MASTER

DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK

INTERNSHIP REPORT

DATA ENGINEERING AND BEST PRACTICES

NAVID SAFFARI

SUPERVISION:

PROF. CARLOS J. COSTA

ENG. EMANUEL A. TAVARES

MARCH – 2023

TABLE OF CONTENTS

INDEX OF FIGURES

ABSTRACT

This report presents the results of a study on the current state of data engineering at LGG Advisors company. Analyzing existing data, we identified several key trends and challenges facing data engineers in this field. Our study's key findings include a lack of standardization and best practices for data engineering processes, a growing need for more sophisticated data management and analysis tools and data security, and a lack of trained and experienced data engineers to meet the increasing demand for data-driven solutions. Based on these findings, we recommend several steps that organizations at LGG Advisors company can take to improve their data engineering capabilities, including investing in training and education programs, adopting best practices for data management and analysis, and collaborating with other organizations to share knowledge and resources. Data security is also an essential concern for data engineers, as data breaches can have significant consequences for organizations, including financial losses, reputational damage, and regulatory penalties. In this thesis, we will review and evaluate some of the best software tools for securing data in data engineering environments. We will discuss these tools' key features and capabilities and their strengths and limitations to help data engineers choose the best software for protecting their data. Some of the tools we will consider include encryption software, access control systems, network security tools, and data backup and recovery solutions. We will also discuss best practices for implementing and managing these tools to ensure data security in data engineering environments. We engineer data using intuition and rules of thumb. Many of these rules are folklore. Given the rapid technological changes, these rules must be constantly reevaluated.

1. INTRODUCTION

The growth of data and customer needs, security threats, and the variety of software available is driving the market for data engineering. As more and more data are generated and collected, organizations need to develop efficient and effective ways to store, process, and analyze this data to gain insights and make informed decisions. Data engineers play a crucial role in this process by building the infrastructure and systems that enable data-driven decision-making. In addition, the increasing demands and expectations of customers are driving the need for data engineering. Customers today expect personalized, real-time experiences, and organizations need to be able to use data to understand and meet these expectations. Data engineers are responsible for building the systems that enable this level of customization and responsiveness.

Furthermore, security threats are a growing concern in the digital world, and data engineers are responsible for building and maintaining secure systems that protect organizations' data. This includes designing systems with security in mind, implementing security protocols, and regularly monitoring and testing for vulnerabilities. Attacks over the Internet are becoming more and more complex and sophisticated. How to detect security threats and measure Internet security is an important research topic. The variety of software available for data engineering presents opportunities and challenges.

On the one hand, there are many tools and technologies to choose from, allowing data engineers to find the best fit for their needs. On the other hand, this variety can make it difficult for data engineers to stay up-to-date and familiar with all available options. Data engineers must know various tools and technologies to effectively build and maintain data systems. Overall, data growth, customer needs, security threats, and the variety of software available drive the demand for data engineering to ensure that organizations can effectively manage and use their data securely and efficiently.

1.1 LGG Advisors

LGG Advisors provides knowledge services in the areas of Finance & Strategy, Risk & Compliance, Digital Transformation, Creative & Design, and Administrative & Support. They provide these services to clients across all industries and geographies. They operate as a "Strategic Thought Partner" for clients, working integrated with their teams and expanding their internal capacity flexibly at a reduced variable cost. Their clients benefit from accessing a wider pool of talented and skilled professionals with unique professional and educational backgrounds.

This company was founded in 2017 and is a young company that has quickly established itself as a leader in the financial consultancy industry. In just a few short years, the company has experienced rapid growth and has attracted a large and diverse client base worldwide. With its expertise and global reach, LGG Advisors has become a trusted partner for businesses and individuals looking to navigate the complex world of finance. The company offers various services, including financial and strategic planning, risk and compliance management, and creating reports and dashboards to help clients make informed decisions.

1.2 Objective

This project aims to design, build, and maintain efficient and reliable data pipelines collaboratively using Python and MySQL. These data pipelines will extract, transform, and load data from various sources into a central data repository or warehouse. In this project, we collaborate with other data team members to understand the organization's needs, identify and acquire data from various sources, clean and process the data, and load it into a suitable storage system for analysis and reporting. In this project, we also need to ensure that the data pipelines are scalable, reliable, and secure and monitor their performance to ensure that they meet the organization's data needs. In this project, we are aiming to change the way of processing and cleaning the data from using Power Query to using Python because we needed complex data manipulation that sometimes Power Query is not able to do that or crashes or sometimes it is very slow in processing the data, Python is a more versatile and powerful tool than Power Query for data cleaning and transforming. Python is a general-purpose programming language with a vast library of tools and functions, a large and active community of users, and better performance and automation capabilities.

Additionally, Python is free and open-source, making it more accessible to a wider range of users and organizations. While Power Query is user-friendly and suitable for simple data cleaning and transforming tasks within Excel or Power BI, Python is preferred for handling complex data manipulation tasks and large datasets. As we needed more advanced data cleaning, transforming, and loading, we needed to change our ETL tool from Power Query to Python. In the end, we need to automate the ETL process that we implement with Python, which can save us at least one hour per day also, we can process the data as we receive it and have our dashboards for the client in a close to real-time way.

1.3 Methodological Approach

Based on the objective of this project, the methodological approach will involve designing, building, and maintaining efficient and reliable data pipelines using Python and MySQL. The approach is similar to other approaches in the literature but with some specificities (Costa & Aparicio, 2020). This will involve collaborating with other data team members to understand the organization's needs, identify and acquire data from various sources, clean and process the data, and load it into a suitable storage system for analysis and reporting. To achieve this, the first step will be to understand the organization's data needs and identify the data sources that will be used in the project. The next step will be to design the data pipelines using Python and MySQL, which will involve selecting the appropriate tools and frameworks to perform the various data processing tasks. The data pipelines will be tested and validated to ensure that they are scalable, reliable, secure, and meet the organization's data needs. Performance monitoring tools will be used to track the data pipelines' performance and identify any potential bottlenecks or issues. The transition from using Power Query to Python will involve migrating the existing data cleaning and transforming workflows to Python and ensuring that the new workflows can handle complex data manipulation tasks and large datasets with better performance, automation, and reproducibility. The benefits of using Python over Power Query will be documented and analyzed.

The final step will be to automate the ETL process implemented with Python, which can save time and increase efficiency by processing data as it is received and presenting client dashboards in near-real-time. This will involve scheduling and monitoring tools to ensure the automated ETL process is reliable and functioning correctly. Overall, the methodological approach for this project will involve a collaborative effort among data analysts and engineers to design, build, and maintain efficient and reliable data pipelines using Python and MySQL. The iterative approach will involve testing and validation at every stage to ensure that the data pipelines meet the organization's data needs and are scalable, reliable, and secure.

2. LITERATURE REVIEW

Data engineering is a vital discipline that enables organizations to harness the power of their data assets. It involves designing, building, and maintaining data pipelines and systems that support the ingestion, transformation, and management of large-scale data sets. In this article, we will delve into the key concepts, techniques, and tools central to the practice of data engineering, including ETL solutions, Python and its use in big data, metrics for data quality, and data security. (O'Donovan et al., 2015, Vassiliadis & Simitsis, 2009) First, we will explore the concept of ETL (extract, transform, and load) solutions, which are commonly used in data engineering to facilitate data movement between different systems. We will discuss the advantages and challenges of ETL solutions and how they can be integrated into data pipelines to support data transformation and integration. (Petersen et al., 2022) Next, we will examine the role of Python in data engineering. We will discuss the benefits of using Python in data engineering projects, such as its extensive libraries and frameworks, and provide examples of how Python can be used to build data pipelines and perform data analysis. We will then turn our attention to metrics for data quality, which are critical for ensuring data accuracy, consistency, and reliability. We will discuss the different types of data quality metrics and how they can be used to assess and improve data quality in data pipelines. Finally, we will discuss the importance of data security in data engineering, including the risks and threats that organizations face when handling large-scale data sets. We will guide implementing measures in data pipelines and systems and highlight best practices for protecting sensitive data. In conclusion, this article has provided an overview of key concepts, techniques, and tools in data engineering, including ETL solutions, Python and its use in big data, metrics for data quality, and data security. Organizations can effectively extract value from their data assets and support data-driven decision-making by understanding these concepts and applying the appropriate techniques and tools.

2.1 ETL Process

ETL (Extract, Transform, Load) is a fundamental process in data engineering that involves extracting data from a source system, transforming it into a format suitable for analysis, and loading it into a target system. The process is vital in ensuring that data is available in a format that can be easily analyzed, ensuring accurate insights for decision-making (Caetano & Costa,. 2014). One of the key challenges in ETL processes is ensuring the scalability and reliability of the process. (Gray & Shenoy, n.d.) suggest that the best practices for ETL processes involve using automated tools, modularizing the process into smaller components, and ensuring that the process can handle varying volumes of data. Automating ETL processes eliminates manual errors and saves time. Modularizing the process ensures that individual components can be easily debugged and updated without affecting the overall process. Ensuring that the process can handle varying volumes of data is crucial in ensuring that the process can accommodate changing data demands ((Kimball & Ross, 2013) Another key consideration in ETL processes is optimization. (Simitsis & Vassiliadis, 2005) Suggest that optimizing ETL processes involves identifying bottlenecks, balancing the workload across multiple processors, and minimizing the impact of transformation logic on the process. Identifying bottlenecks involves identifying the slowest components in the process and optimizing them. Balancing the workload across multiple processors ensures that the process can handle large volumes of data. Minimizing the impact of transformation logic on the process involves optimizing the transformation logic and ensuring that it can run in parallel to minimize the processing time.

2.1.1 Extract Transform Load (ETL) solutions

Extract Transform Load (ETL) solutions are a vital component of modern data engineering, enabling organizations to move, transform, and integrate data between different systems. This thesis will explore the advantages and challenges of using ETL solutions and discuss how they can be integrated into data pipelines to support data transformation and integration (Kimball & Ross, 2013, Caetano & Costa, 2014). First, we will define ETL solutions and provide an overview of their key features and functions. We will discuss the benefits of using ETL solutions, such as their ability to support data integration across different systems and their flexibility in handling various data types and formats. Next, we will examine the challenges of using ETL solutions, including data quality, performance, and security issues. We will discuss strategies for addressing these challenges, such as implementing data cleansing and validation processes, optimizing ETL performance, and securing data in transit and at rest. Finally, we will describe how ETL solutions can be integrated into data pipelines to support data transformation and integration. We will discuss different approaches to data transformation, such as data cleansing, Aggregation, and Normalization, and provide examples of how ETL solutions can be used to implement these processes. (Gokhale et al., 2021). In conclusion, this thesis has comprehensively analyzed ETL solutions and their role in data engineering. Organizations can derive maximum value from their data assets and support data-driven decision-making by understanding the advantages and challenges of using ETL solutions and integrating them effectively into data pipelines.

2.1.2 ETL solutions and their key features and functions

ETL (Extract, Transform, Load) solutions are software tools that facilitate the extraction of data from a variety of sources, the transformation of that data into a suitable format for analysis and reporting, and the loading of that data into a destination such as a data warehouse or data lake. These solutions are commonly used in data engineering and data analytics to manage and process large volumes of data from various sources, making them accessible for analysis and reporting. ETL solutions offer a range of features and functions, including the ability to extract data from multiple sources, transform data into the required format and structure, load data into a destination, schedule the execution of ETL jobs, track the lineage of data, and ensure the quality and integrity of the data. Traditionally, ETL processes run on a periodic basis (weekly, daily). With the increasing popularity of data warehouses and data marts, the ability to refresh data in a timely fashion is more important than ever. (Kimball & Ross., 2013, Tank, 2012). Here are the key features and functions of ETL solutions that we mentioned above(Kimball & Ross,2013): Data extraction, Data transformation, Data loading, Data scheduling, Data lineage and traceability, and Data quality and error handling. ETL solutions allow extracting data from various sources such as databases, text files, and APIs. ETL solutions allow transforming data into the required format and structure, such as converting data types, filtering and selecting specific columns, and applying transformations and calculations. ETL solutions allow the loading of the transformed data into a destination, such as a data warehouse or data lake, for further analysis and reporting. ETL solutions allow scheduling the execution of ETL jobs at regular intervals, such as daily, weekly, or monthly. ETL solutions allow tracking the lineage of data from its source to its destination, making it easier to understand how the data has been transformed and where it came from. ETL solutions allow for ensuring the quality and integrity of the data by validating it and handling errors and exceptions.

Overall, ETL solutions play a crucial role in data engineering and analytics by enabling the extraction, transformation, and loading of data from various sources for analysis and reporting (Biswas, Sarkar, & Mondal, 2019).

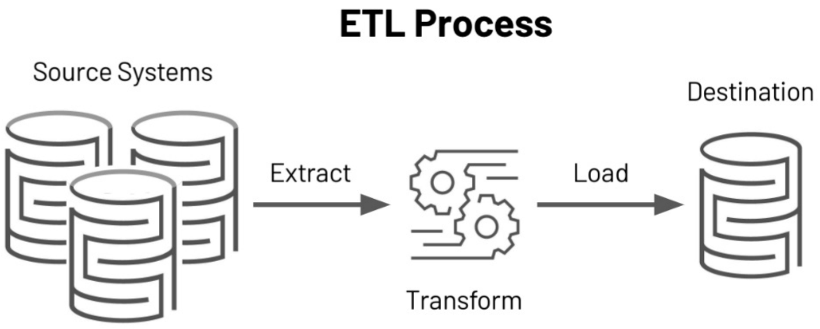


Figure 1 ETL Process

(https://www.databricks.com/wp-content/uploads/2021/05/ETL-Process.jpg)

2.1.3 Challenges of using ETL solutions and possible strategies for addressing them

While ETL solutions provide many benefits, some challenges can arise. Some common challenges of using ETL solutions include Data quality, transformation, performance, and security. (Souibgui et al., 2019). The challenges mentioned above are described below: Data quality, Data transformation, Data performance, and Data security. Ensuring the quality and integrity of the data during the ETL process can be challenging. Data may be dirty, incomplete, or inconsistent, which can lead to errors and affect the accuracy of the results. ETL solutions may provide some data cleansing and validation features. Still, it is often up to the data engineer to ensure the data is clean and accurate before loading it into the destination. Transforming data into the required format and structure can be complex, especially when working with extensive data or data with complex systems. Applying difficult changes and calculations may be necessary, impacting performance and efficiency. ETL solutions may have performance issues when working with large volumes of data, especially when data needs to be transformed and loaded in real-time. Optimizing the ETL process and the destination may be necessary to ensure that it can handle the volume of data and provide fast query performance. Ensuring the security of sensitive data during the ETL process can be challenging, as handling and storing sensitive data may be necessary during the transformation and loading process. ETL solutions may provide some security features, such as data masking and encryption, but it is up to the data engineer to ensure that the data is protected. Here are a few strategies that can be effective in addressing the challenges of using ETL solutions: Data cleansing and validation, Optimizing ETL performance, and Securing data. Data cleaning, also called data cleansing or scrubbing, deals with detecting and removing errors and inconsistencies from data to improve data quality. Data quality problems are present in single data collections, such as files and databases, e.g., due to misspellings during data entry, missing information, or other invalid data (Lomet et al., n.d.). Implementing data cleansing and validation processes can help ensure the data's quality and integrity during the ETL process. This may involve applying transformations and calculations to cleanse and enrich the data and applying validation rules to ensure the data is accurate and complete.

Several strategies can be used to optimize the performance of the ETL process, including: Partitioning data, Indexing data, and Tuning ETL jobs. Splitting large data sets into smaller chunks can help to improve performance and reduce the load on the ETL system. Indexing data can help to improve the speed of data queries and reduce the load on the ETL system. Adjusting the configuration of ETL jobs, such as the number of parallel processes, can help to optimize performance (Seenivasan, 2023). Ensuring the security of sensitive data during the ETL process is essential. This may involve implementing measures such as data masking, data encryption, and authentication to protect data in transit and at rest. Access controls and monitoring may also be necessary to ensure that only authorized users can access sensitive data. In the image below, a few strategies are mentioned for ETL testing and validating the pipeline.

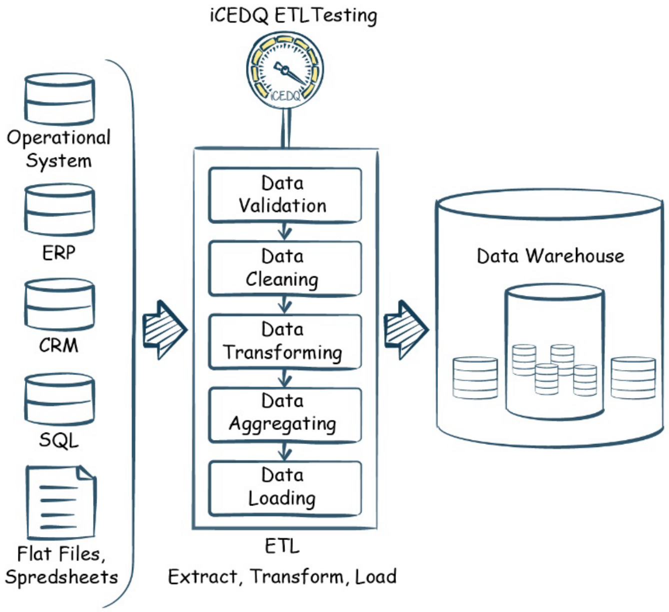


Figure 2 iCEDQ ETL Testing

https://icedq.com/wp-content/uploads/2022/04/ETL-Testing-Importance-iceDQ.jpg

Eventually, these strategies can help to address the challenges of using ETL solutions and ensure the success of the ETL process.

2.1.4 Integration of ETL solution into data pipelines and examples

ETL solutions can be integrated into data pipelines to support data transformation and integration by extracting data from various sources, transforming it into the required format and structure, and loading it into a destination such as a data warehouse or data lake. Data pipelines are processing steps used to extract, transform, and load data from various sources, making it available for analysis and reporting. ETL solutions are often used to implement these processing steps, allowing organizations to integrate and transform data from multiple sources and systems. Several approaches to data transformation can be implemented using ETL solutions, and there are also examples, including Data cleansing, Aggregation, and Normalization. Data cleansing is identifying and correcting errors and inconsistencies in data. ETL solutions can apply transformations and calculations to cleanse data, such as removing duplicates, filling in missing values, and standardizing data formats. For example, an ETL solution can be used for formations and calculations to remove duplicates and fill in missing values in a data set. Aggregation combines data from multiple sources or groups into a cohesive view. ETL solutions can be used to apply aggregation functions, such as sum, average, and count, to the group and summarize data. For instance, an ETL solution can be used to apply aggregation functions, such as sum and average, to the group and summarize data from multiple sources. Normalization is the process of transforming data into a consistent format, such as converting data types or applying transformations to standardize data. ETL solutions can be used to apply normalization processes, such as converting data types, standardizing date formats, and applying calculations to standardize data. An ETL solution can convert data types, standardize date formats, and apply measures to standardize data.

2.2 Python and its use in data engineering

Python is a popular programming language that is widely used in data engineering projects due to its extensive libraries and frameworks, as well as its ease of use and readability. Some benefits of using Python in data engineering projects include Extensive libraries and frameworks, Ease of use and readability, and Flexibility (Crickard, 2020). Python has many libraries and frameworks designed explicitly for data engineering and analytics tasks, such as Pandas for data manipulation and analysis, NumPy for numerical computing, and Scikit-learn for machine learning (Pedregosa et al., 2011). Python is known for its simplicity and readability, making it an accessible language for data engineers with different experience levels. It is also easy to debug and maintain Python code, saving time and effort during development. Python is a universal language that can be used for a wide range of data engineering tasks, including data extraction, transformation, loading, data analysis, and visualization. Here are a few examples of how Python can be used in data engineering projects: Building data pipelines, Performing data analysis, and Visualizing data. Python can build data pipelines using libraries such as Pandas and NumPy to extract, transform, and load data from various sources. Python can perform data analysis using libraries such as Pandas and Scikit-learn. For example, Python can be used to build machine learning models, perform statistical analysis, and visualize data. Python has several libraries, such as Matplotlib (Barrett et al., 2005) and Seaborn (Waskom, 2021).), that can be used to create data visualizations, allowing data engineers to understand better and communicate their findings. Python is a powerful language widely used in data engineering projects due to its extensive libraries and frameworks, ease of use, and flexibility. It can be used to build data pipelines, perform data analysis, and visualize data, making it an essential tool for data engineers. Python and GitHub can be utilized to create data pipelines for extracting, transforming, and loading data from various sources. Data is first extracted from a source using Python libraries such as Pandas, PyMySQL, or MySql to create a data pipeline. The extracted data is then transformed using libraries such as Pandas or NumPy before being loaded into the destination with the assistance of libraries such as PyMySQL or psycopg2. GitHub can store the data pipeline and provide version control, allowing multiple data engineers to collaborate on and track changes to the pipeline. This way, Python and GitHub can be used together to create a data pipeline that extracts, transforms, and loads data from a source to a destination, with GitHub providing essential version control and collaboration features for data engineering projects (Crickard, 2020).

2.3 Data Quality

Data quality refers to the accuracy, completeness, and consistency of data. It is crucial in ensuring that data-driven decisions are accurate and reliable. (Pipino et al., 2002) suggest that data quality assessment involves six dimensions: completeness, consistency, timeliness, accuracy, uniqueness, and validity. Completeness refers to the extent to which all required data elements are present. Consistency refers to the extent to which the data is internally consistent and free of contradictions. Timeliness refers to the extent to which the data is up-to-date. Accuracy refers to the extent to which the data reflects the true values. Uniqueness refers to the extent to which the data is unique and does not contain duplicates. Validity refers to the extent to which the data conforms to the defined business rules and constraints. Ensuring data quality involves a combination of data profiling, data cleansing, and data enrichment techniques. Data profiling involves data analysis to identify patterns, relationships, and anomalies. Data cleansing involves identifying and correcting errors, inconsistencies, and duplicates in the data. Data enrichment involves enhancing the data by adding missing data, correcting errors, or merging data from multiple sources.

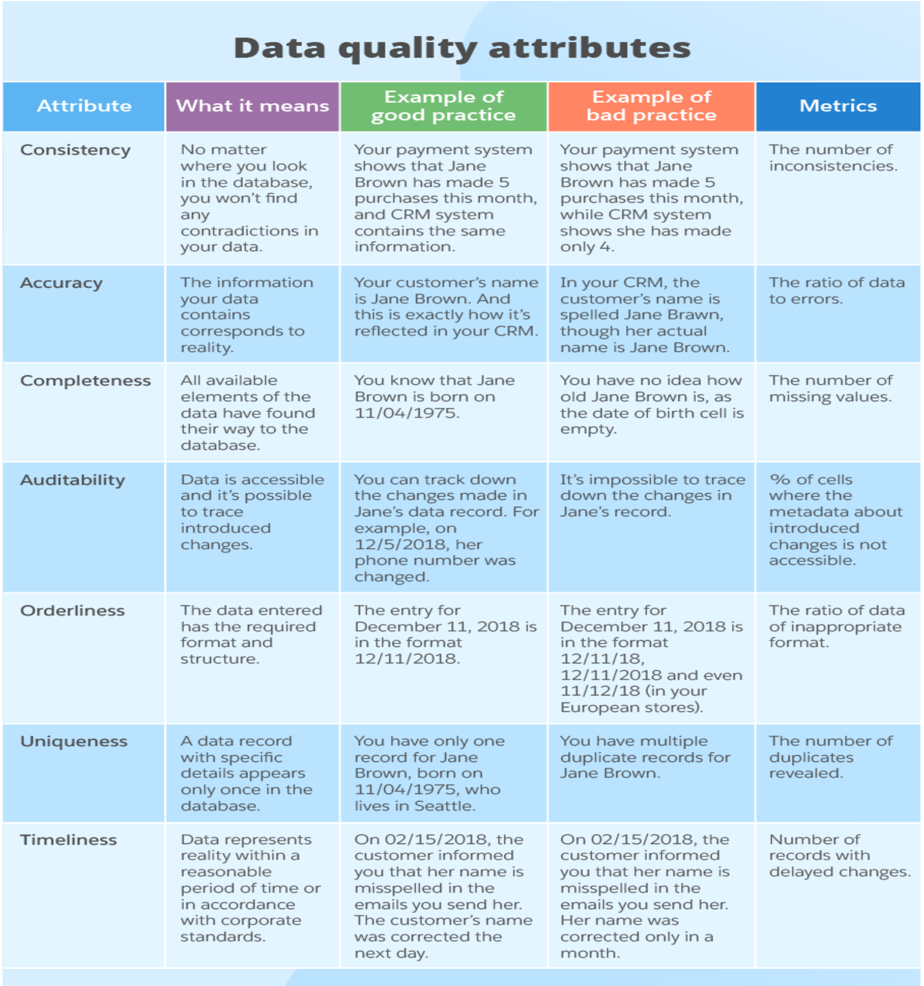
2.3.1 Metrics for data quality

Data quality refers to the data set's overall level of excellence in accuracy, completeness, consistency, timeliness, and validity. Data quality is essential in any field that relies on data, such as data engineering, data analytics, and business intelligence. Data quality can lead to correct conclusions, flawed decision-making, and lost opportunities. (Heinrich, 2018; Pipino et al., 2002)

To ensure data quality, it is essential to implement processes and procedures for collecting, storing, and managing data and ensure that data is accurate, complete, consistent, timely, and valid. This may involve implementing data cleansing and validation processes and applying rules and constraints to ensure the integrity of the data.

There are several ways in which data quality can be measured in a company which are Data quality scorecards, Data quality dashboards, Data quality rules, Data profiling, and Data testing, which are described in more detail in the following: (Ehrlinger & Wöß, 2022): Data quality scorecards, Data quality dashboards, Data quality rules, Data profiling, Data testing. Data quality scorecards summarize data quality across different data domains or sources. The scorecards can be used to track data quality over time and identify areas where improvement is needed. Data quality dashboards provide a visual representation of data quality, highlighting areas where the data meets or does not meet specific quality standards. The dashboards can be used to improve data quality in real-time and identify issues as they arise. Data quality rules define specific standards that data must meet to be considered high quality. The rules can be used to check the data quality and identify any issues that need to be addressed. Data profiling involves analyzing the data to understand its characteristics, such as distribution, completeness, and consistency. Data profiling can be used to identify issues with data quality and develop strategies for improving it. Data testing involves verifying the accuracy and completeness of data by comparing it to a known standard or reference data set. Data testing can identify issues with data quality and ensure that it meets specific standards. (Ehrlinger & Wöß, 2022). For the ways mentioned above of measuring data quality, several metrics can be implemented in each form of measuring. The metrics for data quality can vary depending on the specific requirements and context of the data. However, there are a few key metrics that are generally considered necessary for data quality across different types of data, which contain the following properties Accuracy, Completeness, Consistency, Timeliness, and Validity (Ehrlinger & Wöß, 2022; Pipino et al., 2002): Accuracy, Completeness, Consistency, Timeliness, Validity. Accuracy refers to the degree to which the data is correct and free from errors. Ensuring data accuracy is essential for all data types, as incorrect data can lead to erroneous conclusions and flawed decision-making. Completeness refers to the extent to which the data is complete and contains all required information. Ensuring the completeness of data is essential for all types of data, as incomplete data can lead to incorrect conclusions and flawed decision-making. Consistency refers to the degree to which the data is consistent with itself and other data sources. Ensuring data consistency is essential for all types of data, as inconsistent data can lead to confusion and make it difficult to draw accurate conclusions.

Timeliness refers to the relevance of the data, with timely data being more relevant and useful for decision-making. The importance of timeliness can vary depending on the context of the data. Validity refers to the degree to which the data conforms to its rules and constraints. Ensuring data validity is essential for all data types, as invalid data can lead to errors and compromise the data's integrity.



Data quality attributes

Attribute	What it means	Example of good practice	Example of bad practice	Metrics
Consistency	No matter where you look in the database, you won't find any contradictions in your data.	Your payment system shows that Jane Brown has made 5 purchases this month, and CRM system contains the same information.	Your payment system shows that Jane Brown has made 5 purchases this month, while CRM system shows she has made only 4.	The number of inconsistencies.
Accuracy	The information your data contains corresponds to reality.	Your customer's name is Jane Brown. And this is exactly how it's reflected in your CRM.	In your CRM, the customer's name is spelled Jane Brawn, though her actual name is Jane Brown.	The ratio of data to errors.
Completeness	All available elements of the data have found their way to the database.	You know that Jane Brown is born on 11/04/1975.	You have no idea how old Jane Brown is, as the date of birth cell is empty.	The number of missing values.
Auditability	Data is accessible and it's possible to trace introduced changes.	You can track down the changes made in Jane's data record. For example, on 12/5/2018, her phone number was changed.	It's impossible to trace down the changes in Jane's record.	% of cells where the metadata about introduced changes is not accessible.
Orderliness	The data entered has the required format and structure.	The entry for December 11, 2018 is in the format 12/11/2018.	The entry for December 11, 2018 is in the format 12/11/18, 12/11/2018 and even 11/12/18 (in your European stores).	The ratio of data of inappropriate format.
Uniqueness	A data record with specific details appears only once in the database.	You have only one record for Jane Brown, born on 11/04/1975, who lives in Seattle.	You have multiple duplicate records for Jane Brown.	The number of duplicates revealed.
Timeliness	Data represents reality within a reasonable period of time or in accordance with corporate standards.	On 02/15/2018, the customer informed you that her name is misspelled in the emails you send her. The customer's name was corrected the next day.	On 02/15/2018, the customer informed you that her name is misspelled in the emails you send her. Her name was corrected only in a month.	Number of records with delayed changes.

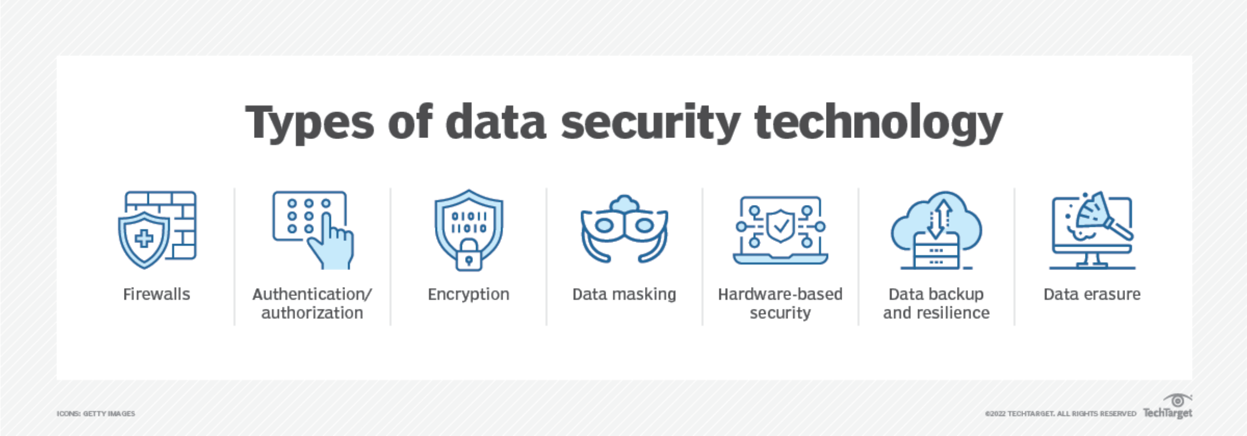
Figure 3 Data quality attributes

https://www.scnsoft.com/blog-pictures/business-intelligence/data-quality-management-attributes_1.png

The specific metrics for data quality may vary depending on the data's context and purpose.

2.4 Data Security

Data security refers to protecting data against unauthorized access, use, disclosure, or destruction. Given the sensitive nature of data in today's business environment, it is a critical aspect of data engineering. (Bertino, 2016) suggests that data security involves three main concepts: confidentiality, integrity, and availability. Confidentiality refers to the protection of data against unauthorized access. This involves ensuring that only authorized personnel have access to sensitive data. Integrity refers to the protection of data against unauthorized modification. This involves ensuring that data is accurate and reliable. Availability refers to the protection of data against unauthorized interruption or destruction. This involves ensuring data is available and accessible to authorized personnel when needed. Data security in data engineering refers to the measures and processes to protect data from unauthorized access, misuse, or damage. Data security is essential in data engineering, as it helps ensure data confidentiality, integrity, and availability. Several measures can be taken to ensure data security in data engineering. Some of the most important ones are Data encryption, Access controls, and Data backup and disaster recovery (Drosio & Stanek, 2016): Data encryption, Access controls, and Data backup and disaster recovery. Data encryption involves converting data into a coded form that can only be accessed with the appropriate key or password. Data encryption can be used to protect data in transit and at rest. Access controls are measures put in place to ensure that only authorized users have access to data. This may involve implementing authentication and authorization processes to verify the identity of users and ensure that they have the appropriate permissions to access data. Data backup and disaster recovery measures are put in place to ensure that data can be recovered in the event of a disaster or data loss. This may involve implementing regular backups of data and having a plan for recovering data in the event of a disaster. Here are some types of data security technologies mentioned in the figure below.


Types of data security technology
Firewalls
Authentication/ authorization
Encryption
Data masking
Hardware-based security
Data backup and resilience
Data erasure
ICONS: GETTY IMAGES
©2022 TECHTARGET. ALL RIGHTS RESERVED
TechTarget

((Makaranka), 2018)

Figure 4 types of data security technology

https://cdn.ttgtmedia.com/rms/onlineimages/types_of_data_security_technology-f.png

Data security is essential in data engineering, as it helps ensure data confidentiality, integrity, and availability. Several measures can be taken to ensure data security. (Davis, 1978)

2.4.1 Data encryption

Data encryption is converting data into a coded form that can only be accessed with the appropriate key or password. Data encryption is a commonly used method for protecting data and ensuring data security, making it more difficult for unauthorized users to access sensitive information. There are several ways in which data encryption can be used to protect data: Protecting data in transit and Protecting data at rest. Data encryption can be used to protect data transmitted over networks or through the internet. This is especially important for sensitive data, such as financial or personal information, as it helps prevent unauthorized access or interception. Data encryption can also be used to protect data stored on devices or databases. This is important for protecting data from unauthorized access or tampering and meeting regulatory requirements for data security.

2.4.2 Data access control

Data access control is a set of measures and processes that are put in place to ensure that only authorized users have access to data. This can be achieved through authentication, which verifies the identity of users, and authorization, which determines which users or groups have access to which data and resources. Data access control can also be enhanced through data encryption, auditing, and monitoring to track access to data and identify any unauthorized access or suspicious activity. In this way, data access control helps to protect data and ensure the security and confidentiality of sensitive information. (Masoumzadeh & Joshi, n.d.)

2.4.3 Data backup and disaster recovery

Data backup and disaster recovery is a critical aspect of data security that involves implementing processes and measures to ensure the availability and integrity of data in the event of a disaster or data loss. This includes creating regular backups of data and storing them in a separate location, such as an offsite data center, and developing and implementing a disaster recovery plan that outlines the steps to be taken in the event of a disaster. Data restoration, the process of recovering data from backups or other sources in the event of data loss or tragedy, is an essential part of data backup and disaster recovery and helps to ensure that data is available and can be used to restore systems and operations. By implementing data backup and disaster recovery measures, organizations can protect their data and ensure the continuity of their operations in the event of a disaster or data loss. (Al-Shammari & Alwan, 2018).

3 EMPIRICAL WORK

As a data engineer intern at LGG Advisors in Lisbon, the intern had the opportunity to work on a project that involved creating and maintaining ETL pipelines using Python and ensuring the quality and security of the data being processed. LGG Advisors is a consulting firm that provides financial solutions to clients in various industries. The project that is being worked on concerns receiving large amounts of financial data every day and creating many data pipelines to extract, transform and load the data into a database and import it from the database to the Data analysis (Microsoft PowerBI) for creating reports and dashboard and giving a presentation of the analysis of the data to the client every week.

3.1 Project Description:

This project involved receiving financial data from two sources, CSV and Excel files, and creating a pipeline to process the data. The pipeline consisted of several steps: data validation, cleaning, and transformation. The pipeline's outcome was loading the processed data into a MySQL database for analysis. This project's most complicated challenges were cleaning the CSV and Excel files. Those files are typically very unorganized and unsuitable for feeding the database, which requires efficient processing and expertise in working with Python and Pandas, also caring about quality and security has always been one of the most significant concerns about this project because any data leakage can have irreparable consequences for the company and hard damages.

In the following paragraphs, the main tools and technologies used in the company are described. As it is always required in the field of Data Engineering and Data Analytics to use various tools for project management, programming, and database administration, in this project, we used multiple tools and technologies to accomplish the project's goals, such as Python, MySQL, and GitHub. Python was used for most of the data processing and pipeline development, with libraries such as Pandas, NumPy, OS, and many other libraries for data manipulation and analysis and also used MySQL.connector, which is for connecting to the database, committing queries, and feeding the database with the daily data that we receive. MySQL was used as the database management system to store the processed data, creating queries to help us check the data quality. For example, can it be reviewed until the database is updated, or have we received the number of specific stores we should receive daily? If not, informing the client about the missing data and GitHub was used for version control and cooperation with our other teammates because, without GitHub, it is almost impossible to work on one project with many people to fix bugs and control versions. These tools and technologies allowed us to efficiently process and store large amounts of data and work in the best way with other members of the digital transformation department.

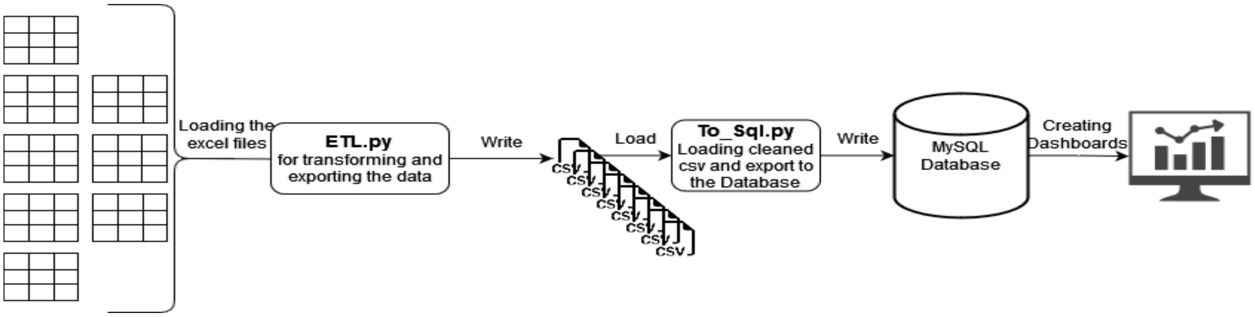
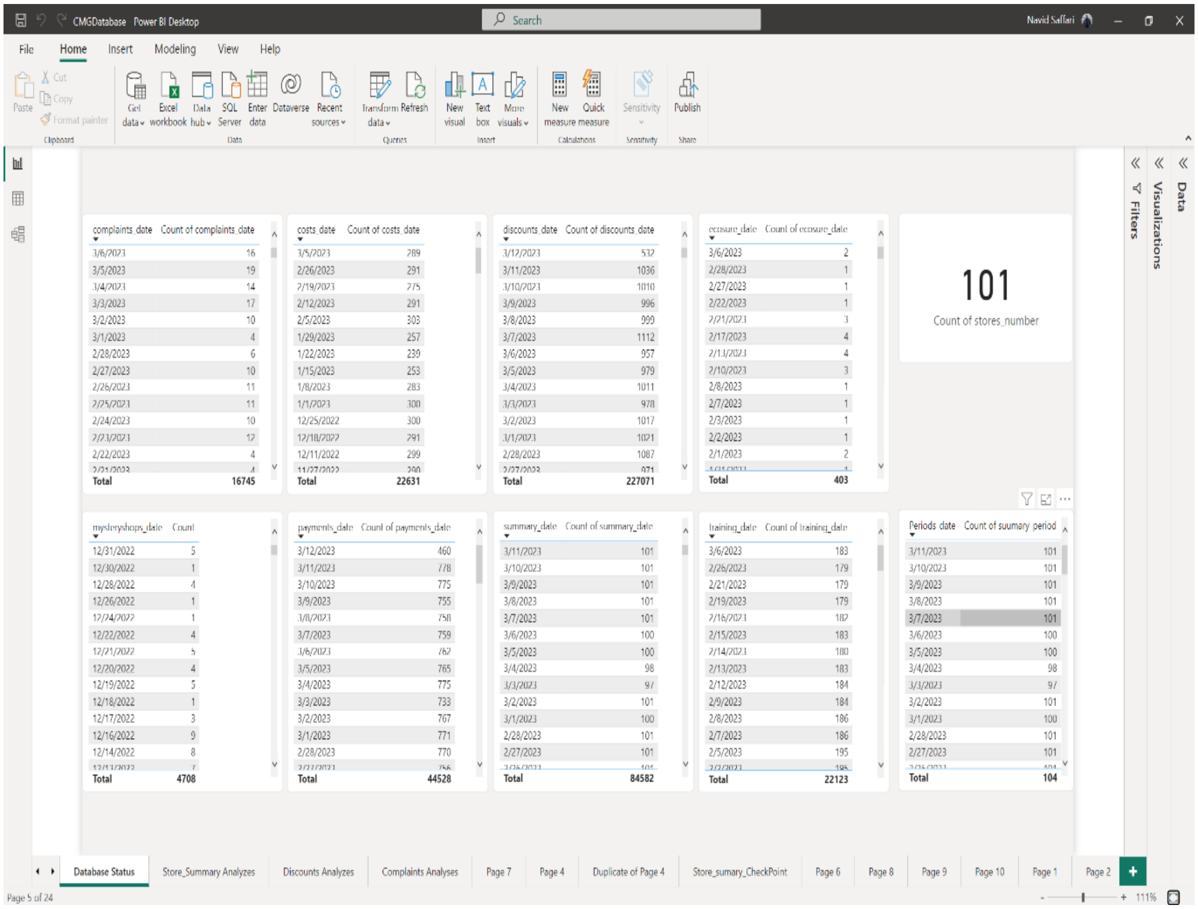


Figure 5 ETL Process Flowchart

3.2 Data Quality and Security:

Guaranteeing the quality and protection of the data has always been a critical aspect of the project. To assure the quality of the data, we implemented several data validation and cleaning techniques, such as checking for missing values, and also a dashboard for each table that we have on the database to monitor each table and make sure we have digested the data to the database in the right way, and also to check if our database is updated with the most recent data or not, because every day we receive data from the client and as our ETL process is automated, we need to see if we have loaded the data of the day to the database or not, as a consequence, we implemented data quality dashboards that is shown in the picture below which provide a visual representation of data quality.



complaints_date	Count of complaints_date
3/6/2023	16
3/5/2023	19
3/4/2023	14
3/3/2023	17
3/2/2023	10
3/1/2023	4
2/28/2023	6
2/27/2023	10
2/26/2023	11
2/25/2023	11
2/24/2023	10
2/23/2023	12
2/22/2023	4
2/21/2023	4
Total	**16745**

costs_date	Count of costs_date
3/5/2023	289
2/26/2023	291
2/19/2023	275
2/12/2023	291
2/5/2023	303
1/29/2023	257
1/22/2023	239
1/15/2023	253
1/8/2023	283
1/1/2023	300
12/25/2022	300
12/18/2022	291
12/11/2022	299
11/27/2022	290
Total	**22631**

discounts_date	Count of discounts_date
3/12/2023	532
3/11/2023	1036
3/10/2023	1010
3/9/2023	996
3/8/2023	999
3/7/2023	1112
3/6/2023	957
3/5/2023	979
3/4/2023	1011
3/3/2023	978
3/2/2023	1017
3/1/2023	1021
2/28/2023	1087
2/27/2023	971
Total	**227071**

ecosure_date	Count of ecosure_date
3/6/2023	2
2/28/2023	1
2/27/2023	1
2/22/2023	1
2/21/2023	3
2/17/2023	4
2/13/2023	4
2/10/2023	3
2/8/2023	1
2/7/2023	1
2/3/2023	1
2/2/2023	1
2/1/2023	2
1/31/2023	1
Total	**403**

mysteryshops_date	Count
12/31/2022	5
12/30/2022	1
12/28/2022	4
12/26/2022	1
12/24/2022	1
12/22/2022	4
12/21/2022	5
12/20/2022	4
12/19/2022	5
12/18/2022	1
12/17/2022	3
12/16/2022	9
12/14/2022	8
12/13/2022	7
Total	**4708**

payments_date	Count of payments_date
3/12/2023	460
3/11/2023	778
3/10/2023	775
3/9/2023	755
3/8/2023	758
3/7/2023	759
3/6/2023	762
3/5/2023	765
3/4/2023	775
3/3/2023	733
3/2/2023	767
3/1/2023	771
2/28/2023	770
2/27/2023	756
Total	**44528**

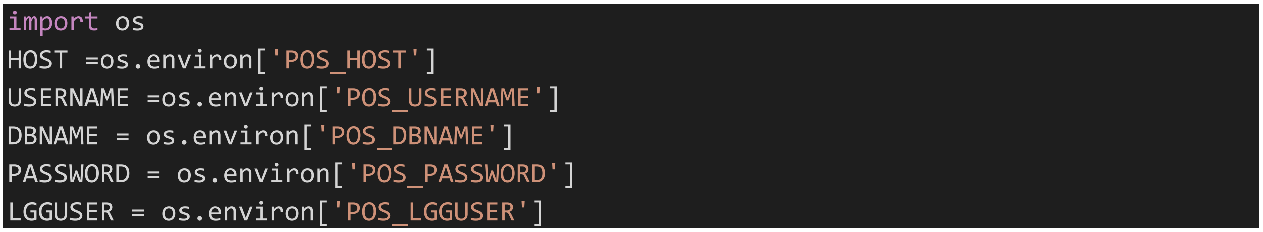
summary_date	Count of summary_date
3/11/2023	101
3/10/2023	101
3/9/2023	101
3/8/2023	101
3/7/2023	101
3/6/2023	100
3/5/2023	100
3/4/2023	98
3/3/2023	97
3/2/2023	101
3/1/2023	100
2/28/2023	101
2/27/2023	101
2/26/2023	101
Total	**84582**

training_date	Count of training_date
3/6/2023	183
2/26/2023	179
2/21/2023	179
2/19/2023	179
2/16/2023	182
2/15/2023	183
2/14/2023	180
2/13/2023	183
2/12/2023	184
2/9/2023	184
2/8/2023	186
2/7/2023	186
2/5/2023	195
2/2/2023	195
Total	**22123**

Periods date	Count of suumary period
3/11/2023	101
3/10/2023	101
3/9/2023	101
3/8/2023	101
3/7/2023	101
3/6/2023	100
3/5/2023	100
3/4/2023	98
3/3/2023	97
3/2/2023	101
3/1/2023	100
2/28/2023	101
2/27/2023	101
2/26/2023	101
Total	**104**

Figure 6 data quality dashboard

In case of any problem in data quality, we will start to investigate where the problem. If the problem is from our department's side, we will try to fix it as soon as possible. If not, we send an email to the person that is responsible on our client's side. We also implemented several security protocols to protect the data, such as encryption and secure data transmission between systems and creating environmental variables to avoid using the username and password for the database or using any directory paths directly on our ETL pipeline, which is a must for data security here is an example of the environment variables:



```
import os
HOST =os.environ['POS_HOST']
USERNAME =os.environ['POS_USERNAME']
DBNAME = os.environ['POS_DBNAME']
PASSWORD = os.environ['POS_PASSWORD']
LGGUSER = os.environ['POS_LGGUSER']
```

Figure 7 Reading Environment Variables

The environment variables are always inserted to the OS environment through a shell script on Linux, which exports the variables and runs the ETL right after exporting the environment variables. The programs run automatically through a crontab on Linux four times a day and in the case of any log, it outputs the log to a log file called posetl.log through the command line that is written in the shell script.

```
export POS_HOST='lggclientsAWS_ServerAddress'
export POS_USERNAME='lgg'
export POS_DBNAME='lggDB'
export POS_PASSWORD='DBpassword'
export POS_LGGUSER='navid.saffari'

/home/zohoapi/pyzoho/bin/python /home/zohoapi/cmgpos/development/CMG_POS/Run.py
```

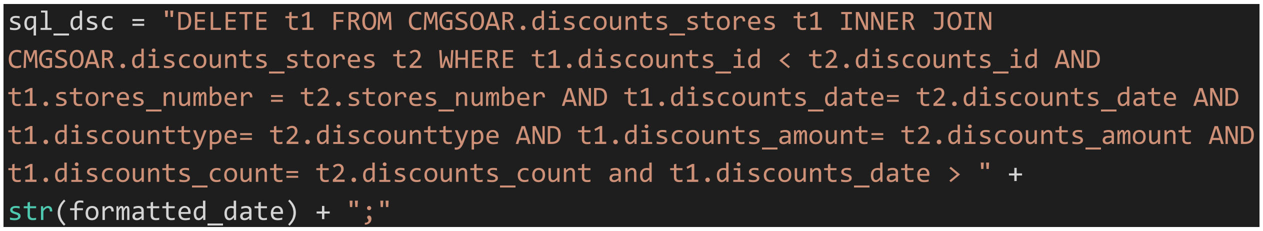
Moreover, the following is the crontab that runs the shell mentioned above script at the specified times of every day and outputs the log to '/tmp/etlpos.log' and by using this part of command '2>&1', in the crontab line, both standard output and standard error will be redirected to the same log file '/tmp/etlpos.log'. This ensures that if the posetl.sh script generates any error messages will also be logged to the same file as regular output, making it easier to troubleshoot any issues that might arise during the execution of the script:

```
30 11 * * * . /home/zohoapi/cmgpos/posetl.sh >> /tmp/etlpos.log 2>&1
15 14 * * * . /home/zohoapi/cmgpos/posetl.sh >> /tmp/etlpos.log 2>&1
45 16 * * * . /home/zohoapi/cmgpos/posetl.sh >> /tmp/etlpos.log 2>&1
15 19 * * * . /home/zohoapi/cmgpos/posetl.sh >> /tmp/etlpos.log 2>&1
15 22 * * * . /home/zohoapi/cmgpos/posetl.sh >> /tmp/etlpos.log 2>&1
```

Furthermore, we change the password of the email that we receive the data from the client every month and the password of the company's GitHub, one of our work's most critical sources. We also made sure to follow best practices for data security. Some examples are, using strong passwords and regularly observing and checking logs of our data pipelines, using a cloud solution (which in our case is AWS), and creating access hierarchies on different data and pipelines. Although, there is always much more room to work and improve data security and data quality.

3.3 Challenges and Solutions:

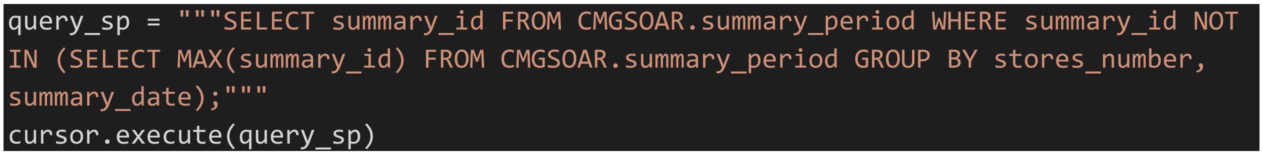
As with any project, we faced several challenges during the experience. One of the main challenges was handling large amounts of data daily. We implemented several solutions to overcome this challenge, such as using efficient data processing techniques like parallel processing and optimizing the pipelines for performance. For example, as we might receive duplicate data, we always need to check on duplicates and remove them. In the beginning, we were using a query that was taking so much time, which is as below:



```
sql_dsc = "DELETE t1 FROM CMGSOAR.discounts_stores t1 INNER JOIN
CMGSOAR.discounts_stores t2 WHERE t1.discounts_id < t2.discounts_id AND
t1.stores_number = t2.stores_number AND t1.discounts_date= t2.discounts_date AND
t1.discounttype= t2.discounttype AND t1.discounts_amount= t2.discounts_amount AND
t1.discounts_count= t2.discounts_count and t1.discounts_date > " +
str(formatted_date) + ";"
```

Figure 8 Slow Query for Removing Duplicates

As the query above was very time-consuming, we tried to define another query that is faster and consumes fewer resources and time, which is as below:



```
query_sp = """SELECT summary_id FROM CMGSOAR.summary_period WHERE summary_id NOT
IN (SELECT MAX(summary_id) FROM CMGSOAR.summary_period GROUP BY stores_number,
summary_date);"""
cursor.execute(query_sp)
```

Figure 9 Faster Query for Removing Duplicates

Another challenge was working with a team of developers with distinct background levels. To overcome this challenge, we communicated effectively with our team fellows and provided clear documentation and instructions for the pipeline. For example, by breaking down the work into different tasks and assigning them to the best person fitting for that task and using ZohoProject for task assignments by using that, we were also able to monitor the performance of each person on their task and the hours they dedicate to their that. Also, it is an appropriate tool that the company can use to consider employee promotions.

3.4 Lessons learned and future work.

Throughout the internship, we learned several worthwhile lessons about data engineering. One of the most powerful lessons is data quality and security. We also know the importance of efficiency and performance when working with large data portions. For future work, we would like to explore more cutting-edge data processing approaches and technologies, such as Docker and Airflow. Speaking about technical lessons that have been learned, we learn learned about how to process the data and get hands-on pandas in detail, program efficiently in Python, and how to deal with the database using Python, which is sometimes a kind of frustrating work that nobody has to underestimate considering that there are always many errors connecting to the database. We always have to consider the safest way of connecting to the database. In terms of security, it is recommended to use docker, which is a very friendly and efficient technology in terms of being replicable and isolated, each project can be in one or many docker containers (A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries, and settings) and it is isolated which means none of the containers have access to each other (can have access through creating networks and assigning containers to the same port), in other words, docker containers are a kind of virtual machines but very lightweight and more efficient. Another tool that is highly recommended to use is Airflow because it gives us much flexibility. With Airflow, we can create, schedule, and monitor workflows and automate and create alerts if any task fails, which can be a massive advantage for any company.

Below is the workflow of the Digital Transformation team from LGG Advisors which indicates that the team, receives the data files which are mostly CSV and Excel files, and loads them to a Python program for transformation of the data to make it suitable for the database and make multiple calculations required for the dashboard and choose needed features and then export the final data as CSV, then load the cleaned and transformed data which is in the new csv files to a Python file.

Again, export the data to the database, and as mentioned, there are queries in the database that check the data quality to determine if all the necessary and required data for each day is there. Connecting PowerBI to our database, creating the necessary measurements, metrics, and KPIs, and analyzing the data, and in the end, the person responsible for the team presenting the dashboard to the client will make a presentation weekly.

4 CONCLUSION:

In conclusion, the data engineering internship at LGG Advisors provided a comprehensive and hands-on experience in the field of data engineering. The project involved creating and maintaining ETL pipelines for financial data, which required skills in Python programming, data manipulation, and database management using MySQL. The project's critical challenges included ensuring data quality and security, handling large amounts of data, and working with a team of developers with varying skill levels. To address these challenges, we implemented several solutions, such as data validation and cleaning techniques, parallel processing, and effective communication and documentation. We also utilized tools and technologies such as Pandas, NumPy, MySQL.connector, and GitHub for efficient data processing and version control. Throughout the internship, we learned the importance of data quality and security and the significance of efficiency and performance when working with large datasets. The project allowed us to enhance our technical skills and collaborate with a team to accomplish project goals. The experience was valuable in understanding the practical implications of data engineering in a real-world setting. For future work, we want to explore different optimization techniques for data processing and pipeline performance. We also aim to continue learning about the latest tools and technologies in the field of data engineering and implement them to improve our project's effectiveness. Overall, the internship was an enriching experience that provided a solid foundation for a career in data engineering.

REFERENCES

Al-Shammari, M. M., & Alwan, A. A. (2018, August 20). Disaster Recovery and Business Continuity for Database Services in Multi-Cloud. 1st International Conference on Computer Applications and Information Security, ICCAIS 2018. https://doi.org/10.1109/CAIS.2018.8442005

Barrett, P., Hunter, J., Miller, J. T., Hsu, J. C., & Greenfield, P. (2005, December). matplotlib--A Portable Python Plotting Package. In Astronomical data analysis software and systems XIV (Vol. 347, p. 91).

Bertino, E. (2016). Data Security and Privacy: Concepts, Approaches, and Research Directions. Proceedings - International Computer Software and Applications Conference, 1, 400–407. https://doi.org/10.1109/COMPSAC.2016.89

Biswas, N., Sarkar, A., & Mondal, K. C. (2019). Empirical analysis of programmable ETL tools. In Computational Intelligence, Communications, and Business Analytics: Second International Conference, CICBA 2018, Kalyani, India, July 27-28, 2018, Revised Selected Papers, Part II (Vol. 2, pp. 267-277). Springer Singapore.

Byun, J. W., Bertino, E., & Li, N. (2005). Purpose based access control of complex data for privacy protection. In Proceedings of the 10th ACM symposium on Access control models and technologies (pp. 102-110). June 2005.

Caetano, T. V., & Costa, C. J. (2014). Data Warehousing num contexto de Sistemas Integrados. In Atas da Conferência da Associação Portuguesa de Sistemas de Informação (Vol. 12, pp. 186-199). March 2014.

Costa, C. J., & Aparicio, J. T. (2020). POST-DS: A methodology to boost data science. In 2020 15th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE. https://doi.org/10.23919/CISTI49556.2020.9140932

Crickard, P. (2020). Data Engineering with Python. Packt Publishing.

Davis, R. (1978). The data encryption standard in perspective. IEEE Communications Society Magazine, 16(6), 5-9. https://doi.org/10.1109/MCOM.1978.1089771

Drosio, S., & Stanek, S. (2016). The Big Data concept as a contributor of added value to crisis decision support systems. Journal of Decision Systems, 25(sup1), 228-239. https://doi.org/10.1080/12460125.2016.1187404

Ehrlinger, L., & Wöß, W. (2022). A survey of data quality measurement and monitoring tools. Frontiers in Big Data, 5, 28. https://doi.org/10.3389/fdata.2022.850611

Gokhale, K. M., Chandan, J. S., Toulis, K., Gkoutos, G., Tino, P., & Nirantharakumar, K. (2021). Data extraction for epidemiological research (DExtER): a novel tool for automated clinical epidemiology studies. European Journal of Epidemiology, 36(2), 165–178. https://doi.org/10.1007/s10654-020-00677-6

Gray, J., & Shenoy, P. (n.d.). Rules of Thumb in Data Engineering. www.tpc.org

Heinrich, B., Hristova, D., Klier, M., Schiller, A., & Szubartowicz, M. (2018). Requirements for data quality metrics. Journal of Data and Information Quality, 9(2), 1-32.

Kimball, R., & Ross, M. (2013). The data warehouse ETL toolkit: The Definitive Guide to Dimensional Modeling. Wiley & Sons.

Lomet, D. B., Gravano, L., Levy, A., & Weikum, G. (n.d.). Editorial Board Editor-in-Chief Associate Editors. http://list.research.microsoft.com/scripts/lyris.pl?enter=debull

Masoumzadeh, A., Joshi, J.B.D. (2008). PuRBAC: Purpose-Aware Role-Based Access Control. In: Meersman, R., Tari, Z. (eds) On the Move to Meaningful Internet Systems: OTM 2008. OTM 2008. Lecture Notes in Computer Science, vol 5332. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88873-4_12

O'Donovan, P., Leahy, K., Bruton, K., & O'Sullivan, D. T. J. (2015). An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. Journal of Big Data, 2(1). https://doi.org/10.1186/s40537-015-0034-z

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830..

Petersen, P., Stage, H., Langner, J., Ries, L., Rigoll, P., Hohl, C. P., & Sax, E. (2022). Towards a Data Engineering Process in Data-Driven Systems Engineering. In 2022 IEEE International Symposium on Systems Engineering (ISSE) (pp. 1-8). October 2022. https://doi.org/10.1109/ISSE54508.2022.10005441

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. Communications of the ACM, 45(4), 211-218.

Seenivasan, D. (2023). ETL (Extract, Transform, Load) Best Practices. International Journal of Computer Trends and Technology, 71(1), 40-44. https://doi.org/10.14445/22312803/IJCTT-V71I1P106

Simitsis, A., Vassiliadis, P., & Sellis, T. (2005, April). Optimizing ETL processes in data warehouses. In 21st International Conference on Data Engineering (ICDE'05) (pp. 564-575). IEEE.

Souibgui, M., Atigui, F., Zammali, S., Cherfi, S., & Yahia, S. B. (2019). Data quality in ETL process: A preliminary study. Procedia Computer Science, 159, 676-687. https://doi.org/10.1016/j.procs.2019.09.223ï

Suguna, S., & Suhasini, A. (2014). Overview of data backup and disaster recovery in cloud. In International Conference on Information Communication and Embedded Systems (ICICES2014) (pp. 1-7). February 2014

Vassiliadis, P., & Simitsis, A. (2009). Extraction, Transformation, and Loading. In Encyclopedia of Database Systems (pp. 1-6).

Tank, D. M. (2012). Reducing ETL Load Times by a New Data Integration Approach for Real-time Business Intelligence. In International Journal of Engineering Innovation & Research (Vol. 1, Issue 2).

Waskom, M. L. (2021). Seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021.