanstalk or RedHat.

Lastly, Software as a Service is a widely used choice for businesses. SaaS makes use of the internet to provide applications to consumers that a third-party vendor operates. The majority of SaaS applications are accessed via a web browser and do not need any client-side downloads or installations. Some prominent vendors are Salesforce, ServiceNow or Google Apps.

MIGRATION. Depending on the size of databases and applications, it will differ in the techniques used for migration. If the amount of data and applications is relatively small, it can be easily migrated as shallow cloud integration. However, it might lead to higher transfer times or charges from the cloud provider for more complex applications and a significant data amount. A good practice to mitigate this risk is to design a multiphase project plan segments work into multiple manageable pieces, each with a realistic technical goal adds business value.

Moreover, Data Quality issues, data modelling, and metadata should be corrected before or during migration. Otherwise, current data problems will continue to exist in the new platform. Ultimately, it is critical to maintain security during the migration by encrypting sensible data and having strict access by managing permission rights.

2.2.2. The Advantages to a Cloud Migration

SCALABILITY. Organizations are explicit on their computing and space needs at the initial level. However, when at a given point, and those needs do not meet the necessary ends, it is much more comfortable to reduce or increase their computing needs. Nevertheless, attending to those needs means going to run out of money rapidly. Via virtualization, scalable cloud architecture is viable. Virtual machines (VMs) - a virtual environment that runs as a virtual computer system, are incredibly scalable and can be quickly scaled up or down, unlike real machines whose assets and performance are relatively fixed. They can be migrated to a separate server or hosted at once on several servers; workloads and applications can be relocated as required to larger VMs.

COST REDUCTION. As with all on-prem systems, DWH, with the related infrastructure and licensing expenses and continuing systems engineering, pay for technologies. If the company is heading towards being data-driven, it will continue to ask for more information. Cloud offers much more cost flexibility, meaning the need to pay for or manage the entire underlying infrastructure stack does not exist. Business can increase their savings, for there is no need to spend money on expensive equipment and systems,

using the Clouds servers and infrastructures capacity, only spending on what they use due to Cloud scalability.

HIGHER MOBILITY. Across the globe, cloud-based data and services are available. So, employees may do their duties from almost anywhere through their smartphones, laptops, or tablets. When operating at the office or home, visiting customers at their offices, or doing other on-field operations, data is accessible at any given time.

PREDICTIVE ANALYTICS SOLUTIONS. Legacy data centres typically fail to keep up with everyday data demands, such as supplying divisions such as finance or sales with reports. It can be challenging to picture having the time and money to start performing predictive analytics when employees are held back by provisioning and computing constraints. With some functionalities, it is possible to shift data analytics work into many more users' hands. Without transferring data or using a third-party method, these large-scale computing opportunities save time and workload and enable corporations to pursue new avenues of progress.

EFFICIENCY. When companies transition into fully digital services, corporate efficiency is the main focus. Take organizations like online retailers with e-commerce and the demand for a competitive environment offering scalability and high-performance tools and extraordinarily reliable and usable software and data (Carroll et al., 2011). Cloud computing tackles mass data storage and mighty computational task, efficiently addressing problems quickly (Tari, 2014). To learn how to support clients properly, an organization must process and interpret data easily to enable its internal teams to perform their best job using the best available data. All these significant, cutting-edge innovations reflect the cultural and technical change, where efficiency is a must.

2.2.3. The Challenges to a Cloud Migration

DATA SECURITY. Among the most doubtful aspects of cloud computing are anonymity and security. A series of interviews conducted by Mariana Carrol and Paula Kotzé show that around 91,7% of the interviewers believe data security is a critical risk in cloud computing. Data is no longer controlled by administration and is open to vulnerability. Application and data hosting in shared infrastructures increases the risk for unwanted access and poses problems such as safety, identity protection, authentication, enforcement, privacy, honesty, data storage, confidentiality, protection of the network, and physical security (Carroll et al., 2011). By adopting a cloud storage framework, corporations with cloud computing servers are entirely entrusted with data protection and confidentiality. User Interfaces (UIs) and Application Programming Interface (APIs) are how users interact, monitor, and manage cloud services. Thus, cloud security relies on those APIs and UIs security; they must be designed to shield against unforeseen and malicious attempts in that order. Therefore, setting controls to solve security challenges is an essential step in protecting the cloud world.

MIGRATION PROCESS. Migrate a fully operational DWH is a high maintenance project, with the possibility of taking a long time depending on the data quantity being relocated. Blindly adopt new technology into the company's infrastructure without a solid strategy design can be a significant risk (Zhao & Zhou, 2014). Before the migration process starts, one must choose an adequate cloud provider for the business and estimate cost analysis, forecasted downtime, employee training, and an approximated time to complete the migration. While corporations are eager to migrate their data to the Cloud, not migrating everything at once is wise, mainly if it is a considerable amount of data. So, corporations should distribute data into blocks and migrate block at a time, beginning with non-essential or redundant data, in order to test the entire process, preventing compromising sensitive information due to staff error or process malfunction.

2.3. DATA QUALITY

As mentioned earlier in this work, companies' decisions are more often influenced by their data analysis insights. In large organizations, data also comes in large quantities, but more times than so, quantity those not necessarily correlates with quality. If companies based their choices and future growth on their data, having incorrect and inaccurate information will harm them. Data Quality indicates how reliable, helpful, and accurate to a given context data can be, helping prevent poor decision making, possible waste of resources, and profit decrease (Karr et al., 2006). The success achieved in the making business today and tomorrow will depend and be determined by the quality of the data collected, stored, and analyzed.

2.3.1. Bad Data

It is critical to know the downfalls when Data Quality analysis is overlooked. Because in big organizations, time and speed are of the essence, they work quickly to collect and use information in near real-time, this fast chain of events can lead to dependence on incomplete, redundant, and inaccurate dataset resulting in future inaccurate numbers and metrics, and finally an inaccurate decision. It can implicate high costs for the company; according to IBM¹, in 2016, companies lost \$3.1 trillion due to lousy quality data. On average, 47% of newly obtained data have at least one significant error, potentially harming the organization². Furthermore, as stated in MIT³, wrong Data can cost companies between 15%-25% of total revenue.

¹ <https://www.ibmdatahub.com/infographic/four-vs-big-data>

² <https://hbr.org/2017/09/only-3-of-companies-data-meets-basic-quality-standards>

³ <https://sloanreview.mit.edu/article/seizing-opportunity-in-data-quality/>



2.3.2. The Quality in Data

Data Quality can be defined as the level of compliance between a dataset and the contextual normality, taking into account various aspects, such as semantic rules incoherences or misspelt errors – consistency-, stored data corresponding to the real-world values -accuracy-, the amount of empty data present in the data collection -completeness-, measure the unwanted duplicate information -uniqueness-, and consider if the data age instore is suitable in the currents days -timeless-. It is tangible by a set of logical rules established by users, reflecting, for example, business processes or corporate know-how. The same entity can have different validation rules according to the organization or area that has been taking into consideration; for instances, Gender information in a financial company may be considered irrelevant, however for medical studies can provide insightful information, as the male body differs from the female, as so the Gender features should be more consistent. Another set of rules to assure DQ, besides business-defined, are statistically derived. Statistical information, just as the mean, standard deviation, correlation, and graphic illustrations, can better understand the analysis. The statistical approach will be described in a few chapters ahead.

Some DQ principles mentioned the pillars on which data values and their quality can be evaluated. Those principles are identified as DQ dimensions. Over the years, and the many studies on the DQ subject, the dimensions increased and mutated; 1999 Alexander and Tate (“Web Wisdom: How to Evaluate and Create Information Quality on the Web,” 2000) considered six dimensions – authority, accuracy, objectivity, currency, audience, and connections for DQ, in 2005 Knight and Burn (Knight & Burn, 2005) concise the most common and used dimensions for DQ frameworks (Cai & Zhu, 2015). The dimensions considered for this work are summarized as follows:

ACCURACY. Refers to the proximity of a measured value to a real or standard one that is considered correct, “the extent to which data are correct, reliable and certified,” as Wang claimed (R. Y. Wang, 1996). A DQ context refers to whether the data values stored to match the “real-world” values. Accuracy can diverge into two types: syntactic and semantic. A syntactic accuracy considers the grammatical structure relationship between the stored data value and its domain, while semantic accuracy regards the meaning given to a data value (Shanks & Darke, 1998). However, DQ methodologists only consider the syntactic approach (Batini et al., 2009) (Table 1).

Table 1. Accuracy Example in Date Birth

	NAME	ADDRESS	NIF	DATE_BIRTH
	CLIENT_A	101 STREET	123000000	03/03/1996
	CLIENT_A	101 STREET	123000000	01/01/1850

COMPLETENESS. The extent to which a data set has sufficient information to describe the corresponding “real-world” objects, thus fulfilling its comprehensiveness expectations. However, completeness can occasionally be a relative term, for what is complete for some users may not be complete for others, for instance, the same amount of information about a particular product may not be enough for Research Department who requires a detailed product view, whistle for Marketing because they only need more high-level information it is sufficient. From a DB perspective, this dimension is often associated with *null values*, an entry in the DB that has not been fill, intently or not; the *null* value may not exists or may not be identified either exists or not (Batini et al., 2009). Nevertheless, it is of great importance to understand its absence. An empty entry can also be encountered in an entire row or at the column level. A column can be X percentage of null values throughout, resulting in bias information that potentially will affect the analysis. When dealing with this situation, one can either exclude the incomplete column or rows containing a threshold *a priori* defined of missing data from the analysis or impute the missing values with a suitable value – mean or mode, for example (Sainani, 2015) (Table 2).

Table 2. Completeness Example in Address

	NAME	ADDRESS	NIF	DATE_BIRTH
✓	CLIENT_A	101 STREET	123000000	03/03/1996
✗	CLIENT_A		123000000	03/03/1996

CONSISTENCY. With data being reproduced multiple times and places, its format and shape have to be concise with previous entries, being non-variant, when comparing two or more values given a specific definition. It is taking into account both syntactic and semantics. On a semantic and syntactic view, respectively, a client’s subscription information has terminated. However, that information has been registered as “Open” or registered with “Clode” instead of “Closed”; as a result, the client will continuously receive campaign communications when was no longer necessary (Table 3).

Table 3. Consistency Example in Address

	NAME	ADDRESS	NIF	DATE_BIRTH
✓	CLIENT_A	101 STREET	123000000	03/03/1996
✗	CLIENT_A	101 STRIT	123000000	03/03/1996

TIMELINESS. Over time, the world changes; however, the stored data values remain the same if no maintenance or updates are done. Timeliness refers to the delay between a real-world state’s change and the subsequent change in the information system’s state. If a previous input data value is considered

accurate for the present time, therefore is called a *current* or *up-to-date*. When records are *out-of-date*, they are considered inaccurate data values (Fox et al., 1994).

Regarding the timeless dimension, there are two elements to take into account- *age* and *volatility*. Being *age*, one element of this dimension can indicate how old the data stored is and the amount of time passed since its collection. Volatility helps rate data stability, an attribute’s frequency of values shifting (Batini et al., 2009); for example, a field ADDRESS calls for update more often than Fiscal Identification Number attribute (Table 4).

Table 4. Timeliness Example by Updating Address

	NAME	ADDRESS	NIF	DATE_BIRTH
✓	CLIENT_A	NEW STREET	123000000	03/03/1996
✗	CLIENT_A	OLD STREET	123000000	03/03/1996

UNIQUENESS. Unique definition, in Merriam-Webster dictionary, is being without a like or equal, having distinctively characteristic. The exact definition can be applied to data. Each real-world object or event should only be represented once in a unique manner within the application. When there is an expectation of uniqueness, data occurrences should not be generated if there is an existing record for that entity. Uncovering this issue proposes identifying an appropriate primary key and performing a duplicate analysis to determine duplicate records - two entries with equal values for every feature (Table 5).

Table 5. Uniqueness Example by Having Duplicate Records

	NAME	ADDRESS	NIF	DATE_BIRTH
✓	CLIENT_A	101 STREET	123000000	03/03/1996
✗	CLIENT_A	101 STREET	123000000	03/03/1996

VALIDITY. Data validity applies to data obtained in compliance with the standards or meanings that relate to that data. Each column has diverse metadata attributes, such as data type, precision, format patterns, etc. Validity measures the degree to which data values are consistent with their metadata attributes (Jayawardene et al., 2013) (Table 6).

Table 6. Validity Example in Date_Birth

	NAME	ADDRESS	NIF	DATE_BIRTH
✓	CLIENT_A	101 STREET	123000000	03/03/1996
✗	CLIENT_A	101 STREET	123000000	March

The dimensions described above are helpful indicators to understand what needs to be corrected or improved in a given Dataset to enhance Data Quality. Once those measures are analyzed, it is possible to determine what data aspects need to be improved. For instance, one of the best practices is to have a helpful description and metadata which provide data context. It is also essential to standardize formats and rules within and across organizations, to improve overall data usage. Good metadata improves DQ by improving validity and consistency, creating a mechanism for assessing quality on the other dimensions through the data lifecycle.

Despite all prevention techniques, some errors will occur. Detecting and correcting these mistakes is a crucial component of DQ. Quality control is often done manually but can be streamlined through data profiling tools and cataloguing common data problems with simple corrections. Using summary statistics for data review can also help uncover potential errors that need correction. Ensuring that resources are dedicated to creating a pipeline that enables correcting common errors helps improve data accuracy, completeness, and consistency. Furthermore, for an organization to prevent duplicate records and thus improve uniqueness, a data pipeline must be clearly defined and carefully designed in data assets, data modelling, business rules, and architecture. Effective communication is also needed to promote and enforce data sharing within the organization, improve overall efficiency, and reduce potential data quality issues caused by data duplications.

Finally, to maintain and guarantee DQ in a company's data, it is fundamental to invest in a capable Data Control team. A Quality Assurance team would monitor the quality of software and programs whenever modifications occur. This team's rigorous change management is essential to ensure data quality in an organization that undergoes fast transformations and changes with data-intensive applications. Depending on the business, a Production Quality Control team would be a combination of the Quality Assurance or Business Analyst team. It must understand the business rules and requirements and be equipped with the tools and dashboards to detect abnormalities, outliers, broken trends, and any other unusual production environment scenarios. This primary team goal is to identify any data quality issues and fix them before they impact the final user and client operations.

2.3.3. Data Types

Data Quality approach aims to understand better the physical world values stored, retrieved, and mined. Not all data has the same shape, resulting in distinct storage and analysis methods; thus, defining the various structures data can take is a valuable asset for DQ analysis. The term data structure is used to describe a specific way of arranging information for specific operations types. Three main types are acclaimed among most authors:

Structured data. An organized and aggregated data by attributes within a domain. The most popular form of structured data is based on relational tables and statistical Data (Batini et al., 2009). Structured

data, often named quantitative data, regards data within fixed fields and columns, typically stored in a relational database management system (RDBMS) (Table 7). Its source consists of numbers and text, which are originated automatically or manually, requiring a predefined data model or schema. Moreover, this data format allows for searchable content through queries.

Table 7. RDBMS Table Example - Structure Data Type

	FULL_NAME	ADDRESS	FISCAL_NUMBER	BIRTHDAY_DATE
1	CLIENT_A	101 STREET	123000000	24/08/1970
2	CLIENT_B	102 STREET	987000000	04/08/1997
3	CLIENT_C	103 STREET	654000000	11/12/1971

Unstructured data. Unstructured Data is Data without a predefined structure that cannot be stored easily in a conventional column-row database. Contrary to structure sources, it is not actively managed in a transactional system, such as an RDBMS, and is often stored in data lakes, having no predefined data model or schema. Thus, it was harder to analyze and not promptly searchable, so it was not functional until in recent years for organizations. Today, however, we have artificial intelligence-driven unstructured data analytics tools explicitly created to access the insights available from unstructured data. Exploiting these complex and voluminous data sources brings immense value to businesses’ goods and services. The most common examples of unstructured data are free-text documents, such as customer reviews or support, resumes, e-mail body, social media, amongst others.

Semi-structured data. The in-between definition of the structure and unstructured data is neither raw Data nor input as a conventional database, being the kind of information that a degree of versatility. At any early processing level, the same piece of information may be regarded as unstructured, but after some research has been carried out, it will later become very organized (Abiteboul, 1997). An excellent example of this data type would be an XML file (Figure 3).

```

<configuration xml-version="0.6">
  <properties>
    <property name="configurationId" value="1083"/>
    <property name="feedName" value="TABLE_A"/>
    <property name="sourceSystem" value="table_source"/>
    <property name="feed_version" value="1.0"/>
    <property name="encoding" value="UTF-8"/>
    <property name="delimiter" value="|;/>
    <property name="headerRow" value="true"/>
    <property name="frequency" value="00:00:01:00:00"/>
    <property name="DSClassification"/>
    <property name="allowedPurposes"/>
    <property name="folderHDFS" value="/user/ptdmaap/TABLE_A/table_source/landing"/>
    <property name="riskScore"/>
    <property name="valueScore"/>
    <property name="GCPRetention"/>
    <property name="updateMode" value="I"/>
    <property name="DSPhase"/>
    <property name="DSTrustIndex"/>
    <property name="DSCustodian" value="DATACUSTODIAN@email.com"/>
    <property name="DSCustodianTeam" value="Business Intelligence"/>
    <property name="DSSteward"/>
    <property name="DSStewardTeam"/>
    <property name="DSFunctionalDataOwner"/>
    <property name="DSFunctionalDataOwnerTeam"/>
  </properties>
  <landing>
    <properties>
      <property name="bucket" value="cloud_bucket"/>
      <property name="folder" value="ptdmaap/TABLE_A/table_source/landing"/>
    </properties>
  </landing>
  <targets>
    <target type="Bucket">
      <properties>
        <property name="bucket" value="cloud_bucket"/>
        <property name="folder" value="ptdmaap/TABLE_A/table_source/landing"/>
      </properties>
    </target>
    <target type="BigQuery">
      <properties>
        <property name="dataset" value="table_source"/>
        <property name="table" value="TABLE_A"/>
      </properties>
    </target>
    <target type="BigQueryStaging">
      <properties>
        <property name="dataset" value="table_source"/>
        <property name="table" value="TABLE_A"/>
      </properties>
    </target>
  </targets>
  <schema>
    <fields>
      <field name="COLUMN_1" type="integer" required="true" personalInformation="false"/>
      <field name="COLUMN_2" type="integer" required="true" personalInformation="false"/>
      <field name="COLUMN_3" type="string" required="true" personalInformation="false"/>
      <field name="COLUMN_4" type="string" required="true" personalInformation="true"/>
      <field name="COLUMN_5" type="timestamp" required="true" personalInformation="false"/>
      <field name="COLUMN_6" type="string" required="false" personalInformation="true"/>
      <field name="COLUMN_7" type="string" required="false" personalInformation="true"/>
      <field name="COLUMN_8" type="integer" required="false" personalInformation="false"/>
      <field name="COLUMN_9" type="string" required="true" personalInformation="false"/>
    </fields>
    <fieldActions>
      <fieldAction fieldName="COLUMN_6" actionType="function" actionParameter="anonymise" actionMethod="bdpmanonptid"/>
      <fieldAction fieldName="COLUMN_4" actionType="function" actionParameter="anonymise" actionMethod="bdpmanonptid"/>
      <fieldAction fieldName="COLUMN_7" actionType="function" actionParameter="anonymise" actionMethod="bdpmanonptid"/>
    </fieldActions>
    <partitions>
      <partition name="year"/>
      <partition name="month"/>
      <partition name="day"/>
    </partitions>
  </schema>
</configuration>

```

Figure 3. Project example of an XML file - A semi-structured data

2.3.4. Metrics for Data Quality

Ensuring a good quality for the data stored in various ways is a complex issue that depends upon a systematic methodology. Thus, it is fundamental to outline a series of steps to measure and optimize information quality based on the Data Quality dimensions previously mentioned. This chapter will

elaborate the how can DQ be measured and put in numbers, based on Wei Dai, Issac Wardlaw, Yu Cui, Kashif Mehdi, Yanyan Li and Jun Long work (Dai et al., 2016).

The dimensions described earlier in this work are considered the most reliable indicators for quantum DQ. However, they are merely qualitative, not providing actual quantitative values for a more accurate assessment. (Dai et al., 2016) created a sense of quantitative metrics based on dimensions definition (Table 8).

Table 8. DQ Dimensions - Definition and Metrics

Dimension	Definition	Metric
Accuracy	The degree of agreement with an identified source of correct information.	The correct data percentage (correct data/total data).
Completeness	The level of data missing or unusable.	Percentage of all complete data.
Consistency	The level of conflicting information.	Percentage of all consistent data.
Timeliness	The degree to which data is current and available for use in the expected time frame.	The delay between a real-world state's change and the subsequent change.
Uniqueness	The level of nonduplicates.	Percentage of all unique data (e.g. primary keys, foreign keys)
Validity	The level of data matching a reference.	Percentage of all valid data (e.g. first name, last name and suffix)

Either using a data sample or the total data volume will lead to the same results.

Table 9. Illustrative Example of Client Information – Entry Date is a Control Field

ID (Int)	Name (String)	Address (String)	E-mail (String)	Date of Birth (DateTime)	Entry Date (DateTime)
1	Client A	101 STREET	Test_01@email.com	03/03/1996	01/01/2021
1	Client A	101 AVENUE	Test_01@email.com	03/03/1996	02/01/2021
2	Client B	102 STREET	Test_02@email.com	24/08/2002	01/01/2021
2	Client B	102 STREET	Test_02@email.com	24/08/2002	01/01/2021
3	Client C	104 STREET	Test_03	11/12/1699	01/01/2021
4	Client D	105 STREET	Test_04@email.com	April	01/01/2021
5	Client E				01/01/2021

Based on the Data Quality dimensions described in Table 8, the following metrics were calculated to summarize the data quality of the illustrative example of Table 9. It is possible to evaluate this dataset with an 80% accuracy score since almost every record is accurate, except for the missing data values, incorrect email information of the client with ID 3, and the client's name with ID 4. Furthermore, this dataset contains five out of seven unique rows, leading to a 71.4% uniqueness score. Moreover, this dataset's validity can be classified with a 96% score, where the only invalid record is the date of birth of the client with ID 4 since its content (April) does not fit with its column DateTime type. It is essential to notice that, in this illustrative example, the table schema allows for nullable values, that despite being inaccurate, they are considered valid. Regarding the timeliness dimension, the client with ID 1 had its Address updated, indicating one day change delay, resulting in 24 hours of outdated information that could impact possible analysis. Lastly, the client's misspelt name with ID 4 leads to a 96% consistency score, where the remaining ones are grammatically correct.

2.4. DATA PROFILE

To assess the Data Quality of a given data sample layout, data profile tasks help discover if the current data is coherent with the defined solution's quality measures. One of the underlying DQ processes aims to continually recognize anomalies, incoherences, redundancies, and incompleteness in data and respective metadata. It saves execution time due to recognizing the data sample's main issues, avoiding redundant processing from not acceptable data sources, especially when those sources do not have quality controls (Manjunath et al., 2010). Data profiling is the first step for every organization aiming to enhance the quality of information and provide educated decisions. It is a critical step for DWH and Business Intelligence (BI) projects, uncovering data source issues and the needed corrections in the Extract, Transform, and Load (ETL) pipeline. Furthermore, it also benefits data migration projects by identifying quality problems during the migration from source to target, unveiling new requirements that were not initially anticipated.

In summary, Data Profile assesses data validity and consistency. Moreover, it analyzes data in-depth, using computational algorithms to detect data set statistics such as means, limits, percentiles, and frequencies. It then uses that information to expose how those factors align with the business' standards and goals. DP can bring numerous benefits like improving DQ and its credibility by eliminating duplications or anomalies and influence predictive decision-making by mitigating minor mistakes before they become significant. It also assists in being proactive during crisis management situations, promptly identifying and solving challenges ahead, and tracing data to its primary source to guarantee suitable security encryption. Additionally, DP analyzes multiple databases, tables, and applications to confirm that the data complies with standard statistical measures and business rules.

2.4.1. Structure discovery or Column Profile

Column Profile validates if the present data is consistent and formatted correctly, reporting basic statistics, such as average, minimum, maximum, among others. Each source data column, or filled value, is analyzed, giving detailed information about data type and size, values spectrum, as well as their distribution, frequency, nullability, and uniqueness. Modern methods generate histograms and statistical information to determine the variable distribution, and, through regular expressions, it is possible to identify common patterns in the data values. Such results are represented in data profiling software, suggesting behaviour such as declaring a column with only specific values as a key-candidate or recommending that the most common trends be enforced (Abedjan et al., 2015).

Furthermore, these methods help to perceive how well data is structured - for example, what percentage of zip-codes do not have the correct number of digits. For this DP type, it is crucial to assess a column's properties, their statistic information. They will be described in more detail as follows.

2.4.1.1. Columns Properties Profile

An analyst must be able to detect irregularities in the columns of multiple tables of a Dataset by verifying a column's properties to seek its characteristics. It is considered a valid practice for property analysis to examine the existing documents and metadata so that the analyst can expect it and quickly locate an anomaly. The analysts may presume that each column's encountered properties are incorrect. They may need to be modified in the metadata or records to show each column's desired properties depending on both technical and business specialists' agreement (Rodrigues, n.d.). The main points that should be taken into consideration when verifying column properties are, for example, the domain, nullability, data type, and others (Table 10).

Table 10. Columns Properties to be Verified

Properties	Description
Column Name	Verify if the name is suitable according to the columns values.
Domain	Verify if the columns values match the domain of the column. E.g. for a binary column, only two values must be considered.
Data type	Verify if the data type matches the columns values.
Nullability	Verify if most field values are null in the column.
Uniqueness	Verify if the field only contains unique values throughout the table.
Length	Verify the length that appears the most throughout the table for the several values of the same column
Precision	Verify for the numeric fields if the number of digits appears on the left side of the comma.
Scale	Verify for the numeric fields if the number of digits appears on the right side of the comma.

2.4.1.2. Columns Statistic Profile

Throughout analyzing Dataset columns, statistical inferences can be deduced. The most straightforward conclusions result from counts or percentages, such as missing and unique values, duplicate rows, and both minimum and maximum value of each column. The other most common statistical information gathered when performing a DP analysis is descriptive statistics, which summarizes central tendencies measures, such as the mean, median or mode. There is variability or spread measures within the data points, such as the standard deviation, skewness, variance, etc. Moreover, quantile statistics information may also be encountered as a DP analysis result. Quantile statistics provide information regarding the quantiles and percentiles, which are the dividing values that split the Dataset into four or one-hundred equal portions.

2.4.2. Relationship discovery or Multi-column Profile

Multi-column profiling extends single-column profiling to several columns, revealing inter-value dependencies and column similarities. The multi-column analysis may be used to identify redundant data as well as determine data normalization opportunities. One role is to use frequent patterns or association rules to find correlations between values. The most commonly used statistical measure for assessing relationships between quantitative features is the correlation. It is possible to have a positive, neutral or negative correlation. Once a positive correlation is present within two features, it indicates that Feature_A and Feature_B moves towards the same direction; if one increases or decreases, the other shall do the same. The identical logic follows for the negative correlation. However, instead, they move in the opposing direction, i.e. if Feature_A increases Feature_B will decrease and vice-versa. There are several methods to calculate, with numerical data, the correlation between features, depending on the chosen coefficient. The correlation can be estimated with the correlation coefficients whose values range from [-1;1]. A correlation coefficient of 1 indicates that the two variables are entirely related in a positive linear manner. A correlation coefficient of -1 indicates that two variables are entirely related in a negative linear manner. In contrast, a correlation coefficient of zero indicates the non-existence linear relationship between the two variables being considered (Gogtay & Thatte, 2017).

The most commonly used correlation coefficient is Pearson's Coefficient (r) (1). It is used to measure the strength and direction of a linear relationship between two features and is based upon three assumptions: a linear relationship, independent features and normally distributed. Mathematically this can be determined by dividing the covariance of two features, X and Y, by the product of their standard deviations, S_x and S_y .

$$r = r_{x,y} = \frac{cov(x,y)}{S_x \times S_y} \quad (1)$$

Correlation is employed to denote the association or relationship between two (or more) quantitative variables. It is often considered that correlation and causation are synonymous. It is fair reasoning when one event triggers another. The two are usually linked and therefore correlated (e.g., effort and results, inspection and quality, investment and return). However, the correlation must not always be mistaken with causality. If two features are related, it does not imply that one may affect the other or the cause for its happening. One potential interpretation for correlation without causation is the presence of a third, unobserved factor that causes one variable to appear to trigger the other when, in reality, each is caused by the missing variable (Siegel, 2012). The spurious correlation concept outlines how one can obtain a significant value for a coefficient of correlation when the two variables, in reality, are uncorrelated. Spurious correlations directly result from having a high correlation due to some third factor for two independent variables.

2.5. DATA ANONYMIZATION

Information diversity flows in every corporate DWH, from a description or calling date entry to social security number, cellphone number, or client address. In today's data-driven business practice, it is trusted that companies keep Personal Identifiable Information safely in their systems. Any data that could potentially identify a given person has to be appropriately handled. A PII is considered any piece of information that can identify or intel a given individual, whether alone or combined with other additional information. When fallen into the wrong hands, it can be catastrophic. It can take form in multiple areas like Finances (credit card number and credit balance), Medical Information (medical history and identification number), Contact Information (e-mail address and telephone number), and many (Raghunathan, 2013). It is vital to classify this kind of information to disguise and make it impossible to code unless accurately. Balaji Raghunathan described Data Anonymization (DA) to classify this delicate information and transform it while still conserving its original format and data type. Proper implementation and use of data anonymization decreases the likelihood of misuse of personal data and improves compliance with data privacy laws. Companies that make the proper use of data anonymization processes significantly reduce penalization risk. For example, Netflix published 100 million documents showing how its customers rated movies from December 1999 to December 2005. The information was first anonymized, omitting personal details such as usernames. A few weeks later, two University of Texas researchers confirmed that some of the Netflix users in the data collection had been found by deanonymizing the data (Matsunaga et al., 2017). When Data Security is one of the main concerns for organizations for storing their data in the Cloud, it is vital to have a reliable data anonymization strategy. DWH migration to the Cloud anonymize data before deploying to the Cloud is a must, also while processing this information through an ETL pipeline. How can a PII be identified, and secondly, how can it later be anonymized?

Personal Identifiable Information can be identified in a more archaic way or a more modern approach. A more traditional way is to have a knowledgeable team stored in DWHs company, set and trained to identify every PII by hand, which is still a used method. Nevertheless, it can be subjective to human error due to high data volume, turning it into a non-bullet-proof approach. A more efficient way of increasing the PII identification while minimizing mistakes is to use an ML algorithm capable of classifying PII, such as Named Entity Recognition (NER). Automatic processing of natural language text is analyzed to detect and classify the designated entities present. This analysis’s input is plain text, and as output is expected to return, the PII in that text is identified accordingly (De & Do, 2020).

It exists various methods for anonymizing what is considered a PII to protect private and sensitive information. The following techniques are the most utilized (Matsunaga et al., 2017):

RANDOMIZATION. This strategy consists of replacing the actual data values for random values of eliminating the strong correlation between the data and the users (Table 11). There are several ways for randomization to be applied. Any of these include a Noise Addition - applying random noise to the initial data; Permutation - rearranging attribute values in a table so that some of them are arbitrarily connected to various data topics; Differential privacy - adds sufficient noise to the query result.

Table 11. Randomization Technique Example in Full_Name

	FULL_NAME	ADDRESS	FISCAL_NUMBER	DATE_BIRTH
Original	CLIENT_A	101 STREET	123000000	24/08/1970
Anonymized	LCNEIAT	101 STREET	123000000	24/08/1970

PSEUDOANONYMIZATION. By replacing one attribute (typically a unique feature) in a record with a new feature. A particular one. The primary methods for implementing this strategy are encryption function with a hidden key, Hash functions and tokenization. Pseudonymization is a de-identification method that uses cryptographically generated tokens to replace sensitive data values. Commonly used in finance and healthcare to help reduce data in use risk, narrow compliance scope and mitigate sensitive data exposure systems while maintaining data usefulness and accuracy. These techniques enable either *one-way* or *two-way* tokens. A one-way token has been transformed irreversibly, while a two-way token can be reversed. Because the token is created using symmetric encryption, the same cryptographic key that can generate new tokens can also reverse tokens (Table 12).

Table 12. Pseudoanonymization Technique Example using Hash function SHA256 in Fiscal Number

	FULL_NAME	ADDRESS	FISCAL_NUMBER	DATE_BIRTH
Original	CLIENT_A	101 STREET	123000000	24/08/1970
Anonymized	CLIENT_A	101 STREET	a29339197e931c2f0079f7718cce73afb3442497 016c8a18166646b7d49103b9	24/08/1970

SUPPRESSION. The primary identifier or quasi-identifiers is removed to generate the anonymization table. It is used in computational databases, which have only summaries of the table data instead of individual data (Table 13).

Table 13. Suppression Technique Example in Address and Fiscal Number

	FULL_NAME	ADDRESS	FISCAL_NUMBER	DATE_BIRTH
Original	CLIENT_A	101 STREET	123000000	24/08/1970
Anonymized	CLIENT_A	*	*	24/08/1970

GENERALIZATION. This approach substitutes quasi-identifier values for less precise but semantically compatible values. Another more generic value substitutes a value in this form, which is faithful to the original. For instance, the date of birth may be extended to a range such as year of birth to minimize the probability of recognition (Table 14).

Table 14. Generalization Technique Example in Date Birth

	FULL_NAME	ADDRESS	FISCAL_NUMBER	DATE_BIRTH
Original	CLIENT_A	101 STREET	123000000	24/08/1970
Anonymized	CLIENT_A	101 STREET	123000000	1965 < Year < 1975

3. TOOLS AND TECHNOLOGY

Due to the high dimensionality and complexity of the project described in this work, a wide range of tools was needed to fulfil its needs. This chapter will outline the tools used both on-premises and on the GCP and used on the DP.

GOOGLE CLOUD PLATFORM. GCP is a series of computational tools from Google, made available as a public cloud product to the general public through services. The GCP tools consist of physical hardware equipment located within Google's globally dispersed data centres, including servers, hard disk drives, solid-state drives, and networking. All of the modules are custom built using patterns close to those available in the Open Compute Project. As an option to consumers constructing and managing their physical networks, this hardware is made available to users through virtualized resources, such as VMs. GCP offers over 50 services in Compute, Storage & Databases, Networking, Big Data, Machine Learning, Identity & Security, and Management & Developer tools. These platforms may be used individually or in conjunction to create their personalized cloud-based infrastructure for developers and IT specialists.

TALEND OPEN STUDIO FOR DATA QUALITY. Talend is an open-source data-oriented integration framework. It offers numerous data integration tools and facilities, data management, business systems integration, data consistency, cloud computing, and Big Data. Talend released its first product, Talend Open Studio, now known as Talend Open Studio for Data Integration, in October 2006. Also, an open-source platform that supports ETL-oriented implementations and is typically supported for deployment on-site. The integration of operating structures, ETL processes, and data transformation is commonly used. Talend Open Studio for Data Integration is structured so that data present at various locations across a company can be conveniently integrated, translated, and modified. Talend Open Studio for Data Quality specific modules was used for profiling analysis – Talend Data Quality profiling tools. These modules (Figure 4) are the following⁴:

- **Structural Analysis:** Through reviewing the database material, this review helps to create a summary of the layout of the DB, index, or schema. It also analyzes essential structural elements such as the number of tables in the DB, the number of rows per table, the number of primary keys, and the number of indices. It will show which tables are complete, empty and how the structure is constructed. It can be used for the complete DB or an illustrative sample of tables.
- **Cross Table Analysis:** To find the relationship between columns or to discover the relationship between international keys. There is just one analysis routine in the CrossTable Analysis,

⁴ <https://help.talend.com/r/8taYQkblNoWRWJGmRtNq3g/0DqVjpaBsgkhmJrp7ebEug>

namely Redundancy Analysis. It is used to match similar columns in separate tables or align foreign keys in one table to the other table's primary keys, and vice versa.

- **Tables Analysis:** This is mainly used for the analysis of data in single columns. However, data may be evaluated in various tables, including developing corporate rules containing clauses. Note that will establish a "dependencies analysis" that will evaluate any variations and dependencies between the two columns. Such findings will reveal whether the variables are determined by the relationship between the tables or based on them.
- **Column Analysis:** Allows doing the similarities analysis and frequencies in data columns, existing different types of analysis routines to benefit. It helps analyze columns in databases and delimited files as well. This analysis can be performed on several columns, but each column is evaluated independently, with no relation to any other column.
- **Correlation Analysis:** This functionality will compare different columns with each other, helping to detect a correlation between them. Exists a variety of chart forms available to visualize the data in and of the analysis routines. Numerical and time-based association analysis is taking into account into the analyses. However, there are two points to consider when applying this routine. First, Correlation Analyses should not be used on files; thus, they can only be applied to DB columns. Secondly, it is used to search data type correlations, hence not providing statistical information about DQ. One helpful variation of this component is the Nominal Correlation Analysis used to examine, in the same Table, the minor correlations between nominal columns. To visualize the correlation of the nominal values, it is primarily used a line chart, where the thicker the line, the stronger the correlation is.

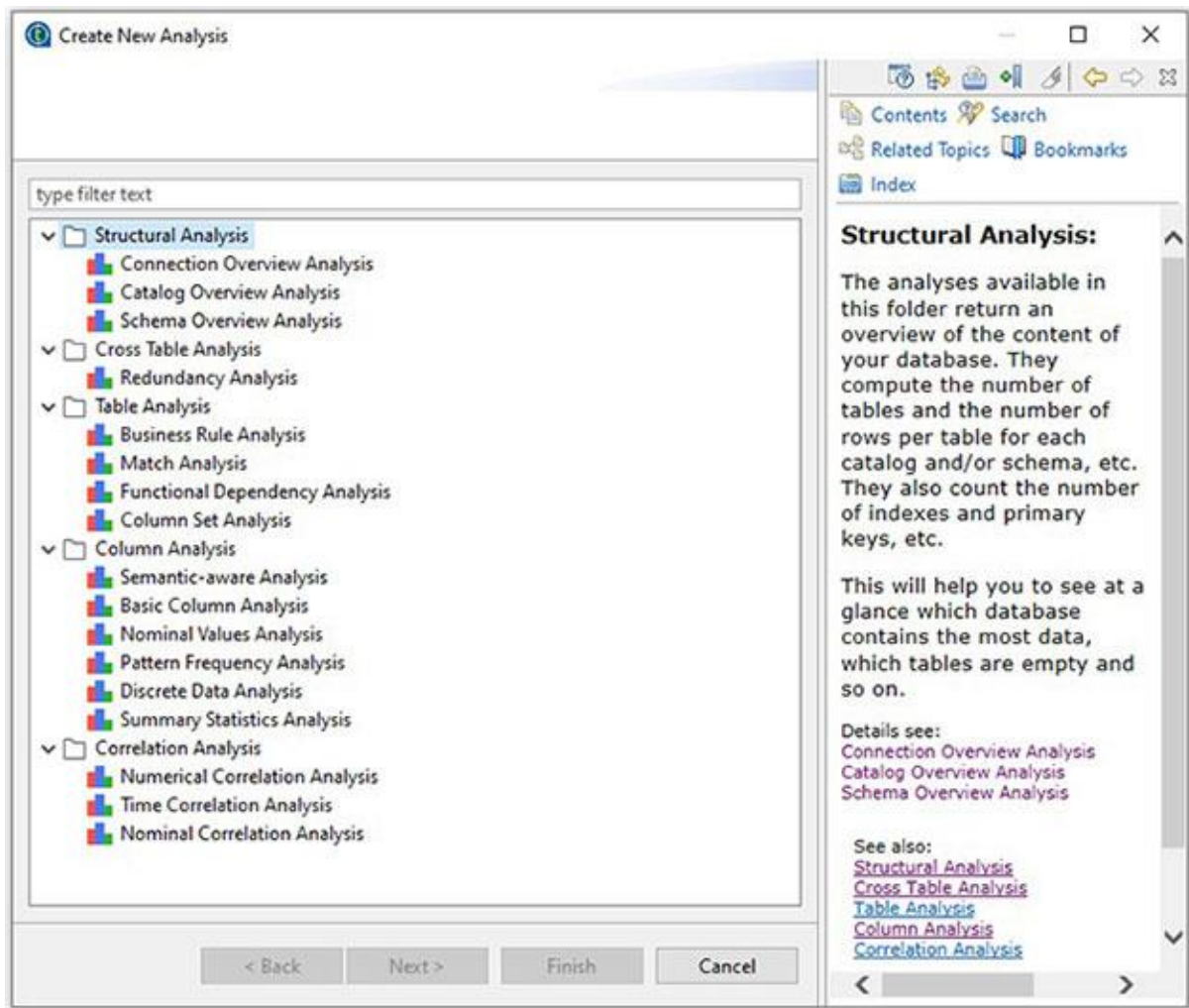


Figure 4. Talend Profiling Modules

ORACLE DATABASE & ORACLE ENTERPRISE DATA QUALITY. Oracle database (Oracle DB) is a relational database management system (RDBMS) from the Oracle Corporation. Oracle DB was created by Lawrence Ellison and other developers in 1977 and is one of the most trusted and commonly used relational database engines. The architecture is designed on a relational database paradigm in which users can directly access data structures using the structured query language. This tool was mainly used for querying and developing PL/SQL procedures – a language procedure built explicitly within its syntax to embrace SQL statements. Oracle Enterprise DQ (EDQ)⁵ delivers a robust management environment for DQ to understand, optimize, secure, and monitor data quality. EDQ enables master data processing, data integration, business analytics, and data migration initiatives for best practice. In customer experience management and other technologies, EDQ accommodates advanced data consistency. It allows address verification, data profiling on files, DB, and spreadsheets;

⁵ <https://www.oracle.com/assets/oracle-enterprise-data-quality-ds-430148.pdf>

standardization recognizes incorrect values, missing data, inconsistencies, duplicate records, and critical quality metrics; matching and merging columns by checking duplicates. Moreover, it can utilize prebuilt templates or user-defined rules to profile data. EQD also connects to other Oracle data governance products, including Oracle Data Integrator and Oracle Master Data Management (Dai et al., 2016).

PANDAS PROFILING. Pandas Profiling (PP) is a Python open-source module that efficiently conducts exploratory data analysis with only a few code lines. Besides, it also produces web-format interactive reports that can be provided to any user, even if not familiar with programming. It generates a report with the visualizations and understanding of each variable regarding their distribution, also adding warnings for variables containing null values, categorical features with high cardinality, and duplicate records. For each feature, the following information is present⁶:

- **Type inference:** detect the types of columns in a data frame.
- **Essentials:** data type, unique values, missing values
- **Quantile statistics:** minimum value, Q1, median, Q3, maximum, range, interquartile range
- **Descriptive statistics:** mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness
- **Most frequent values**
- **Histogram**
- **Correlations:** focusing on the highly correlated variables, Spearman, Pearson, and Kendall matrices
- **Missing values:** matrix, count, heatmap, and dendrogram of missing values
- **Text analysis:** learns about categories (Uppercase, Space), scripts (Latin, Cyrillic), and blocks (ASCII) of text data.
- **File and Image analysis:** extract file sizes, creation dates, and dimensions and scan for truncated images.

All information is handed over in a clean, readily available design, multiplying ways of outputting the results in HTML, Excel, or Jupyter Notebooks format. Although, this Python library main disadvantage are the positive correlation between the dataset size and the time it takes to generate the report, for if one increases, the other does too; moreover, if the problem requires additional visuals and statistical information, adding these functionalities to PP is challenging, since it depends on modifying the package source code.

⁶ <https://github.com/pandas-profiling/pandas-profiling>

MULTIPROCESSING. A python module that enables parallel processing with an API to divide jobs between multiple processes. Multiprocessing refers to the capacity of a device to operate in more than one processor simultaneously. In a multiprocessing method, programs are broken down into more straightforward routines that operate independently. The operating system assigns these threads to the system's performance-enhancing processors. Consider a single-processor operating system. Suppose multiple processes are allocated at the same time. In that case, each task will have to be stopped and temporarily transferred to another. To keep all the processes running, with the parallel processing, a Central Processing Unit (CPU) can efficiently perform many tasks at once, with its processor used for each operation.

DASK. Dask in Python is a versatile library for parallel computation. It has computer-optimized dynamic task scheduling and has collections for many larger datasets, such as DataFrames, Bags, and Arrays that extend standard interfaces such as Pandas, NumPy Python iterators to more expansive or distributed environments than memory. These parallel collections operate on top of complex task schedulers. Dask provides the following benefits⁷:

- **Familiar:** Provides parallelized NumPy array and Pandas DataFrame objects.
- **Flexible:** Provides a task scheduling interface for more custom workloads and integration with other projects.
- **Native:** Enables distributed computing in pure Python with access to the PyData stack.
- **Fast:** Operates with low overhead, low latency, and minimal serialization necessary for fast numerical algorithms.
- **Scales up:** Runs resiliently on clusters with 1000s of cores.
- **Scales down:** Trivial to set up and run on a laptop in a single process.
- **Responsive:** Designed with interactive computing in mind, it provides rapid feedback and diagnostics to aid humans.

Dask has three main collections, Dataframes, Bags, and Arrays. For this work, the focus collection will be Dataframe. When Pandas are commonly used, Dask DataFrame is usually used, usually when Pandas fails due to data size or computation speed. A Dask DataFrame is a broad parallel data frame, divided along with the index, consisting of several smaller Pandas DataFrames. For larger-than-memory computing on one single computer or several separate machines in a cluster, these Pandas DataFrames can live on disk. On the respective Pandas DataFrames, one Dask DataFrame process causes several operations.

⁷ <https://docs.dask.org/en/latest/>

4. DATA PROFILE: THE PROJECT

Given the vast volume of data available today, corporations are often frustrated by all the data they have gathered. As a result, they fail to give full advantage of their knowledge to reduce its importance and usefulness. For maximizing its full potential and provide powerful insights, data profiling organizes and handles big data.

Over the past 20 years, data in the DWH has been centralized, generating a high amount of data from various business areas of Company B, as all data is grouped into columns, also known as feeds, or files (positional or delimited) of information from many business areas and clients. Feeds are categorized into nine releases to boost efficiency and organizational manageability, in which release is divided into Migration and BAU feeds. Migration feeds relate to historical records, so migration feeds are only processed once for each release to migrate to the GCP. BAU streams are daily loading procedures corresponding to all relatable data from previous modifications (inserts or upgrades) on operating systems, often requiring configuration to ensure daily database data updates in the DWH to be processed via the GCP on several occasions.

The fundamental purpose of WS3, where this study is currently based, is to transfer Migration and BAU feeds from DWH to GCP from different DBs. It is split into three primary teams Extraction, Dmaap, and Google Storage teams, converting all the feeds into the modern EIM.

The prime focus was to catalogue and extract from several DB's data according to a given release when initiating this project. Later, this data would pass through an ETL process to be processed into the Cloud. When this extract and processed data reach the Google Storage teams, it must be anonymized and normalized in agreement with cataloguing earlier defined. One of the many criteria that must be fulfilled is the correct specification of each feed to execute these processes smoothly. They must be both adequately anonymized and normalized. The recurrent problem is that countless fields are not being identified as they should, so there are times when a Telephone Number or a Fiscal Number are not noticeable leading to wrong anonymization and normalization, violating the DQ of this project. With the urge to resolve this issue and aid the analysis for the considerable amount of migrating information, Company_B proposed creating a pipeline able to understand better the data stored in the DWH. In light of this request, a DP sub-project was put in motion with the intent to dismiss or support data assumptions. It had to present statistical information and the probability of whether a column would contain a Telephone Number, a Date, a Fiscal Number, or other considered PII. Additionally, the achieved results had to be presented efficiently and user-friendly so that a broader audience beyond data experts can understand it.

4.1. THE PLANNING

By proposing this sub-project to help unveil data patterns and mysteries, the first step would be planning. Be that as it may, the migration process was already in motion when this was requested, so the soon to be implemented solution had to be versatile for both feeds previously migrated and future ones. Tables are partitioned into nine releases. Release_6 and Release_7 were being processed by the time planning started, with five Releases already deployed and two more to come. The DP pipeline must be capable of analyzing both information on the Cloud and the DWH. With that in mind, the following steps would be to find the proper tool for this application according to the problems needing solving and create an implementation flow. Initially, the team had about six weeks to complete a Proof of Concept (POC) - a possible application capable of fulfilling the established requirements with room for improvement - before it was presented to Company_A and Company_B in its totality.

4.1.1. Tool Assessment

DP does not need to be performed manually. Automating it with a tool is the most effective way to handle the profiling task. By removing errors and adding accuracy to the data profiling process, data profiling tools improve data credibility. After research, three possible options were selected to implement the DP solution, Talend Studios Pros, PL/SQL Pros, and Python's library PP. The EDQ solution was not later considered for the decision-making due to a license requirement with a high cost; for that reason, both Company_A and Company_B disregarded it. The pros and cons were evaluated for each tool, as it can be shown in Table 15.

Once the principal usage advantages were unambiguous, each platform solution was evaluated considering the fundamental management requirements. Only a feed sample was contemplated for this process, and so all tools asserted were based upon that assumption. As presented (Table 16), a PL/SQL procedure fills most of the rules defined; however, due to a significant implementation effort, it was decided not to pursue the solution given that it does not provide a graphical solution. With two tools remaining, given the great importance of automation in this solution to save time and resources, the Talend Open Studio for Data Quality was discarded once it did not have that ability, presenting more as a data cleaning solution rather than data analysis. For every new analysis, different settings had to input manually to achieve the goal report. Thus, the chosen, most suitable tool for this DP sub-project was Python library PP.

Table 15. Tools Assessment Pros and Cons

Tools	Pros	Cons
Talend Studio	<p>Integration with the source system.</p> <p>Patterns validation, such as e-mail and Telephone Number.</p> <p>Graphical output report for a more straightforward analysis of the data profiling results.</p>	<p>Requires a manual configuration for each analysis report.</p> <p>Closed solution.</p> <p>Limited export options (PDF, XML, XLS and HTML).</p>
Pandas Profile	<p>Processes automation.</p> <p>An open-source solution, allowing the possibility for some improvements (workarounds).</p> <p>Graphical output report for a more straightforward analysis of the DP results.</p> <p>High process performance.</p>	<p>The graphical solution is static, meaning it is not possible to configure new ones.</p> <p>The library version needs to be locked.</p> <p>Integration with Data Sources (DB/Files) requires an additional effort.</p>
PL/SQL	<p>Allows a more customized solution.</p> <p>Processes automation.</p> <p>Patterns validation, such as e-mail and Telephone Number.</p>	<p>Requires a significant effort for solution implementation.</p> <p>No graphical option.</p> <p>Process performance decrease when compared to other solutions.</p>

Table 16. Tools Functional Requirements Estimation

Functional Requirements				
Description	List of requirements	Talend Studio	Pandas Profiling	PL/SQL
Data Profiling – Behavior The solution should provide the option to generate samples and run them automatically.	The solution should run automatically.	✗	✓	✓
	The solution should be able to generate one random sample of 1000 lines per table.	✓	✓	✓
	The solution should count the total number of rows present on the sample file/input table.	✓	✓	✓
Data Profiling – Quantity The solution should calculate the data quantity for all selected data sources, displaying a list or graphic with the result.	The solution should be able to retrieve the table/file total size.	–	✓	✓
	The solution should count the number of duplicated records present on the sample file/input table based on the selected fields.	✓	✓	✓
	The solution should count the number of unique records present on the sample file/input table based on the selected fields.	✓	✓	✓
Data Profiling – Quality The solution should evaluate the data quality for all selected fields, displaying a list or graphic with the result.	Based on sample/available data, the solution should retrieve the maximum length for this field.	✓	✓	✓
	Based on sample/available data, the solution should retrieve the minimum length for this field.	✓	✓	✓
	Based on sample/available data, the solution should count the number of NULL/BLANK values present in this field.	✓	✓	✓
	Based on sample/available data, the solution should identify and count the number of records that match a specified pattern expression.	✓	–	✓
	Based on sample/available data, the solution should calculate a frequency distribution for all distinct values present in the selected field.	✓	✓	✓
Data Profiling – Statistics The solution should calculate statistics for all selected fields, displaying a list or graphic with the result.	The solution should count the number of records present on the sample file/input table based on the selected fields.	✓	✓	✓
	The solution should calculate the average amount for the selected fields present on the sample file/input table.	✓	✓	✓
	The solution should have the option to extract the result into a file (excel or word).	✗	✓	✓
Data Profiling – Extraction The solution should allow the option to export all the results.	The solution should have the option to extract the result into an HTML page.	✓	✓	✓



Fulfil requirement



Partial fulfils the requirement



Does not fulfil the requirement

4.1.2. The strategy

After selecting the more suitable tool for management's requirements layout, defining a strategy and the solutions architecture is the next step. The goal was to create a method that could provide statistical information for each dataset and feature, a brief data sample for each column, and pattern recognition - understand if a given column may contain a Telephone Number or Fiscal Number. However, before assembling those requirements into a pipeline, the architecture had to be designed first. The main intention was to design a solution as automated as possible with only an input file and two output files. With that purpose in mind, the solution must extract the feeds from both DBs and those deployed to GCP. Once the extraction is complete and data is ready to be analyzed by a Python script, it will process pattern classification through regular expressions – regex validation. As an output, it resulted in two reports, an excel file with Regex conclusions and an HTML report with the statistical analysis made by PP. It is essential to underline both tool decision, and the plan definition was considered just a feed sample, not complete extraction. This pipeline was implemented using both a Shell and Python script.

As input, to run the DP pipeline for all options laid out was created an input configuration file where could directly extract data from an Oracle DB, pull deployed feed from GCP, or analyzing a delimited file from a source folder. Nevertheless, each input file refers to only one configured feed. Along with the options described above, only one could be selected. Once the input file was structured, a different course of action would be set, agreeing with the chosen option. Regardless of how the feed was configured and extracted, all options converge in a final preparation implemented in Shell script before Python script processes them. This data preparation consists of decrypting, decompressing from a zipped format, and modifying the delimiter from “|;|” to “;” due to performance issues described in the Implementation chapter in this work. Afterwards, the feed goes through a Python script, and the feed data is loaded into a DataFrame format. Each row's column is compared with a Regex pattern, such as Dates, E-mail Address or Telephone Numbers, previously specified in a text file form. It estimates the probability of a given column categorized as a Date, E-mail Address, or any other pattern defined; the output is saved in an Excel file. The Regex patterns are listed in a separate text file to be modified as necessities change. Ultimately, the PP library is used for the feed data sample, producing a second output for this pipeline, an HTML report (Figure 5).

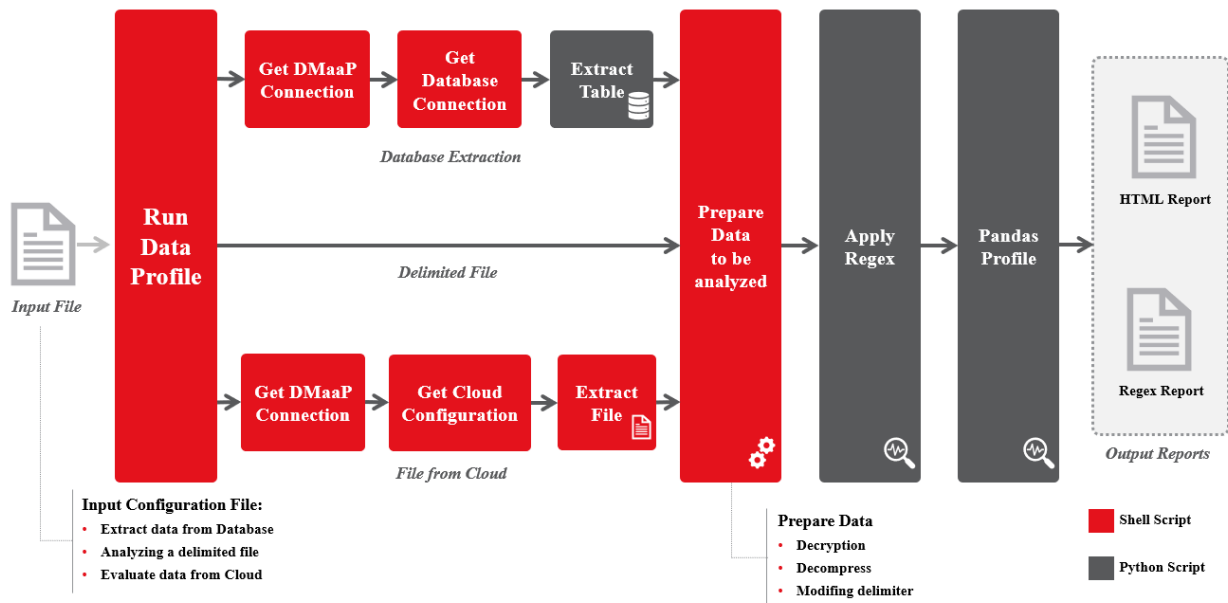


Figure 5. Functional Architecture

4.2. THE IMPLEMENTATION

The goal was set to implement a solution capable of profiling information present on DWH and Cloud to provide statistical insights and assess DQ. With all requirements prompt and ready, all conditions to start implementing the described strategy with the chosen tool were in place. The entire solution is set to run in a VM setting with both Shell and Python 3.7 version. Since Python libraries will be used for this work, a Virtual Environment (VE) with the specified package versioning was conceived to not disrupt the procedure with library updates further in the future.

The starting point would be the input configuration file creation, which had to be parameterized before executing a validation. A JSON file format was specified if it would be an extraction from a DB, GCP, or a delimited file in a folder. The implementation will be divided into two main parts according to the environments used – Shell and Python, for explanatory purposes.

4.2.1. Shell

When initializing the Shell script, immediately change the desired directory, begins the log file, and fetch the input configuration parameters into memory variables. **Extraction_Type** is one of those variables that give the feed source, whether from GCP, DB or a delimited file. Each extraction type has its method and function associated, and thus, for explanatory purposes, each method will be described as follows:

GOOGLE CLOUD PLATFORM EXTRACTION. When passing GCP as the feed source, it can either extract one single feed or multiple feeds from a given context, depending on the input files given parameters. If only one feed is extracted, then the variable **Id_Load** has the feeds identification number

that enables its tracking on the GCP. On the other hand, if the intention is to extract various feeds within a specific situation, then it is the variable **Context_Name** that contains the necessary information. For example, if the purpose is to analyze all feeds from HDMs Release_5, instead of passing N Id_Loads, only the Context_Name would be enough.

Subsequently, relying on the chosen parameter, Id_Load or Context_Name, one of two separate queries is executed in a specific DB designed for this project logs and parametrizations – Dmaap DB. The script connects to the Dmaap DB through a credential file already configured, executes the query to return the GCP feed path. It connects directly to the GCP with the resulting feed path and extracts to a newly created VM directory. Once the file is loaded into the new directory, it has to be decrypted, decompressed, and the delimiter has to be altered. The feed file decryption already existed a developed JSON script made for other intents, previously to the DP sub-project started. Decryption set only occurred if GCP extraction was in place. The following two methods could be applied for the DB and delimited file extraction.

With the file already decrypted, it had to be decompressed if being in a zipped format. According to the file extension, e.g .zip, .gz and .bz2, a specific shell function command line has been employed for decompressing. Ultimately, the file delimiters need to be altered to a one-byte character due to performance limitations later experienced when uploading the CSV file to a Dataframe on the Python module. The original delimiter was “;|” and the final intended delimiter was “;”. However, the data file contained “;” characters that were not separators. Thus, replacing directly every “;|” for “;” could compromise the original file structure. Therefore, each occurrence of “;|” was replaced with a unique expression “<<Dmaap>>”, and the “;” was then modified for a “;”. Once there were no “;” left, the unique expression “<<Dmaap>>” was substituted for “;” as planned.

DATABASE EXTRACTION. Considering a DB extraction implicates establishing a DB connection. It was only possible to connect to Oracle DB in this early stage, although it aims to develop for more DB; hence the input file and script were prepared for such, even though it is only functional for Oracle DB.

It begins by executing a query on Dmaap DB, a specific DB designed for this project logs and parametrizations, to obtain the following information: Host, Port, Service Name, Schema, User, Password and System. With the previous variables populated, the source DB connection string (1) is formed, alongside the extraction query (2) for a complete feed data or with constraints (3) where the table_name and filter_condition variables were evident in the input configuration file:

(1) *DBSourceConnectionString =DBSourceHost:DBSourcePort/DBSourceServiceName*

(2) *query="SELECT * FROM DBSourceSchema.\$tableName"*

(3) *query="SELECT * FROM DBSourceSchema.\$tableName WHERE filterCondition"*

Through **cx_Oracle**, a Python extension module that enables Python access to Oracle Database, the connection is established by opening a cursor to execute a query using the following command line (4):

```
(4) cx_Oracle.connect(User,Password,DBSourceConnectionString).cursor().execute(query)
```

The query result is saved in the created file directory, as a CSV file with ";" as the separator. Lastly, the file extraction is verified, and if so, no decompression will be necessary for this situation, only delimiter adjustment.

DELIMITED FILE EXTRACTION. The File Path and File Delimiter are explicit as parameters in the input configuration file in the eventuality that the feed is already in a delimited file format. As such, the file is first validated for its existence and copied from its original File Path to the created working directory, decompressed if needed, and has its delimiter modified.

When the feed or set of feeds are obtained, regardless of the extraction method chosen, this Shell script's last step is executing the Python script described in the next chapter. Throughout the entire process, all steps are being documented in the log file.

4.2.2. Python

As the Shell script creates the input folder with the feed dumps in a CSV format, a Python script is called in the Shell one. When entering the Python domain, an output folder is immediately created on the path passed by Shell script as a parameter. The feed dumps are then converted to a DataFrame through Python's library – Pandas. A function was created for a more robust approach to receiving either a CSV or an XLSX format. The first Dataframe was now generated – **df_sample**.

It was intended for each row's column to match with a Regex pattern and calculate the probability of a given column being labelled as a Date, E-mail Address, or any other expression defined. With that in mind, a function was created to convert the Regex text file into a dictionary since it only has a Key – Regex Category - and an associated Value – Regex Pattern - (Figure 6). This function has two-loop cycles embedded to iterate each column and each Regex Pattern, transforming the columns to a string type and trimming - removing the strings. However, if the Regex Pattern being analyzed is a Fiscal Number, extra measures had to be taken into consideration, for a Fiscal Number has the same pattern as a Telephone Number leading to false positives. After some research, a verification function was implemented to guarantee a Fiscal Number. Finally, it counts the rows that match each pattern and produces a dataframe with the column's name, Regex Category, Regex Pattern and the count as an output. The result Dataframe is filtered for counts higher than one, so it can be divided by the original feeds total number of rows, giving the percentage of matches results for each column's Regex Pattern. Otherwise, zero values would be divided, which would result in both mathematical and computational

errors. This course of action is embedded in a function called **count_regex**. Likewise, the final Dataframe will be named **df_results**.

```
{'email': '^([A-Za-z0-9_\.]*@[A-Za-z0-9]*\.[a-z]{1,5})$',  
  'cellphone': '^2[1-9]{1,2}[0-9]{7}$',  
  'fiscal_number': '^([12589][0-9]{8})$'}
```

Figure 6. Dictionary format of Regex Patterns Examples

The percentage needed to be significant to classify a particular column as a Telephone Number or a Date. It was hence added two more columns to **df_results** with the returned information of **get_classification** and **get_observation** functions. Both have the same inputs, Regex Category, its frequency, and a threshold defined by the user. **Get_classification** returns the label for a column if the Regex Category's frequency is greater than the established threshold, and if the frequency is not higher than the threshold, it classifies it as Unknown. On the other hand, **get_observation** will return the percentages of the most frequent(s) Regex Category for a column. Conversely, if the column is classified, the Regex Category is greater than the threshold, then no observation will be provided. Other columns were added to **df_result** for information tracking purposes, such as the feed name being analyzed. Just after, PP library is called using the following command line (5):

```
(5) profile = ProfileReport(feed_sample, title="Pandas Profiling Report", html={ 'style':{  
    'full_width':True } }, minimal=True)
```

Lastly, every information computed in **df_result** is recorded in an Excel file on the output folder created alongside the HTML PP report.

Aside from all the work done until this point, new criteria had to be taken into account, requested by one of the Company_B project owners. The whole feed has to be analyzed instead of the feed sample first contemplated and implemented. The strategy planned and already implemented was prepared for receiving a feed sample with approximately 1GB or 2GB of memory. However, now it would receive feeds with around 35 GB of memory or more. Too much data for a Python library to handle, leading to a significant computing time increase (Table 17).

Table 17. Computing time of a 30 GB Feed

Process	Start Time	End Time	Duration
GCP Extraction	14:11:53	14:53:30	00:41:37
Regex Validation	15:21:15	08:39:36	17:18:21
Pandas Profile	08:39:36	10:15:26	01:35:50

Conditions	Feed ID: 1
	Feed Name: Feed_A
	File Size: 30 GB
	Total Number of Rows: 79.066.523
	Number of columns: 50
	Number of Regex Patterns: 5

Ideally, the best and most correct approach would be a new solution, with a more appropriate tool more fitting for extensive data analysis, such as PySpark. Apache Spark is an open-source, general-purpose distributed computing engine that uses a vast volume of data to store and analyze. To promote the partnership between Apache Spark and Python, PySpark was released, and it is a Python API for Spark. Besides, PySpark enables working with Apache Spark and Python programming languages with Resilient Distributed Datasets (RDDs). The Spark data frame is the leading data form used in PySpark. This object can be considered as a cluster-wide table and has features identical to data frames in Pandas. Still, restart the solution with everything on track and time would demand a schedule and planning restructure plus other license expenses. Consequently, Company_A and Company_B obliged other approaches within the previous tools selected. After research, two alternative options were considered, use Dask library instead of Pandas or adding another package to the process called Multiprocessing.

Although Dask is a much better and consistent solution due to its scalability after some test runs, it did not correctly integrate with PP. It was discarded as a possibility since it meant to go back to the start implying a significant effort in redesigning the solution and impact the deadline defined, leaving the Multiprocessing approach to be implemented. Some code alterations had to be made; as of now, all feeds will be divided per GB and cores, each CPU is going to process 1GB of feed. Firstly, the command line used to load the feed into a dataset was added a new parameter **low_memory** to processing the file in blocks lowering memory usage while parsing, yet the whole file is saved into a single Dataframe.

After the function **multi_proc** was created to set the parallelism process in motion, it received a Dataframe and a given function to multiprocessing as parameters – df_sample and count_regex. Firstly, it summed each column’s memory usage in bytes to obtain the total memory and divided it into 1GB parts.

Those same parts will be the number of CPUs used to process the passed Dataframe. Later, using the multiprocessing pool, a process-based parallelism providing a more straightforward way to parallelize a function execution through several input values and thus distributing the input data across processes - data-based parallelism. With this module, the pools were formed based on the number of CPUs needed. Applying the map function alongside all the Dataframe (df_sample) parts, previously dived, will be distributed as arguments asynchronously to the same function (count_regex). Finally, after waiting for every pool to be finished, all results are concatenated into a single Dataframe, closing the pools. The new **df_results** is produced in this multiprocessing manner, later continuing the pipeline earlier described.

4.3. THE RESULTS AND CONCLUSION

For data anonymity reasons, the following results and their consequent conclusions will be based on a generated synthetic dataset, where each row represents single client information. This dataset was created for explanatory purposes, not corresponding to real-life data. Two main outputs are generated once the execution is complete – An Excel and an HTML Report.

4.3.1. HTML Report

Once the HTML Report is opened, it is possible to have a high-level overview vision of the processed data, including statistical summarizations and feature data types. In this case study, the number of variables and observations can be observed, as well as missing values, duplicate records, their absolute and relative frequency, and the total memory size. Furthermore, on the right side of the overview screen, there is an indication of each data type’s number of features. In this example, there are six numerical variables and four categorical (Figure 7).

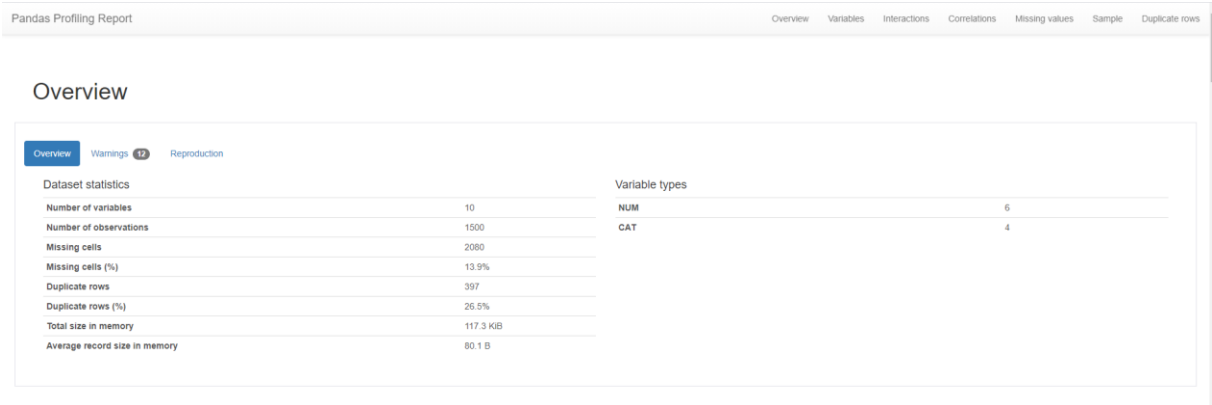


Figure 7. HTML Report Overview - Case Study

Additionally, a Warning tab is displayed next to the Overview tab. The Warning tab (Figure 8) provides upfront with the most critical information, such as the highly correlated features, missing values and their percentage, duplicate rows, and high cardinality. These warnings can also be found in the detailed section for each variable.

Overview



Figure 8. HTML Report - Warnings Tab

Afterwards, it is also possible to get more detailed information regarding each feature. Starting with categorical data, which, for explanatory and visual reasons, all four features will be explained together, the statistical information available for this type of data is less than for numerical data. For example, it is impossible to calculate the mean for Gender, so statistical information such as mean, median, and standard deviation is not calculated for categorical variables. Only the distinct and missing values, along with their percentage, are provided, having a Bar Plot that represents the frequency of each category for each feature. Four categorical variables are identified (Figure 9), the first being an **ID**, hence the high cardinality warning. It can be assessed that variable **ID** has no missing values and a high distinct value percentage, implying, as the name suggests, a unique identifier feature for each record. However, a relative frequency of 66.7% indicates the existence of duplicate **IDs**.

Variables

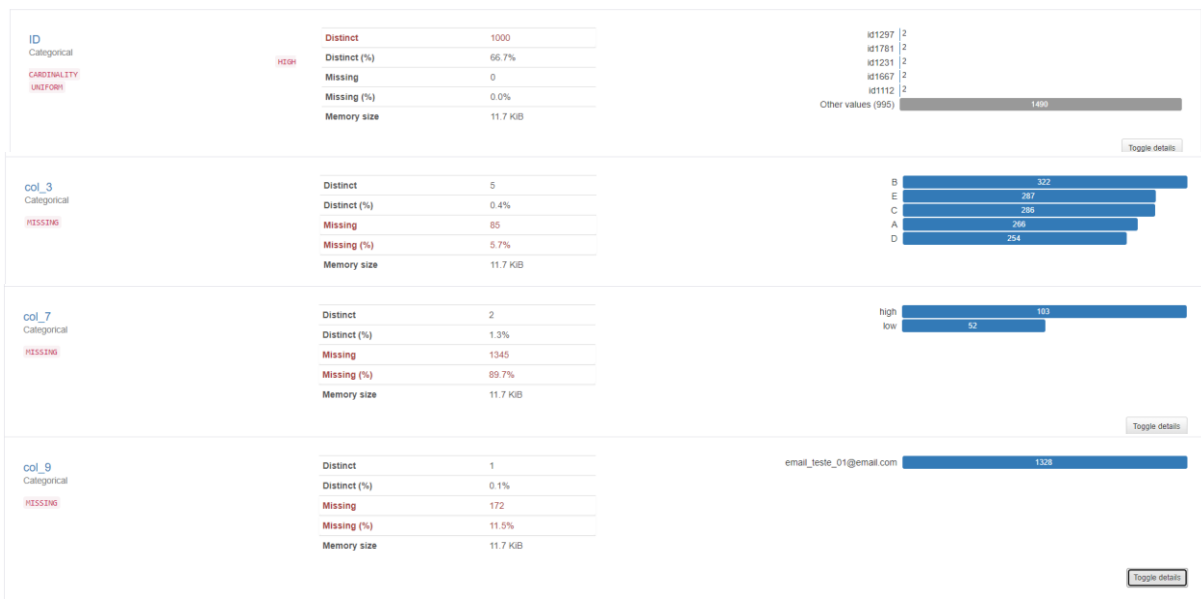


Figure 9. HTML Report - Information for Categorical Variables

The following variable, **col_3**, shows five distinct values with a small percentage of missing values, indicating the majority of each record belonging to a group *A*, *B*, *C*, *D* or *E*. If the widget to the right, **Toggle details**, is clicked, it displays the Common Values (Figure 10), the count and frequency in the percentage of each record belonging to each group, and the missing ones. It then directs to the Chart tab, revealing a Pie Plot representing the previous information in a more visual format (Figure 11). The Toggle details widget is available for every variable.

Value	Count	Frequency (%)
B	322	21.5%
E	287	19.1%
C	286	19.1%
A	266	17.7%
D	254	16.9%
(Missing)	85	5.7%

Figure 10. HTML Report - Common Values for Col_3

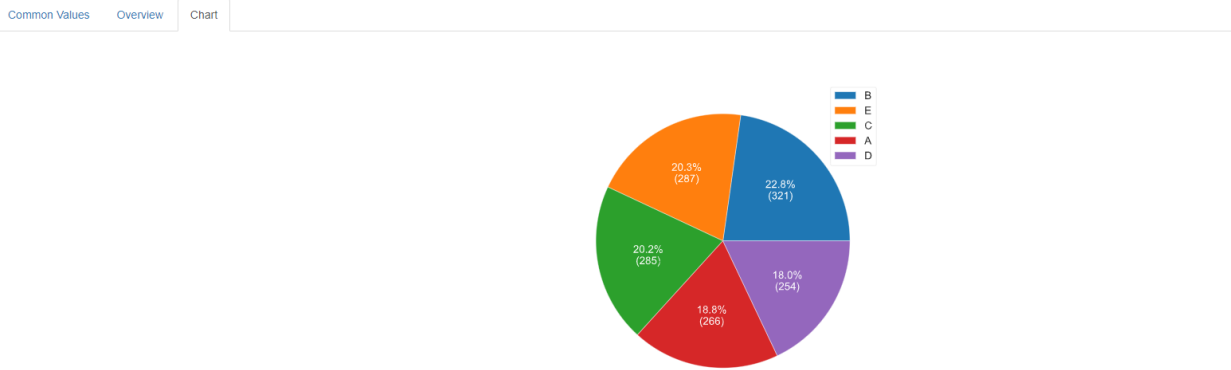


Figure 11. HTML Report - Chart for Col_3

The following categorical variables are **col_7** and **col_9** (Figure 9). For **col_7**, the report indicates two distinct values, either *low* or *high*. However, 87.7% of its data is missing, meaning that the inclusion of this feature in the analysis will not add much value. **Col_9** does not have a significant missing value percentage, although it does have only one distinct value - *email_teste_01@email.com* - indicating **col_9** is very likely to have e-mail information, but since it has only one distinct value, it also indicates that this information is not accurate, for each row represents a client. Therefore, it is expected for each client to have a unique e-mail address.

On the other hand, having six features from the dataset identified as numerical data types, a broader set of statistical information is presented instead of categorical data. Both the distinct and missing values and their percentage are calculated as well, yet, for numerical data, the mean, the minimum and maximum value, and the zero rates are provided. A graphical representation of each variable is also presented. Instead of a Bar Plot, a Histogram Chart which groups numerical data into bins, or intervals,

displaying them as segmented columns. This visualization is often used to depict the distribution of numerical data.

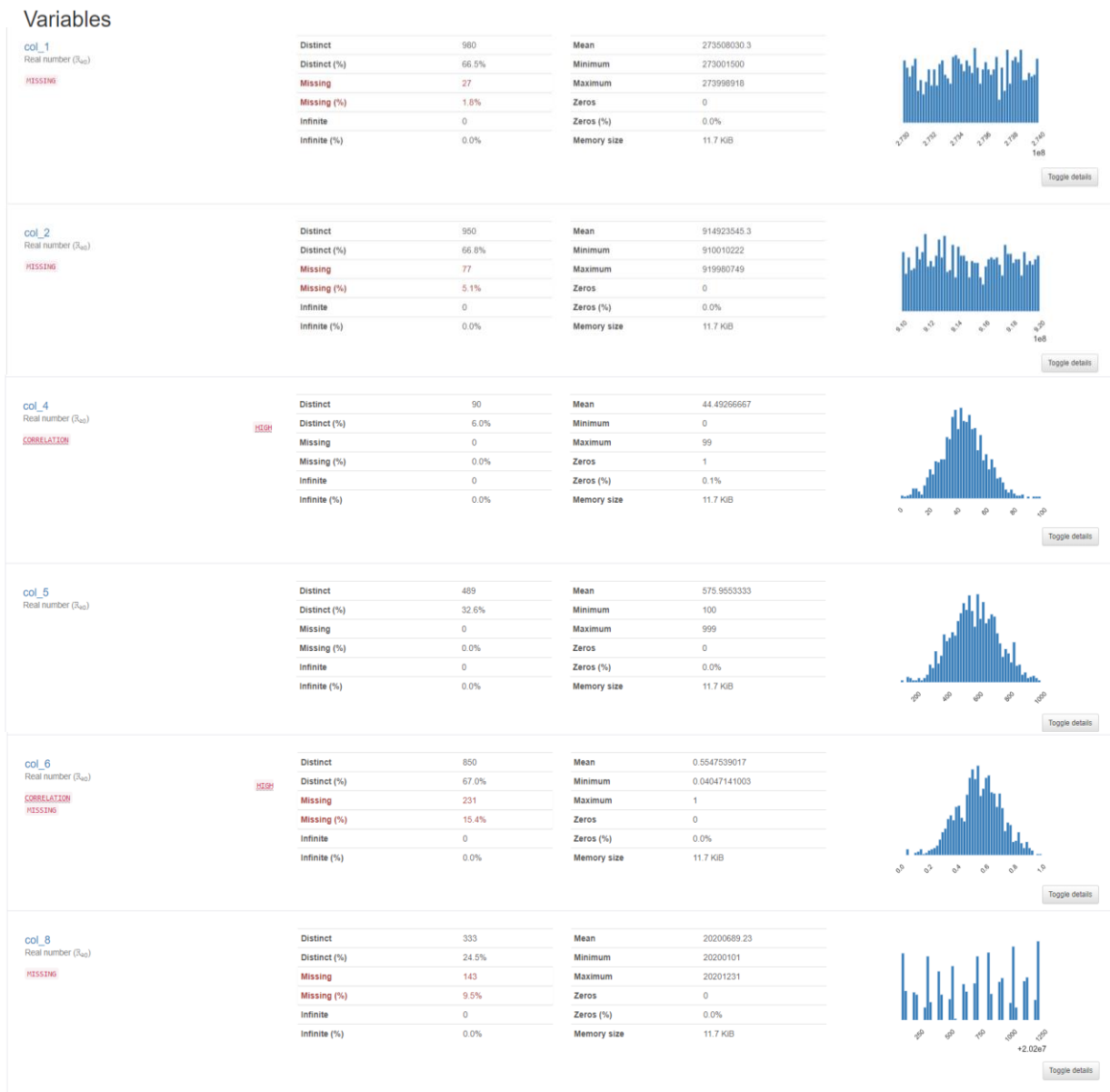


Figure 12. HTML Report - Information for Numeric Variables

For explanatory purposes, the **col_4** will serve as an example of the kind of information this HTML Report presents (Figure 12). First, there is an indication that **col_4** has 90 distinct and no missing values. Its values range between 0 and 99, with a mean of approximately 44,49 and an evenly distributed histogram. Once the Toggle details widget is pressed, a wider variety of statistical information appears, both quantile and descriptive. Regarding the quantile information, this report will provide both the minimum and maximum value and the median, which is the most central number in the values range, and for this case study presents as 44. It also provides the first (Q1) and third (Q3) quantile, representing the lowest 25% and the second-highest 25% values, respectively. Together with the Interquartile Range

(IQR), the difference between Q3 and Q1, indicating how data is distributed around the mean; the higher value IQR has, the more dispersed the data points are from the mean. In this case study, the IQR is 19, suggesting that the data points are more concentrated nearby the mean. Lastly, both the 5th and 95th percentile are given.

Furthermore, regarding the descriptive statistical analysis, this report presents the mean, variance, standard deviation indicating the average distance between a data point and the mean is an approximate value of 15,57. Also displaying the Coefficient of Variation (CV) - the ratio of the standard deviation to the mean, the greater the dispersion around the mean, the higher the coefficient of variance; the Kurtosis – the likelihood of data being heavy-tailed or light-tailed relative to a normal distribution, the higher the kurtosis, the higher the chance of being heavy-tailed and of having outliers; the Median Absolute Deviation (MAD) - the average distance between each data point and the mean, describing data variation in a dataset; and Skewness – the asymmetry that deviates from the symmetrical bell curve, or normal distribution. With the granted descriptive statistical information, it can be concluded for col_4 (Figure 13) that data points are not dispersed from the mean, for its CV is approximately 0.35. Also, it implies a light-tailed due to the low kurtosis value of closely 0.26. Finally, the skewness is proximately 0.13, confirming the initial assumption, resembling a normal distribution.

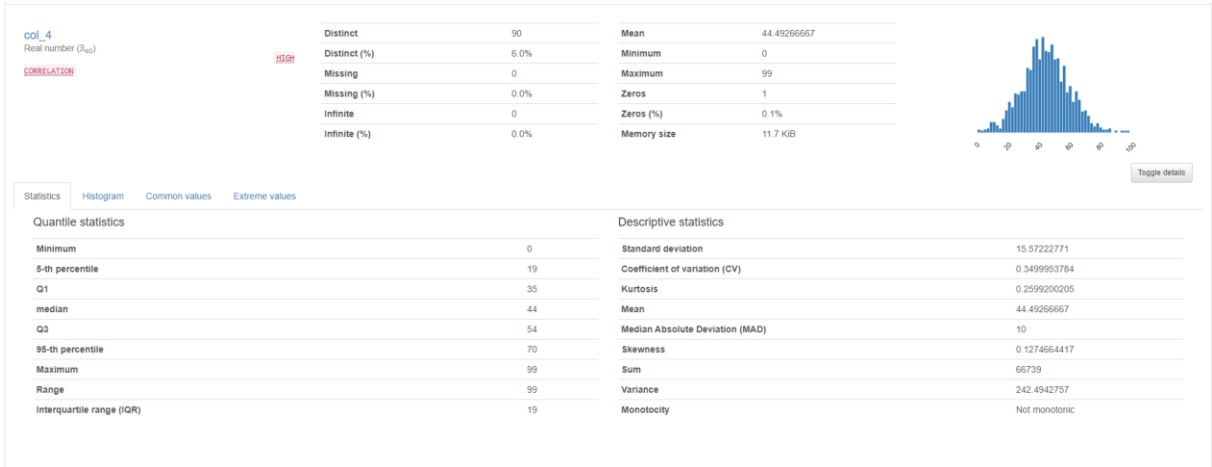


Figure 13. HTML Report - Statistics for Col_4

Having an analysis of the categorical and numerical data, this report offers other helpful features: the ability to show data samples, such as the top and bottom ten rows, the most frequent duplicated rows, and the graphical representation for the missing values. Most importantly, it displays the correlation between features with various methods (Pearson’s, Spearsman’s, Kendall’s, Philk and Cramér’s) (Figure 14) in the form of a Heatmap, a graphical representation of data that uses colour systems to represent different values.

Correlations

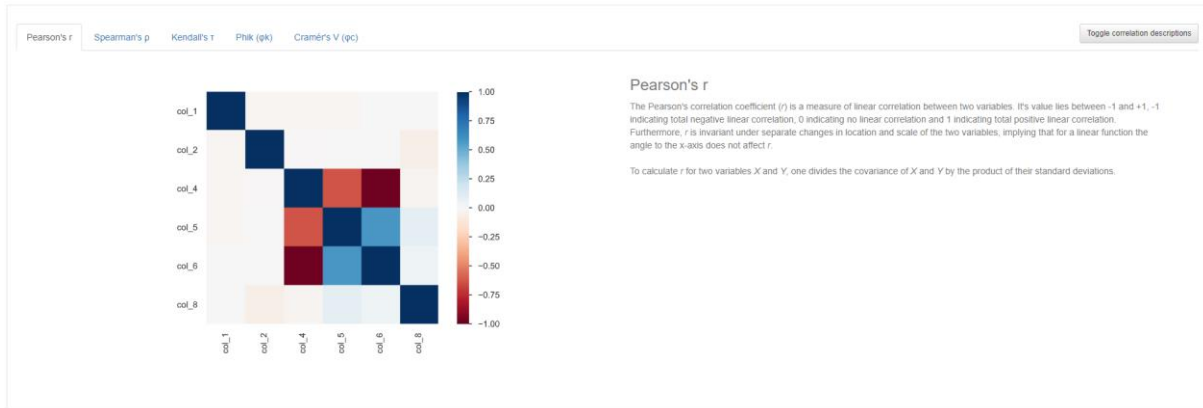


Figure 14. HTML Report - Correlations Feature

There is an essential final note to be made regarding the HTML Report. This information is only provided under the condition that the dataset size is within reasonable limits. In other words, for datasets of considerable sizes, like the ones used for this project, around 30GB, it will take a tremendous time to produce this output. When faced with a situation alike, a parameter is added to the command line - *minimal = True*, which creates an HTML Report with just the Overview, Warnings, and Variables information, discarding all the other components. However, all statistical information will remain.

4.3.2. Excel Report

The other generated output is an Excel file containing similar information as the HTML report. However, it has particular information the HTML report does not provide – the Regex validation. The first sheet is presented the same overview (Table 18) as in the previous report containing the number of rows and columns, the memory and records size, and finally, the missing values, duplicate records, as well as their absolute and relative frequency. Afterwards, a feature summarization (Table 19) including the number of occurrences, missing and distinct values, along with their relative frequency. However, in this output, the number of unique values and the correspondent relative frequency are also present. Distinct values are the set of different values a feature can contain within the dataset, where each value can appear once or multiple times in different rows. Typically, for categorical features, the number of distinct values is much smaller than the total data points. Contrarily, unique values appear once and only once in a specific feature, serving as a unique characteristic of that respective data row. Including, for example, unique identifier features (ID), where the number of distinct values of that feature equals the total number of observations and consequently equal to the unique values.

Table 18. Excel Report – Overview

OVERVIEW	
Number of Rows	1500
Number of Columns	10
Memory size (MB)	120128
Record size (b)	80.085333
Number of missing cells	2080
Percentage of missing cells	13.8667%
Number of duplicates	397
Percentage of duplicates	26.4667%

Table 19. Excel Report - Features Summary

	Number of Occurrences	Number of Distinct	Percentage of Distinct	Number of Missing	Percentage of Missing	Number of Unique	Percentage of Unique
ID	1500	1000	66.67%	0	0.00%	500	33.33%
Col_1	1473	980	66.53%	27	1.80%	487	33.06%
Col_2	1423	950	66.76%	77	5.13%	477	33.52%
Col_3	1415	5	0.03%	85	5.67%	0	0.00%
Col_4	1500	90	6.00%	0	0.00%	7	0.05%
Col_5	1500	489	32.60%	0	0.00%	117	7,80%
Col_6	1269	850	66.98%	231	15.40%	431	33.96%
Col_7	155	2	1.29%	1345	89.67%	0	0.00%
Col_8	1357	333	24.53%	143	9.53%	38	2.80%
Col_9	1328	1	0.00%	172	11.47%	0	0.00%

Lastly, in this second report, it is represented the one requirement the HTML did not fulfil and was demanded as a necessity for the management team of both Company_A and Company_B – the Regex Validation Result. One of this project's main goals was to detect specific patterns along with the data to improve the PII classification. As such, a list of Regex patterns (Figure 6) was created and compared with every column record. The result was the following table (Table 20) with the feature in question and

the corresponding analysis. The 1st Regex Key column shows the predominant Regex key in each component, the one with the highest number of matches to a given Regex pattern, followed by its absolute frequency. The 2nd Regex Key column represents, if it exists, the second-highest number of matches to a given Regex pattern, also followed by its absolute frequency.

Furthermore, the column Classification shows if a particular feature is, in fact, the Regex key. If the relative frequency is greater than an 80% threshold, then the feature is classified as the Regex key; on the contrary, if the relative frequency is below that threshold, the feature classification will be marked as Unknown. Consequently, when the classification value is Unknown, the column Observation will summarize the most frequent pattern and their relative frequency. However, if the feature is classified, then the column Observation will be “No Obs.”. If no observation matches the Regex pattern, all columns will be filled with “-”.

Table 20. Excel Report - Regex Validation Results

Feature	1 st Regex Key	Freq	2 nd Regex Key	Freq	Classification	Observation
ID	-	-	-	-	-	-
Col_1	Phone_number	0.4279	NIF	0.1007	Unknown	Most freq. pattern <Phone_number> (42.79%) Most freq. pattern <Cellphone_number> (10.07%)
Col_2	Cellphone_number	0.9934		0.0000	Cellphone_number	No Obs.
Col_3	-	-	-	-	-	-
Col_4	-	-	-	-	-	-
Col_5	-	-	-	-	-	-
Col_6	-	-	-	-	-	-
Col_7	-	-	-	-	-	-
Col_8	Date (YYYYMMDD)	1.0000		0.0000	Date (YYYYMMDD)	No Obs.
Col_9	E-mail	0.8853		0.0000	E-mail	No Obs.

4.3.3. Main Conclusions

With all the information gather, it is possible to compose a profile for the case study data. The following bullet points will review the conclusions for each feature in this case study.

ID. Categorical feature with no missing values and high cardinality due to the significant percentage of distinct values (66.67%). However, it has only 33.33% of unique values indicate duplicate records. This feature appears to be an ID and should be anonymized accordingly.

Col_1. Numerical feature with 1.8% of missing values, 66.5% of distinct records and 33.06% unique values suggesting duplicate records. According to the Regex validation, this feature appears to have some records similar to a Fiscal Number and a Portuguese Phone Number (e.g. 213876999), which have similar regex patterns. Although it does not have a relative frequency enough to be classified, it is more prone to be a Portuguese Phone Number rather than a Fiscal Number since it exists a logical function implemented to detect a Fiscal Number. In sum, it is presumable **col_1** to be a Portuguese Phone Number. However, since it does not achieve the desired classification threshold, it is advised to analyze this particular feature in-depth to assess if anonymization will be needed.

Col_2. A numerical feature with 5.1% of missing values, 66.8% of distinct records and 33.52% unique values suggesting duplicate records. According to the Regex validation, this feature appears to be a Portuguese Cell Number (e.g. 915939946), and it should be anonymized accordingly.

Col_3. A categorical feature with 5.7% of missing values, five distinct values – *A, B, C, D, E*. Each category is proximately evenly distributed, with category *B* having 21.5% of the records, being the highest amount. No match was found with any of the Regex patterns, so it most likely is not considered a PII. Hence it will not need to be anonymized.

Col_4, Col_5 & Col_6. Numerical features, with only 15.4% of missing values for **col_6**. All of these three features resemble a normal distribution. Both **col_4** and **col_6** have a high correlation between them and seems to have a lower CV, Kurtosis, and skewness values, meaning a small dispersion from the data points to the mean. However, **col_5** has a higher value for IQR and MAD, suggesting a wider data points dispersion from the mean. No match was found with any of the Regex patterns, so it is probably not considered a PII. Therefore, it will not need to be anonymized.

Col_7. A categorical feature with only two distinct values - *high* and *low*, and a considerable amount of missing values, nearly 90%, means this feature does not have proper information due to its incompleteness. No match was found with any of the Regex patterns, so it most likely is not considered a PII. Therefore, it will not need to be anonymized.

Col_8. A numerical feature with 9.5% of missing values, 24.5% of distinct records and 2.80% unique values suggesting duplicate records. Still, according to the Regex validation, this feature appears to be a date in YYYYMMDD format. As a result, it should be normalized to a timestamp format.

Col_9. A categorical feature with 11.5% of missing values and only one distinct values – *email_teste_01@email.com*. Due to the Regex validation, this feature seems to be an e-mail address, considered a PII. Thus, it should be correctly anonymized.

Throughout this analysis, it is possible to measure a DQ Index based on the chapter's 2.3.4 theoretical metrics. These academic metrics are established upon DQ dimensions, which, by themselves, have a subjective usage and application. This report proposes its measurable implementation by adapting these dimensions to the projects and case study reality. Each metric was calculated accordingly to its granularity, either by each cell or row. Therefore, it is possible to have a DQ Index that provides multiple measurable perspectives on a dataset's quality. In agreement with the metrics described in that chapter (Table 8), it is possible to reconstruct the following table (Table 21).

Table 21. DQ Dimensions - Definition, Metrics and Results

Dimension	Definition	Metric	Results
Accuracy	The degree of agreement with an identified source of correct information.	The correct data percentage (correct data/total data).	67% - 77% <i>per cell</i>
Completeness	The level of data missing or unusable.	Percentage of all complete data.	86.1% <i>per cell</i>
Consistency	The level of conflicting information.	Percentage of all consistent data.	100% <i>per cell</i>
Timeliness	The degree to which data is current and available for use in the expected time frame.	Percentage of all timeliness data (e.g. ages, educational degree at a particular time or date).	Not Applicable.
Uniqueness	The level of nonduplicates.	Percentage of all unique data (e.g. primary keys, foreign keys)	73.5% <i>per row</i>
Validity	The level of data matching a reference.	Percentage of all valid data (e.g. first name, last name, and suffix)	84.5% <i>per cell</i>

Either using a data sample or the total data volume will lead to the same results.

4.4. USEFUL APPLICATIONS

This sub-project's main goals were to understand better what types of data is being migrated and help classify the various feeds fields as PII when it is not apparent. If considered a PII, this field must be anonymized and normalized accordingly.

A Data Custodian (DC), someone in charge of designing, developing and operating the DWH (Giannoccaro et al., 1999), makes the PII classification. A DC must have strong DWH and business knowledge and insights to identify a PII correctly. Nevertheless, it is prone to human error, as such

complementary information resulting from a DP analysis mitigates the risk of uploading data to the Cloud that is not anonymized. Thus, resulting in a more accurate PII classification, it is easier to attribute the correct anonymization and normalization function. The anonymization processes are needed to secure clients identifiable data. Company_A developed a set of anonymizations functions according to the PII classification (Table 22). Using the randomization technique, replacing the actual values for random ones, eliminating the relationship between the data and the users.

Table 22. PII Examples & Anonymizations and Normalizations Functions

PII Name	Anonymization Functions	Normalization Functions	Normalization Output
Unique ID	bdpmanonptid	ptdmdereplacecr ptdmdetrimall	Hello[enter] there → Hello<<cr>>there “ Good Morning Maria ” → “Good Morning Maria”
Telephone Number	bdpmanonptmsisdn	ptcldecountrycode	910000000 → 351910000000
Postal Code (7 digits)	bdpmanonptpostcode	ptcldepostalcode7	1234-123 → 1234-123 234-3 → 0234-003
E-mail Address	bdpmanonptemailaddress	ptcldeemail	Email.Example@Email.com → email.example@email.com

Another practical application for DP analysis, beyond the DC skills, is to distinguish a particular type of information that is not consistent with its data type, going against the DQ dimension - Validity. For example, every information containing dates must be normalized into a timestamp format (YYY-MM-DD hh:mm:ss). However, on some occasions, information like Dates in the form “YYYYMMDD” presents itself, in the DB, as an integer data type rather than a datetime, making it challenging to be normalized. With the DP report, these particular cases can be recognized and dealt with, respectively.

5. CONCLUSIONS

With this work, the importance of a good storage solution and not blindly using data just because it is available were great lessons learned. It takes time and expertise to handle data before applying any modelling or jumping to conclusions. It is crucial to efficiently store information, such as in a Cloud Service, and know what information is being dealt with by outlining a profile and maintaining sound data quality.

The knowledge acquired from the Master's degree was fundamental, both intellectual and technical, throughout the internship. Without it, it would not be possible to have the analytical vision and insight need for this type of project. The theoretical concepts taught in the Data Science and Advanced Analytics Master explained how data could be store and organized, either on-premises and on the Cloud. Also, and most importantly, it gave the sensibility and know-how to deal, mine, and understand data to uncover or explain particular aspects leading to valuable insights for decision making.

Although the main challenges faced in this project were mainly its dimension and all the dependencies, this project was composed of many workstreams interconnected, for migrating a DWH to the GCP requiring good coordination and communication that were both lacking from time to time. Furthermore, the constant mutation and adjustments were an everyday struggle due to the complexity and dimension, thus tangling the project evolution. When encountered with some wrongly classified PIIs, the necessity to profile the data before migrating originated the sub-project described in this work – Data Profile. Lack of access and adversity to acquiring the connections needed some of the most significant limitations in that project.

Despite that, this project's aftermath was a good and learning experience. Accostumated with the academic world, this internship served as a reality check for the differences between academia and the corporate world. In companies, there are always more dependencies and hierarchies to address. The most significant difference, and perhaps the most important, was experienced with actual data and all its perks and downfalls. If there is one thing that can be concluded in this work: there is no perfect dataset, there is always room for improvement. All the work described in this report has a margin for improvement, namely the DP sub-project. Aspiring to create a better visual dashboard for a more rapid conclusion and integrate the DP pipeline in GCP, particularly Big Query optimizing processed feeds analysis. As the solution evolves, so must the person who creates it; this project provided a broader notion of the real world, how data is stored, whether in a DWH on-premises or the GCP. Politician Helmut Schmidt said, *“The biggest room in the world is the room for improvement.”*

6. BIBLIOGRAPHY

- Abedjan, Z., Golab, L., & Naumann, F. (2015). Profiling relational data: a survey. *VLDB Journal*, 24(4). <https://doi.org/10.1007/s00778-015-0389-y>
- Abiteboul, S. (1997). Querying semi-structured data. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/3-540-62222-5_33
- Atkins, D. E., Droegemeier, K. K., Feldman, S. I., García Molina, H., Klein, M. L., Messerschmitt, D. G., Messina, P., Ostriker, J. P., Wright, M. H., Garcia-molina, H., Klein, M. L., Messerschmitt, D. G., Messina, P., Ostriker, J. P., & Wright, M. H. (2003). Revolutionizing Science and Engineering Through Cyberinfrastructure: Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure. In *Science*.
- Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. *ACM Computing Surveys*. <https://doi.org/10.1145/1541880.1541883>
- Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data Science Journal*. <https://doi.org/10.5334/dsj-2015-002>
- Carroll, M., Van Der Merwe, A., & Kotzé, P. (2011). Secure cloud computing: Benefits, risks and controls. *2011 Information Security for South Africa - Proceedings of the ISSA 2011 Conference*. <https://doi.org/10.1109/ISSA.2011.6027519>
- Dai, W., Wardlaw, I., Cui, Y., Mehdi, K., Li, Y., & Long, J. (2016). Data profiling technology of data governance regarding big data: Review and rethinking. In *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-3-319-32467-8_39
- De, A., & Do, N. (2020). *Detecting and Protecting Personally Identifiable Information through Machine Learning Techniques*.
- Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems Sixth Edition. In *Database Systems*.
- Fahmideh, M., Daneshgar, F., Rabhi, F., & Beydoun, G. (2019). A generic cloud migration process model. *European Journal of Information Systems*, 28(3). <https://doi.org/10.1080/0960085X.2018.1524417>
- Fox, C., Levitin, A., & Redman, T. (1994). The notion of data and its quality dimensions. *Information Processing and Management*. [https://doi.org/10.1016/0306-4573\(94\)90020-5](https://doi.org/10.1016/0306-4573(94)90020-5)
- Giannoccaro, A., Shanks, G., & Darke, P. (1999). Stakeholder perceptions of data quality in a data warehouse environment. *Journal of Research and Practice in Information Technology*, 31(4).
- Gogtay, N. J., & Thatte, U. M. (2017). Principles of correlation analysis. *Journal of Association of Physicians of India*, 65(MARCH).
- Handbook of Cloud Computing. (2010). In *Handbook of Cloud Computing*. <https://doi.org/10.1007/978-1-4419-6524-0>
- Imhoff, C., Galemno, N., & Geiger, J. G. (2003). Mastering Data Warehouse Design: Relational and Dimensional Techniques. In *Wiley Publishing, Inc.*
- Jayawardene, V., Sadiq, S., & Indulska, M. (2013). An Analysis of Data Quality Dimensions. *ITEE Technical Report No. 2013-01*.

- Karr, A. F., Sanil, A. P., & Banks, D. L. (2006). Data quality: A statistical perspective. *Statistical Methodology*. <https://doi.org/10.1016/j.stamet.2005.08.005>
- Knight, S., & Burn, J. (2005). Developing a Framework for Assessing Information Quality on the World Wide Web Introduction – The Big Picture What Is Information Quality? *Informing Science Journal*.
- Manjunath, T., Hegadi, R., & Ravikumar, G. (2010). Analysis of data quality aspects in datawarehouse systems. *International Journal of Computer Science and Information Technologies*.
- Matsunaga, R., Ricarte, I., Basso, T., & Moraes, R. (2017). Towards an Ontology-Based Definition of Data Anonymization Policy for Cloud Computing and Big Data. *Proceedings - 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, DSN-W 2017*. <https://doi.org/10.1109/DSN-W.2017.28>
- MobiThinking. (2012). *Global mobile statistics 2012*. MobiThinking.
- Raghunathan, B. (2013). The Complete Book of Data Anonymization. In *The Complete Book of Data Anonymization*. <https://doi.org/10.1201/b13097>
- Rodrigues, A. (n.d.). *Data Pro ling : Identi cation of Data Quality Problems through Data Analysis 1 Motivation 2 Data Quality*. 1–10.
- Sainani, K. L. (2015). Dealing With Missing Data. *PM and R*. <https://doi.org/10.1016/j.pmrj.2015.07.011>
- Shanks, G., & Darke, P. (1998). Understanding Data Quality in a Data Warehouse. *Australian Computer Journal*.
- Siegel, A. F. (2012). Chapter 11 - Correlation and Regression: Measuring and Predicting Relationships. In *Practical Business Statistics (Sixth Edition)*.
- Tari, Z. (2014). Security and Privacy in Cloud Computing. *IEEE Cloud Computing*. <https://doi.org/10.1109/MCC.2014.20>
- Wang, R. Y. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*. <https://doi.org/10.1080/07421222.1996.11518099>
- Wang, Z., Yan, W., & Wang, W. (2020). Revisiting Cloud Migration: Strategies and Methods. *Journal of Physics: Conference Series, 1575*(1). <https://doi.org/10.1088/1742-6596/1575/1/012232>
- Web wisdom: how to evaluate and create information quality on the Web. (2000). *Choice Reviews Online*. <https://doi.org/10.5860/choice.37-2818>
- Zhang, S., Zhang, C., & Yang, Q. (2003). Data preparation for data mining. *Applied Artificial Intelligence*. <https://doi.org/10.1080/713827180>
- Zhao, J. F., & Zhou, J. T. (2014). Strategies and methods for cloud migration. *International Journal of Automation and Computing*. <https://doi.org/10.1007/s11633-014-0776-7>

