NOVA IMS Information Management School

# MGI

Master Degree Program in
**Data Science and Advanced Analytics**

# ENHANCING DATA EXTRACTION AND TRANSFORMATION FOR BUSINESS INTELLIGENCE

## Integrating Database Sources with Cloud Storage for Streamlined Reporting

Denzel Basso

Project Work

Presented as partial requirement for obtaining the Master 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

# ENHANCING DATA EXTRACTION AND TRANSFORMATION FOR BUSINESS INTELLIGENCE

Integrating Database Sources with Cloud Storage for Streamlined Reporting

by

Denzel Basso

Project Work presented as a partial requirement for obtaining the Master’s degree in Data Science and Advanced Analytics with a specialization in Business Analytics

**Supervisor:** Bruno Jardim, PhD, NOVA Information Management School

June 2024

# STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

*Denzel Basso*

*Milan, 1st of June 2024*

# ABSTRACT

This thesis explores the field of data management, focusing on migrating from on-premises SQL solutions to cloud-based Data Lakehouse (DLH) structures using Microsoft Azure. The main challenge addressed is the inefficiency of traditional data storage systems in handling the exponential growth of big data, in this case in an international manufacturing company. The objectives are to reduce data storage costs, increase flexibility for future expansions, and upgrade the current data structure.

This study will be based on a case study methodology to address these challenges, focusing on an international logistics company operating primarily in Italy, the USA, and Mexico. The migration process involves moving the current store procedures to the new environment, modifying tabular structures for business intelligence integration, and creating the supporting code for managing the data. A team of eight, including senior managers, consultants, and analysts from Deloitte's Digital Control branch, participated in this project.

Key results indicate significant cost reduction and enhanced flexibility in data storage architecture. The migration to a cloud-based DLH structure allows for efficient handling of high-volume, high-velocity, and high-variety data. This transformation also facilitates better scalability, reliability, and integration with business intelligence tools, addressing the inefficiencies of conventional systems.

The major contribution of this work lies in demonstrating the practical benefits of adopting a DLH structure in a real-world logistics scenario. It provides a comprehensive framework for other organizations looking to move from traditional data management systems to more efficient and scalable cloud-based solutions. This research not only contributes to the understanding of DLH implementation but also sets the stage for future studies on cloud-based data management innovations.

By bridging the gap between theoretical knowledge and practical application, this thesis aims to advance the field of data management, offering valuable insights for both academia and industry.

# KEYWORDS

Data Migration; Cloud Computing; Data Lakehouse; Business Intelligence; Integration; Cloud-based system; Azure Synapse Analytics; SQL.

## Sustainable Development Goals (SGD):

# INDEX


1. Introduction .......... 1
   1.1. General Context .......... 1
   1.2. Organizational Context .......... 1
   1.3. Thesis Objectives .......... 2
   1.4. Structure of the Thesis .......... 3
2. Literature review .......... 4
   2.1. Business Intelligence on Cloud .......... 4
   2.2. Pros and Cons of Cloud Computing with Microsoft Azure .......... 5
      2.2.1. Advantages of Cloud Computing .......... 5
      2.2.2. Pros and Cons of Microsoft Azure .......... 5
      2.2.3. Challenges and Considerations .......... 7
   2.3. Data Lakehouse Structure: Bridging Gaps in Spatial Big Data Management .......... 8
      2.3.1. Data Warehouses (DWHs) .......... 8
      2.3.2. Data Lakes (DLs) .......... 9
      2.3.3. Data Lakehouse (DLH) .......... 10
      2.3.4. Comparison of the three models .......... 11
   2.4. Case Studies: Migration Strategies and Business Cases .......... 11
      2.4.1. General Guidelines for Data Migration .......... 11
      2.4.2. Business Cases: Migration Experiences .......... 12
3. Methodology .......... 15
   3.1. Current State Analysis .......... 16
      3.1.1. Scalability of the infrastructure .......... 16
      3.1.2. Maintainability of the infrastructure .......... 16
      3.1.3. Data Integration .......... 17
      3.1.4. Data Governance .......... 17
      3.1.5. Data Quality .......... 17
      3.1.6. Business Intelligence (frontend) .......... 17
   3.2. Scenarios proposed .......... 18
      3.2.1. Scenario 1: Managed SQL Database .......... 18
      3.2.2. Scenario 2: Databricks .......... 18
      3.2.3. Scenario 3: Serverless Services .......... 18
      3.2.4. Scenarios comparison .......... 19
4. EMPIRICAL STUDY .......... 21

4.1. Analysis and Transportation of Stored Procedures....23
4.1.1. Challenges from an on-premises database to a serverless one....24
4.2. Tabular and PowerBI Layer for Reporting....26
5. Results & Discussions....30
5.1. Impact Analysis....30
5.1.1. Converting Stored Procedures in Azure Synapse....30
5.1.2. Performance Optimization and Time Efficiency Issues....31
5.1.3. Suboptimal Tabular Structure and Integration Challenges....31
5.1.4. Transition Challenges and Extended Data Processing Time....31
5.2. Discussion....32
6. Conclusion....34
7. References....35

# LIST OF FIGURES

Figure 2.1 - Public cloud PaaS end-user spending worldwide from 2015 to 2024 ...........5
Figure 2.2 - Data Warehouse Architecture ........................................................................8
Figure 2.3 - Data flow in a pond architecture ....................................................................9
Figure 2.4 - Data Lakehouse architecture ..........................................................................9
Figure 3.1 - Kimball Lifecycle Diagram from Landutama & Chowanda (2023).................12
Figure 4.1 - Managed SQL Database Architecture ..........................................................18
Figure 4.2 - Frame for the block in the Stored Procedure ..............................................22
Figure 4.3.A - New code for INSERT INTO ......................................................................22
Figure 4.3.B - Elimination of INTO a temporal table .......................................................22
Figure 4.4.A - Original form for UPDATE .........................................................................23
Figure 4.4.B - The new version for UPDATE ....................................................................23
Figure 4.5 - Tabular of AR-SD ..........................................................................................24
Figure 4.6 - Merge two data sources ...............................................................................25
Figure 4.7 - Monitoring report .........................................................................................26

# LIST OF TABLES

Table 2.1 - Advantages of Microsoft Azure ........................................................................5
Table 2.2 - Disadvantages of Microsoft Azure ..................................................................6
Table 2.3 - Key steps in End-to-end Data migration ......................................................10
Table 3.1 - Architecture’s scenarios ...................................................................................17
Table 5.1 - Summary of Impact Analysis ..........................................................................17

# LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| **DLH** | Data Lakehouse |
| **ACID** | Atomicity, Consistency, Isolation, Durability |
| **API** | Application Programming Interface |
| **AR-SD** | Accounts Receivable – Sales Distribution |
| **AWS** | Amazon Web Services |
| **Azure** | Microsoft Azure |
| **BI** | Business Intelligence |
| **Blob** | Binary Large Object |
| **CDS** | Core Data Service |
| **CO** | Controlling |
| **CS** | Customer Service |
| **DBMS** | Database Management System |
| **DL** | Data Lake |
| **DLH** | Data Lakehouse |
| **DWH** | Data Warehouse |
| **ECC** | SAP ERP Central Component |
| **ERP** | Enterprise Resource Planning |
| **ETL** | Extract, Transform, Load |
| **IT** | Information Technology |
| **IoT** | Internet of Things |
| **KPI** | Key Performance Indicator |
| **MPI** | Message-Passing Interface |
| **OLAP** | Online Analytical Processing |
| **PaaS** | Platform as a Service |
| **REST** | Representational State Transfer |
| **S4** | SAP S4/HANA |

| | |
|---|---|
| **SAP** | Systems, Applications, and Products in Data Processing |
| **SD** | Sales Distribution |
| **SD-CO** | Sales Distribution – Controlling |
| **SME** | Small and Medium-sized Enterprise |
| **SP** | Stored Procedure |
| **SQL** | Structured Query Language |
| **WBS** | Work Breakdown Structure |

# 1. INTRODUCTION

## 1.1. GENERAL CONTEXT

Since the beginning of the use of data, the modality, and the structure in which it was stored played a crucial role in data management. Making the right storage decision regarding how to store them can positively impact the speed of delivery insights with a tremendous effect on the decision-making process of the organization (Marr, 2023). The creation of new technologies and the exponential growth of the amount of data that the organizations needed to process made the initial storage solutions obsolete in a relatively short time. An example of this phenomenon is the transition from conventional data management systems to NoSQL technology, driven by the need to meet the storage demands of big data (Siddiqa et al., 2017).

The rise of cloud storage solutions took an important role in the never-ending battle between the amount of data generated and the capability of analyzing and extracting valuable information from it. As cited by Curry et al. (2016): "As cloud computing grows in popularity, its influence on big data grows as well". In various data acquisition scenarios, the underlying assumption involves dealing with high-volume, high-velocity, high-variety, yet low-value data. Consequently, there arises a significant requirement for adaptable and time-efficient algorithms for gathering, filtering, and cleaning. The purpose is to selectively process only the high-value segments of the data during the subsequent data warehouse analysis (Curry et al., 2016).

This internship report will focus on the data migration from an on-premises SQL solution to an on-cloud Data Lakehouse (DLH) structure using Microsoft Azure and its other complementary programs. The final solution will be implemented in several countries, more specifically Italy, the U.S.A., and Mexico, in an international company that operates in the logistics sector. The scope is to reduce the cost-per-data of the client and to increase the flexibility of its data storage architecture for future extensions in other countries in which it's present.

This project is the first phase of a multi-year vision in which the client will extend the cloud solution to all its factories and was developed by a team of 8 from the Digital Control branch of Deloitte. The team consisted of 1 senior manager and 1 manager, 2 senior consultants and 2 consultants, and two analysts. The focus of the work will be on the translation of the store procedure of the client in the new environment language, the modification of the tabular[1] that will be implemented to connect to the business intelligence solutions and the creation of supporting code for information retrieval.

## 1.2. ORGANIZATIONAL CONTEXT

The internship on which this thesis will be based was offered by Deloitte, a worldwide famous company with a strong presence in various industries. Deloitte operates as a multinational consultancy firm that provides a wide range of services, including tax, advisory, consulting, and audit. The company was founded in 1845 and has since established itself as a leader in the professional services industry

[1] Tabular models in Analysis Services are databases that run in-memory or in DirectQuery mode, connecting to data from back-end relational data sources ("Tabular modeling overview," 2022)

(Deloitte, 2023a). Currently, the number of people working in it is around 455.000, presence in more than 150 countries (Deloitte, 2023b).

Deloitte's mission revolves around making an impact that matters and helping clients address their most complex challenges. With a commitment to excellence and innovation, Deloitte aims to assist organizations in navigating the evolving business landscape and staying ahead of industry trends. The company's headquarters are situated in London, and it operates across multiple locations worldwide.

Recognized as one of the Big Four firms for consultancy, Deloitte has consistently been acknowledged for its expertise and contributions to various domains. In 2023, it was recognized as one of the best management consulting firms in the world, reaffirming the high value of its project and the overall knowledge it can provide to its clients (Peachman, 2023).

Deloitte services are divided into 6 main areas, which are:

- Tax Services
- Financial Advisory
- Risk Advisory
- Consulting
- Audit and Assurance

The project will be held by the Digital Controls, Data & Analytics team, a specialized force in the Risk Advisory team that helps companies reduce the new risks that can appear while using new technologies (Deloitte, 2023c). The major services offered are:

- Technology Advisory
- System Integration
- Design and development of custom solutions

This internship is a great opportunity offered by Deloitte to be part of a global team of experts and to learn from their knowledge of how to deal with complex tasks. The valuable insights that will be taken from applying cutting-edge technologies will be an important addition to the quality of results achieved during the entire duration of the internship.

## 1.3. Thesis Objectives

The primary challenge of this research is to evaluate the benefits and effectiveness of migrating from an on-premises SQL solution to a cloud-based Data Lakehouse (DLH) structure using Microsoft Azure within an international manufacturing company. This study builds on previous research by authors such as Ellison et al. (2018) and others who have examined the strategic advantages of adopting cloud-based data management solutions in enhancing business intelligence and decision-making capabilities. The objectives of the thesis are the following:

- To assess the current state of the on-premises SQL solution and identify areas for improvement.

- To design and implement a cloud-based Data Lakehouse (DLH) structure using Microsoft Azure.
- To evaluate the impact of the DLH migration on data management efficiency, cost reduction, and scalability.
- To analyze the challenges and solutions encountered during the migration process.
- To provide recommendations for other organizations considering a similar transition.

## 1.4. Structure of the Thesis

This project is structured into six main chapters that guide the research process, following a logical structure. These chapters, organized in a logical order, form the base of this thesis. While the first chapter is focused on the introduction of the thesis, the second one provides a detailed review of existing research on cloud computing, data migration, and the application of Data Lakehouse (DLH) frameworks using Microsoft Azure within an organizational setting. Chapter three presents the specifics of transitioning from on-premises SQL systems to cloud-based Data Lakehouse structures. This chapter outlines the benefits and challenges of this migration, describing the methods used to analyse the current state of the client's Business Intelligence (BI) infrastructure, the proposed migration scenarios, and the new system's architectural design. It also includes an overview of the qualitative research methodology applied in this project. The following chapters present the findings of the research, including data analysis and the conclusions drawn from the study. These results are further examined and discussed throughout the subsequent chapters.

In today's fast-changing business environment, a significant shift has occurred towards making data-driven decisions, moving away from reliance on intuition. This project focuses on data management, particularly the transition to a cloud-based Data Lakehouse (DLH) structure, which is central to this transformative process. This approach emphasizes the importance of using evidence-based methods to generate actionable insights from data analysis, fostering effective collaboration between organizational leaders and IT experts. Such a partnership is crucial for effectively managing modern data challenges and achieving long-term business success and a competitive advantage.

# 2. LITERATURE REVIEW

The convergence of Business Intelligence (BI) and cloud computing has reshaped the landscape of data analytics and decision-making. This transformative shift, marked by the advent of cloud technologies, brings forth a new era in which BI on the cloud is not only a reality but a pivotal factor in organizational success. As we delve into this literature review, we navigate through the multifaceted aspects of this amalgamation, exploring the opportunities, challenges, and implications for modern businesses. From the scalability and flexibility offered by cloud platforms to the intricacies of spatial data management and the diverse realm of open-source cloud solutions, this review provides a comprehensive understanding of the dynamic interplay between BI and the cloud.

## 2.1. BUSINESS INTELLIGENCE ON CLOUD

Business Intelligence (BI) has witnessed a transformative shift with the advent of cloud computing, redefining the dynamics of data analytics and decision-making processes (Alsufyani & Chang, 2015). Cloud computing revolutionizes BI economics by providing on-demand access to hardware, networking, security, and software, utilizing a pay-as-you-go model (Alsufyani & Chang, 2015). This shift has presented both opportunities and challenges.

The integration of BI with cloud computing has become a key success factor for organizations globally (ElMalah & Nasr, 2019). The flexibility and scalability offered by cloud platforms such as Microsoft Azure, Google Cloud, Amazon AWS, and IBM Bluemix have led to a substantial adoption rate (ElMalah & Nasr, 2019). The three-layer architecture, encompassing the Data Layer, Logic Layer, and Access Layer, forms the foundation for understanding the potential of Cloud BI (ElMalah & Nasr, 2019). This architecture improves the analysis of data, has an user-friendly access, and, most importantly, facilitate the storage of both structured and unstructured data.

However, the adoption of Cloud BI is not without challenges. Concerns about legislation, privacy, and security arise due to the inherent nature of cloud-based technologies, where the principle of "everyone accesses everyone" may compromise data privacy (Alsufyani & Chang, 2015). Nevertheless, cloud computing's extensive storage capacity, flexibility, and multiple layers of protection enhance risk and incident management (Alsufyani & Chang, 2015).

Cloud BI offers several benefits, including fast deployment, reduced costs, reliability, and flexibility (ElMalah & Nasr, 2019). The pay-as-you-go model ensures cost-effectiveness, and the multi-tenancy nature of Cloud BI allows for resource optimization (ElMalah & Nasr, 2019). Furthermore, free automated software upgrades and maintenance contribute to the overall appeal of Cloud BI solutions (ElMalah & Nasr, 2019).

The integration of BI with cloud computing presents various application scenarios. From Cloud-based front-end tools to end-to-end BI stacks, organizations have options that range from tool replacement scenarios to comprehensive cloud-based solutions (Ereth & Baars, 2015). Cloud BI can also facilitate data sharing in a community cloud, enhancing cross-company collaboration (Ereth & Baars, 2015).

Data integration, a critical aspect of BI frameworks, involves various components such as data sources, data integration, Online Analytical Processing (OLAP), extract, transform, load (ETL), and visualization tools (Chairungruang et al., 2022). ETL processes are particularly essential for integrating data from different sources, ensuring data homogeneity, cleaning, and loading into data warehouses (Rodzi et al., 2015).

In conclusion, the intersection of Business Intelligence and cloud computing is reshaping decision-making processes, offering scalability, efficiency, and cost-effectiveness. Despite challenges, the benefits and potential applications highlight the pivotal role of Business Intelligence on the Cloud in shaping the future of data analytics.

## 2.2. Pros and Cons of Cloud Computing with Microsoft Azure

Cloud computing, a transformative force in modern IT, reshapes how organizations manage and analyze data. This section delves into the advantages and challenges of cloud computing, focusing on Microsoft Azure.

### 2.2.1. Advantages of Cloud Computing

In general, the main advantages of Cloud Computing are the following:

- Scalability and Flexibility: Cloud platforms, notably Microsoft Azure, provide unparalleled scalability. Azure's elasticity allows businesses to ensure the best performance possible by adjusting dynamically the level of resources based on the current demand, without unnecessary overhead (Fernandes et al., 2023).
- Cost Efficiency: Cloud computing's pay-as-you-go model, exemplified by Microsoft Azure, transforms capital expenses into operational costs. This approach offers a cost-effective solution that is particularly beneficial for small and medium-sized enterprises (SMEs) due to the relatively limitation of their budget compared to bigger firms (da Costa & Da Cruz, 2012).
- Accessibility and Remote Collaboration: Cloud services enhance remote access and collaboration. Microsoft Azure's cloud-based infrastructure facilitates seamless teamwork and information sharing (da Costa & Da Cruz, 2012).

### 2.2.2. Pros and Cons of Microsoft Azure

Microsoft offers Windows Azure as a Platform as a Service (PaaS). It operates like an operating system, offering services that developers can access to build applications and store data. Unlike being installed on a single server, it spans across several hundred servers (da Costa & Da Cruz, 2012). Currently is one of the most popular Cloud providers, second only to AWS (Gupta et al., 2021). The general market for PaaS public cloud has had a healthy growth since the early 2010s, with an estimated growth for 2024 of 170 billion U.S. dollars, with an increase of 22,6% compared to the previous year, as seen in Figure 2.1.

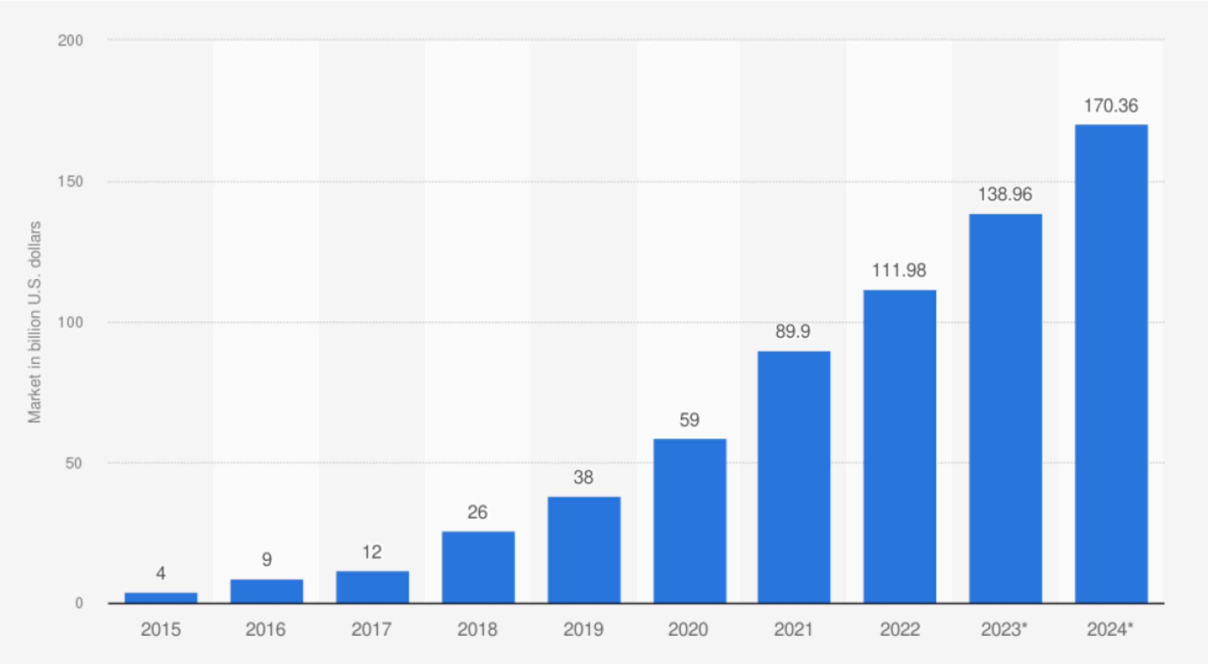

Figure 2.1 – Public cloud PaaS end-user spending worldwide from 2015 to 2024 (Vailshery, 2023)

Microsoft Azure, in comparison with other Cloud service providers, has the following benefits (Yevge et al., 2022) (Gupta et al., 2021):

Table 2.1 – Advantages of Microsoft Azure

| **Benefit** | **Description** |
|---|---|
| Flexible Pricing | Tailor your costs according to your needs |
| Security in Services | Microsoft's shielded platform ensures robust security |
| Scalability and Precision | Scale your solutions with accuracy and reliability |
| Reliable Uptime | Maintain high availability for uninterrupted operations |
| Economical Solutions | Cost-effective options compared to alternatives |
| Leading Service Provider | A frontrunner among cloud service providers |
| Seamless Integration | Harmonize effortlessly with Microsoft tools and devices |
| Unified Cloud Support | Comprehensive backing for both public and private cloud setups |
| Open to Open Source | Embraces and supports open-source technologies |
| Widespread Accessibility | Available widely for easy access |
| Tool Harmony | Works smoothly with various Microsoft tools and software |
| Discounted Contracts | Enjoy cost savings with service contract discounts |

| Benefit | Description |
|---|---|
| Multilingual Support | Built-in apps cater to multiple languages, including Node.js, Python, .NET, PHP, and JavaScript |
| Budget-Friendly Rates | Features reasonably priced on-demand services |
| Downtime Mitigation | Incorporates redundancies for minimal disruptions |

To better understand the limitations of Azure, it's important to also identify some of the challenges in this platform. The advantages of Microsoft Azure are the following (Yevge et al., 2022) (Gupta et al., 2021):

Table 2.2 – Disadvantages of Microsoft Azure

| Limitation | Description |
|---|---|
| Documentation Issues | Challenges related to the adequacy of documentation |
| Limited Management Tools | Insufficient tools for effective management |
| Steep Learning Curve | Perceived difficulty in learning and using the platform |
| High Cost | Considered expensive, including data transfer expenses |
| Platform Expertise Required | Demands a certain level of expertise for effective utilization |
| Data Management Shortcomings | Lack of comprehensive data management capabilities |
| Network Management Challenges | Reports of difficulties in managing network aspects |
| Perceived Difficulty in Learning | Some users find it more challenging compared to alternatives |
| Less Professional Design | The design may feel less polished or professional |
| Technical Support Concerns | Users have reported issues with technical support |

### 2.2.3. Challenges and Considerations

There are 3 main challenges presented for the future of Cloud Computing (da Costa & Da Cruz, 2012):

- Security and Privacy Concerns: Despite robust security measures, organizations, including those using Microsoft Azure, must address ongoing concerns about data privacy and security.
- Vendor Lock-In: Selecting a specific cloud provider, such as Microsoft Azure, may lead to vendor lock-in. Organizations must carefully weigh the benefits against potential challenges, considering future flexibility (da Costa & Da Cruz, 2012).

- Technical Support and Expertise: Effective cloud utilization requires technical expertise. Organizations transitioning to Azure must ensure their teams possess the necessary skills or invest in training to optimize platform benefits (da Costa & Da Cruz, 2012).

Understanding these dynamics is crucial for businesses navigating cloud adoption. The next section explores the evolving concept of Data Lakehouse structure and its implications for modern data architectures.

## 2.3. Data Lakehouse Structure: Bridging Gaps in Spatial Big Data Management

Spatial Big Data management has witnessed a paradigm shift with the advent of Big Data technologies, necessitating new approaches to storage and spatial representation. Traditional methods, including Data Warehouses (DWH) and Data Lakes (DLs), have faced challenges in addressing the growing complexities of spatial data.

The emergence of the Data Lakehouse (DLH), as discussed in Errami et al. (2023) and Harby and Zulkernine (2023), represents a comprehensive solution that combines the strengths of both DWHs and DLs while mitigating their respective limitations.

### 2.3.1. Data Warehouses (DWHs)

DWs are designed for analytical processing, offering a centralized, consistent view of processed data. Known for their structured nature and high integrity, DWHs are suitable for structured data analytics and business intelligence. However, their limitations include high costs, time-consuming configurations, and challenges in handling unstructured data types. As Harby and Zulkernine (2023) note, DWHs guarantee data integrity through ACID compliance but lack the flexibility to handle the diversity and scale of modern Big Data.

The typical DWH architecture consists of 4 main structures: Source Layer, Data Staging (ETL), DW layer, and Analysis. Figure 2.2 by Blažić et al., (2017), provides a comprehensive overview of a DWH with the presence of the Data Mart, a “Subset of a data warehouse focused on a particular line of business” (IBM, 2023).

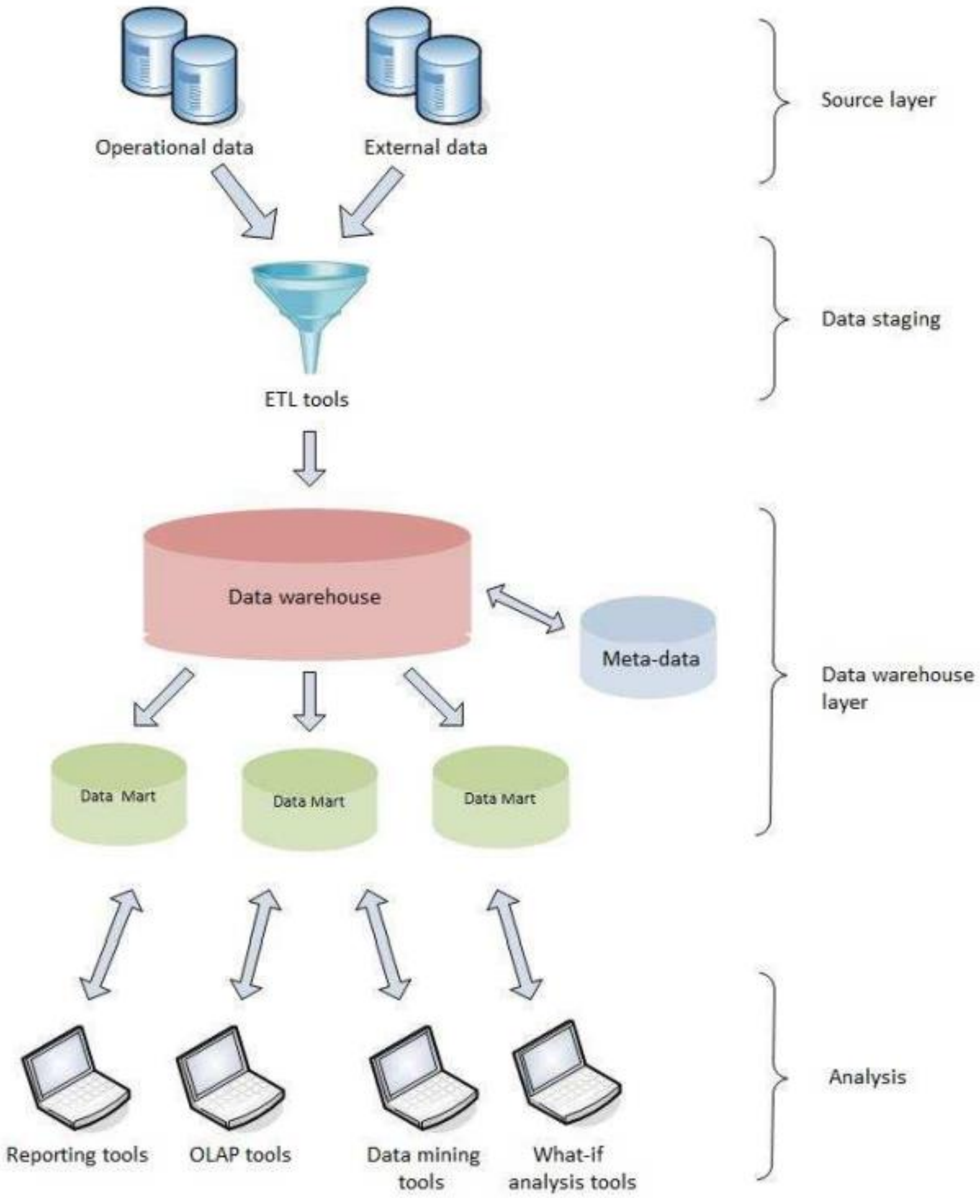


Figure 2.2 – Data Warehouse architecture (Blažić et al., 2017)

### 2.3.2. Data Lakes (DLs)

DLs, on the other hand, provide low-cost storage for raw, unstructured data. They support diverse data types and are scalable, making them an attractive solution for managing large volumes of data. However, DLs face challenges related to consistency, governance, and the potential transformation of data swamps due to the lack of schema enforcement. As Nambiar and Mundra (2022) highlight, DLs follow a schema-on-read architecture, enabling agile storage but potentially leading to difficulties in analysis without proper tools for data cataloging. There are different structures of DL, like the Two Layer Architecture presented by Hlupić et al. (2022), or the Data Pond Architecture, created by Bill Inmon (2016) and represented visually in Figure 2.3 by Sawadogo & Darmont (2021).

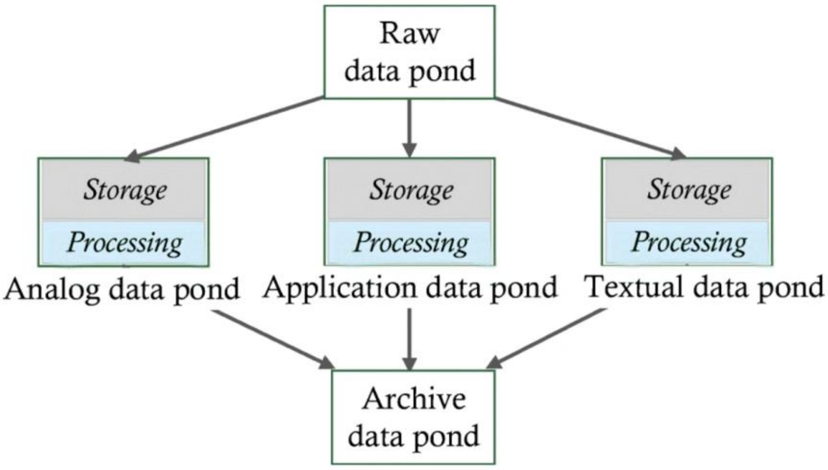


Figure 2.3 – Data flow in a pond architecture (Sawadogo & Darmont, 2021)

### 2.3.3. Data Lakehouse (DLH)

The DLH, as proposed by Errami et al. (2023), integrates the governance and reliability of DWs with the flexibility and scalability of DLs. It aims to expedite efficient knowledge extraction by combining the strengths of DWs, which provide a consistent and structured view of data, with DLs, which excel in storing diverse and raw data types. The DLH architecture inherits ACID compliance from DWs, ensuring reliability and consistency even in heavy write environments. It also addresses challenges faced by DLs by incorporating optimization techniques for SQL analytics and supporting data skipping to enhance query performance. The typical structure of a DLH architecture consists of an Extraction and integration layer, that interacts with the external sources, and the Process Control Framework, as shown in Figure 2.4. This second layer identifies 2 different logical divisions: a Data Lake and a Data Warehouse Model. The ingested raw data coming from the Extraction and ingestion layer is stored first in the landing area and secondly in the foundation area. From here the second logical division, the DW, will extract, manipulate, and analyze the now-refined data (Hlupić et al., 2022).

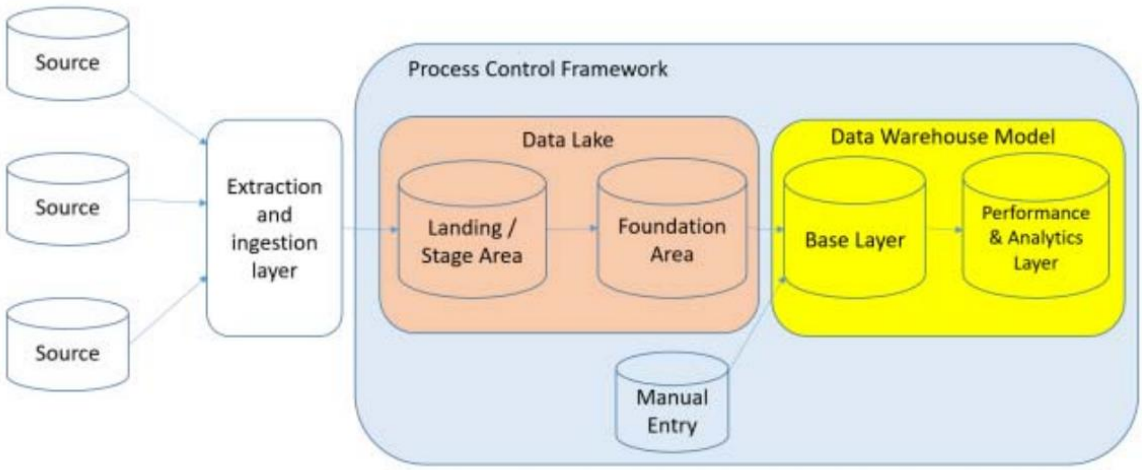


Figure 2.4 – Data Lakehouse architecture (Hlupić et al., 2022)

### 2.3.4. Comparison of the three models

While DWs are proficient in structured data analytics, DLs excel in handling diverse and raw data. The DLH, acting as a unified platform, aims to offer the best of both worlds. It addresses the cost and scalability issues of DWs and enhances the governance and consistency concerns of DLs. The DLH's support for semi-structured and unstructured data, along with SQL analytics optimization, positions it as an industry-standard solution for organizations dealing with Spatial Big Data (Errami et al., 2023; Harby and Zulkernine, 2023).

In conclusion, the DLH emerges as a transformative architecture, providing a great integration of DWs and DLs. By reconciling the strengths and weaknesses of both approaches, the DLH offers a unified platform for managing Spatial Big Data efficiently, ensuring data integrity, scalability, and flexibility.

## 2.4. Case Studies: Migration Strategies and Business Cases

We will now discuss the importance of data migration, a critical aspect of transitioning from an obsolete system to a modern platform. More in detail, we will discuss the transformation from on-premises to cloud and from a traditional DW to a DLH by emphasizing the need for informed decisions and a smooth transition to modern data management architectures, with some business cases.

### 2.4.1. General Guidelines for Data Migration

To execute a successful data migration is important to follow structured guidelines. Sarmah, S. S. (2018) outlines a specific strategy for end-to-end data migration that involves a series of 13 key steps to follow, portrayed in Table 2.3.

Table 2.3 – Key steps in End-to-end Data migration (Sarmah, 2018)

| Steps | Description |
|---|---|
| 1. Identify Legacy/Source Data | Clearly define and identify the data residing in legacy/source systems that need to be migrated |
| 2. Identification of Specific Configuration Data | Recognize any specific configuration data required from legacy applications for a seamless migration process |
| 3. Classification of Migration Process | Determine whether the migration process will be manual or automated, aligning with project requirements |
| 4. Profile Legacy Data in Detail | Conduct comprehensive profiling of legacy data, understanding its structure, quality, and potential challenges |
| 5. Identification of Data Cleansing Areas | Identify areas within the data that require cleansing or improvement to meet quality standards |
| 6. Mapping Attributes Between Legacy and Target Systems | Establish a mapping strategy for attributes, ensuring a smooth transition between legacy and target systems |
| 7. Identification and Mapping of Data to Historical Data Store Solution (Archive) | Recognize and map data that needs to be migrated to a historical data store solution for archival purposes |

| Steps | Description |
|---|---|
| 8. Gather and Prepare Transformation Rules | Develop transformation rules outlining how data will be modified or standardized during the migration process |
| 9. Conduct Pre-Migration Cleansing | Implement cleansing procedures before migration, addressing data quality concerns where applicable |
| 10. Extract the Data | Retrieve data from the legacy/source systems in preparation for migration |
| 11. Transform the Data with Limited Cleansing or Standardization | Modify and standardize the data as needed during the transformation process |
| 12. Load the Data | Load the transformed data into the target systems |
| 13. Reconcile the Data | Ensure data reconciliation, validating the completeness and accuracy of the migrated data |

By following these 13 steps we will be able to operate a data migration with a structured solid approach, with a minimized chance of risks, and by ensuring a successful transition. This list will be used later as the guideline to conduct the business case, however, these steps will be taken more as a reference to the migration process rather than a strict manual, to better follow the specific requirements of the real-life project.

### 2.4.2. Business Cases: Migration Experiences

A list of five business cases relatively similar to the one in this thesis was analyzed, by defining the main challenges, the method used, and the lesson learned by the specific case, with interesting results.

For example, in Jadhav's (2021) case study on migrating data from an on-premises Oracle database to an Azure cloud-based SQL Managed Instance using Azure Data Factory, several key challenges were encountered. These included encryption issues, testing strategies, and network considerations during the migration process. The study delves into the specifics of Azure Data Factory implementation, highlighting its crucial role in facilitating the migration. Key lessons learned from this experience emphasize the importance of thorough data validation and robust encryption practices to ensure a smooth and secure migration.

Examine Diamond's (2022) insights into migrating from a traditional Data Warehouse (DWH) to a Data Lakehouse (DLH) using Databricks: Diamond's study provides a comprehensive analysis of transitioning from a traditional Data Warehouse to a modern Data Lakehouse architecture using Databricks. The migration faced several significant challenges, including data schema transformations, integration with existing ETL processes, and ensuring data consistency across platforms. The main benefits identified from adopting the Data Lakehouse architecture include enhanced scalability, flexibility in data types and structures, and improved support for advanced analytics and machine learning models. In Diamond's research is highlighted also a couple of success stories from the companies Bread and Amgen, illustrating the transformative impact of migrating to the Databricks Lakehouse Platform. Bread, for instance, successfully migrated to efficiently handle growing data volumes, democratized data sets for analytics, and achieved a substantial 90% reduction in computing costs. Meanwhile, Amgen's migration facilitated increased cross-functional collaboration, accelerated data ingestion rates, and achieved a remarkable 75% reduction in processing times. These improvements resulted in faster delivery of insights and significant savings in computing costs, demonstrating the strategic advantages of adopting a Data Lakehouse approach with Databricks.

Analyze the migration of a web application from on-premises to Microsoft Azure, as presented by da Costa and Da Cruz (2012): the authors' study provides a detailed examination of migrating a web application from an on-premises environment to Microsoft Azure. The migration process involved several important steps, including the migration of databases and the modification of connection strings to adapt to Azure's cloud infrastructure. Their analysis included a detailed performance evaluation between the original on-premises setup, Windows Azure, and traditional web hosting solutions, presenting Azure's scalability and performance advantages in managing varying workloads and user demands. The main lesson learned from the study is the importance of meticulous planning, scalability considerations, and cost efficiency in cloud migration projects. Collaboration among IT teams and stakeholders is important for the successful adaptation to new technologies and architectures, such as Azure's cloud services. Embracing a cloud-native approach, including leveraging Azure's Platform-as-a-Service (PaaS) offerings, can yield transformative benefits in terms of scalability, flexibility, and operational efficiency.

The fourth case presented is focused on investigating the two-stage approach presented by Ellison et al. (2018) for accurately estimating duration, migration, and cloud running costs for relational databases: Ellison et al. (2018) proposed a strong methodology for estimating the duration, migration costs, and ongoing cloud running costs for relational databases using a two-stage approach. Their methodology involved developing workload and structure models to accurately simulate migration scenarios and predict resource consumption in the cloud environment. They employed discrete-event simulation techniques to validate these models, ensuring precise estimations of both upfront migration costs and long-term operational expenses. The key lesson to take from the study is that a successful evaluation of cloud database migration options necessitates a deep understanding of workload patterns and a rigorous simulation approach. By accurately predicting migration costs and duration upfront, organizations can make informed decisions about cloud adoption strategies and optimize resource allocation effectively.

The final case is based on the exploration of the challenges and optimization strategies involved in migrating a modular web application from on-premises to the cloud, as discussed by Olariu and Alboaie (2023): the authors examined the complexities of migrating a modular web application from on-premises infrastructure to cloud environments. Their study focused on identifying specific cost-saving measures and infrastructure optimization strategies to streamline the migration process and minimize operational expenses. They proposed a structured two-stage migration strategy, similar to the previous example. However, the case is more centered on transitioning initially from Infrastructure-as-a-Service (IaaS) to more managed Platform-as-a-Service (PaaS) solutions. The study highlights the importance of strategic planning and phased migration approaches to achieve cost reductions and operational optimization during cloud migrations. Informed architectural decisions, coupled with a comprehensive understanding of workload requirements, are essential for facing challenges and taking the most out of cloud benefits. The two-stage migration strategy proposed by Olariu and Alboaie proved effective in balancing scalability, cost management, and operational control, showcasing a pathway to successful cloud adoption for modular web applications.

In conclusion, the importance of correct data migration must be considered to minimize the potential risks and guarantee a successful transition to the new data architecture. This can be made not only by following a pre-defined structure but also by extrapolating the experience of similar business cases that can provide real-world insight. These lessons might be the key to understanding better the complexity of data migration, offering a valuable lesson in modern data management architectures. Nevertheless, these examples and Sarmah's 13 steps for data migration reflect only one side of the business case that this document will provide in the next chapter. This is due to the conditions of the case proposed that combines the more standard data migration from an on-premises structure to an on-cloud structure, with the relatively new migration from a DWH to a DLH architecture. Furthermore, there will be a period of transition in which both the old and new solutions will be merged, increasing the already complex environment.

# 3. METHODOLOGY

The objective of this chapter is also to explain the methodology used in this research. The chapter will be divided into three sections: in this first part it will be explained which method was the inspiration for the thesis and the general project planning, the second will describe the current state analysis and requirements, and the third will cover the proposed scenarios with their comparison.

This research utilized the Kimball lifecycle methodology, applying both quantitative and qualitative techniques. The Kimball lifecycle is a widely used structure and life cycle in the business intelligence domain. It consists of four steps: project planning, defining business requirements, creating the dimensional model, and ETL design (Landutama & Chowanda, 2023).

The Kimball lifecycle steps were defined to assist in the creation of system architectures and infrastructures tailored to business needs. Figure 3.1 below provides a summary diagram of the Kimball lifecycle methodology.

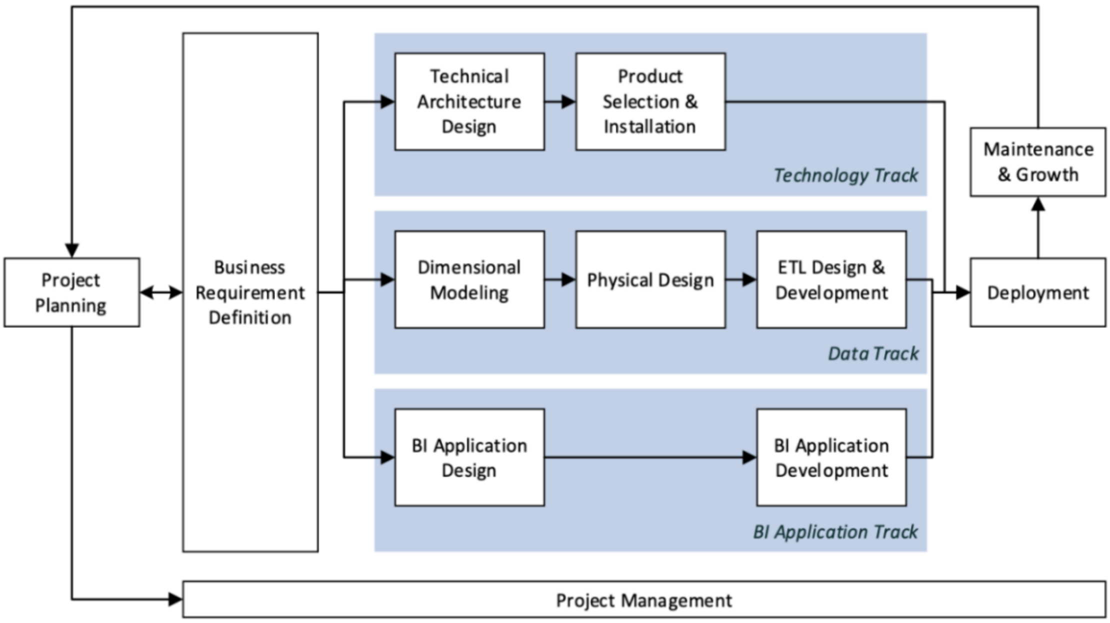


Figure 3.1 - Kimball Lifecycle Diagram from Landutama & Chowanda (2023)

Similar to the Kimball Lifecycle Diagram, an initial Project planning was drafted with the definition of the scope, objectives, methodologies, and timeline needed. The scope of this project was focusing on utilizing the BI capabilities within the client's organization. Key aspects included improving data integration, governance, quality, and frontend usability to support informed decision-making. The main objectives were to analyse the current state of the client's BI infrastructure and formulate recommendations for improvement, specifically for the fourth point mentioned above. The project used a mixed-method approach combining qualitative and quantitative research methodologies such as document analysis, case studies, and observational data. The project timeline is structured into phases, beginning with preliminary research and analysis, followed by recommendation formulation and validation. Implementation planning and reporting are important components, ensuring alignment with project milestones and client expectations. Regular progress reviews and adjustments are scheduled to maintain project momentum and address emerging challenges rapidly.

## 3.1. Current State Analysis

Within the program of digital transformation and adoption of S4/Hana, it was decided to start an evolution journey for the Business Intelligence area of the client. The everyday data that the firm was using to manage the day-to-day operations now required to be held and processed through a BI phase to be able to generate various analyses and predictions. With this purpose in mind, it was created a first analysis and the corresponding recommendations, aligned with the best practices and goals of the client's current BI.

The area of investigation is transversal and embraces multiple disciplines, both hardware and software. The analysis delineated six macro areas and proposed for each at least one recommendation.

### 3.1.1. Scalability of the infrastructure

Current situation of the client:

- Analysis of the spectrum of the data with a near real-time /intraday cadency.
- Dynamics of the scheduling of the data flows are not always predictable.
- Functions of the IT department are based on an on-premises DWH that limits the scalability required by the business soon.

The recommendations proposed are changing to a cloud-based solution that guarantees limitless scalability and focuses on this hybrid approach: an ETL layer based on Serverless technologies, a layer of Data Intelligence Services[2], and the presence of multiple segregated environments powered by the same data source (Production, Test, and Development).

### 3.1.2. Maintainability of the infrastructure

Currently, the situation for the maintainability of the firm's infrastructure is the following:

- The current infrastructure is composed of an SAP-centric architecture for integrating the Data Services, with multiple Connectors[3] and a list of isolated services created ad-hoc for specific tasks.
- Is required now a high effort to maintain the cluster operating.
- No structured documentation for the maintenance process is available.

The recommendations for this area are to switch to a cloud solution based on Microservices that will bring the following benefits:

- The possibility to operate in 2 environments completely separated one from the other: the old data from ECC, or ERP Central Component, and the new proposed one from SAP HANA/S4, or S4 for simplification.

[2] Data Intelligence Services Service is a shared, managed service that enables powering the Analytics and Machine Learning Capabilities of GBU Apps (Oracle, 2021).

[3] Architectural element that models interactions among components and rules that govern those interactions (N. Taylor et al., 2008)

- Connectors unified provided by the Service Provider

### 3.1.3. Data Integration

The as-is situation is defined by the following:

- Presence of a Data Warehouse Microsoft SQL Server containing most of the logic of Data Integration in the form of Stored Procedures[4] (SP), and partially in Data Services
- The ETL is SAP Data Services with a not very structured composition.
- Sequel Services Integration Services (SSIS) as the execution scheduler of the queries. A chain connection of the SP makes the consequentiality of the processes.

The recommendations provided are to integrate the Extraction and Data Transformation processes with a unified platform and manage the scheduling with a dedicated solution to guarantee stability and reliability even in case of a failure in the process connected with data or the procedure itself. Another suggestion is the introduction of layers to unify the data models and guarantee compatibility with the dashboards (the risk is to have different data for different dashboards not aligned).

### 3.1.4. Data Governance

At the moment the client is missing a list of important instruments that were recommended to implement: a Monitoring and Tracking tool centralized for the pipelines, an instrument for Data Lineage[5], and one dedicated to Health Monitoring of the data (a set of logic for the notification of the presence of errors and/or significant events that need to be taken into consideration), and, finally, the implementation of a Data Catalog[6].

### 3.1.5. Data Quality

The current situation is characterized by the absence of a dedicated structure for Data Quality and an approach of trial and error to solve problems. A more efficient structure is defined by the deployment of a microservice dedicated to validating data from the sources and the possibility of developing a set of KPIs to inform the users and to be used in the validation process.

### 3.1.6. Business Intelligence (frontend)

The situation that the client is facing now has many areas for improvement: there is the presence of many cases of Shadow IT[7], the PowerBI Pro license that does not consent to have versioning of the dashboards and reports currently present, Row-Level Security logics that do not allow in PowerBI to have cross-department visualizations, the access of the dashboards that need to be updated, and

[4] A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system (RDBMS) as a group, so it can be reused and shared by multiple programs (Hunghes, 2024)

[5] Data lineage is the process of understanding, recording, and visualizing data as it flows from data sources to consumption (Imperva, 2024)

[6] Data Catalog is an organized inventory of data assets in the organization (Oracle, 2024a)

[7] Shadow IT is the use of IT-related hardware or software by a department or individual without the knowledge of the IT or security group within the organization (Cisco, 2024)

modifications of measures and calculated columns often done in PowerBI directly and not in the Tabular (impact on the performance of the reports).

## 3.2. Scenarios proposed

Based on the 6 areas' recommendations, three possible scenarios for the new architecture of the client were designed. Each of the different proposals was trying to maximize the benefits taking into consideration the limited number of resources and funds available. To do so were explored 3 different technologies with a different degree of innovation, price, and performance.

### 3.2.1. Scenario 1: Managed SQL Database

The first scenario consists of an SQL-DB on-prem solution in which the entire ETL process is based on an SQL database, as well as the data storage function. The solution is a relatively easy and standard solution that has the advantage of being familiar with the client's experience and the ability to be a robust scenario for data management. Another characteristic is the presence of an automatic tuning [8] present in the Azure Functions[9] to interface the data model ECC with the equivalent S4/Hana, the storage based in Azure's Data Lake Storage Gen 2, and the refresh of the Analysis Services triggered by the competition. The overall solution has the higher price of all 3 due to the physical database.

### 3.2.2. Scenario 2: Databricks

This second scenario is primarily focused on the Databricks service, a tool that "Help to connect the sources of data to one platform to process, store, share, analyse, model, and monetize datasets solutions from BI to generative AI" (Databricks, March 2024). In this proposal, Databricks is used as a cloud-based big data processing platform with a higher performance than the other solutions and a medium-high price.

### 3.2.3. Scenario 3: Serverless Services

The third and last proposal was also the one voted by the client. This solution is based on a fully serverless architecture, in which the ETL process is entirely based on SQL Pools Serverless, a solution that does not require a physical component. The Serverless Service architecture stands out as the preferred option due to its inherent advantages:

- Cost-Effectiveness: By using serverless computing and storage resources, this architecture optimizes cost efficiency by only consuming resources as needed, eliminating the need for costly infrastructure investments. Of the 3 proposals is the cheapest one due to the lack of a physical DB.
- Scalability: The serverless nature of the architecture enables full scalability, allowing our data platform to adapt to variable workloads and data volumes without manual intervention or resource provisioning.

---

[8] Azure SQL Database and Azure SQL Managed Instance automatic tuning provides peak performance and stable workloads through continuous performance tuning based on AI and machine learning (Microsoft, January 2023)

[9] Azure Functions is a cloud service available on-demand that provides all the continually updated infrastructure and resources needed to run your applications (Microsoft, May 2024)

- Flexibility: Serverless services offer high flexibility, allowing developers to focus on application logic and data processing tasks without facing infrastructure management limitations.
- Simplicity: The streamlined nature of serverless architectures simplifies deployment, management, and maintenance, reducing operational costs and increasing agility.

### 3.2.4. Scenarios comparison

The strengths and the weaknesses of all these 3 scenarios are now represented in a unified table, to better visualize the 3 proposals, as shown in Table 3.1

Table 3.1 – Architecture's scenarios

| Scenarios | Strengths | Weaknesses |
|---|---|---|
| 1. Managed SQL Database | Familiarity with the client's experience | Higher price compared to other scenarios due to physical database |
| | Robust scenario for data management | No scalability |
| | Automatic tuning present in Azure Functions | Expansion of data will require substituting the server with a pricey one |
| | Integration with Azure's Data Lake Storage Gen 2 and Analysis Services | Is a temporary solution that will last 3-4 years |
| 2. Databricks | High performance for big data processing | Medium-high price |
| | All-in-one platform for processing, storing, analyzing, and modeling datasets | Overperforming: The solution is providing a higher number of perks that, however, are not needed by the client |
| | Cloud-based solution, reducing infrastructure overhead | Dependency on a cloud-based solution: reliant on external cloud services |
| 3. Serverless Services | Cost-effectiveness due to serverless computing and storage resources | The lack of a physical database might limit certain functionalities |
| | Full scalability to adapt to variable workloads and data volumes | Dependency on a cloud-based solution: reliant on external cloud services |
| | High flexibility for developers without facing infrastructure management limitations | The possibility of increased complexity in monitoring and debugging |
| | Simplified deployment, management, and maintenance | |

Among the three scenarios considered, the third one, Serverless Services, was the one chosen mainly because of the more affordable price by the client itself. It uses serverless computing and storage, cutting down on costs by avoiding expensive infrastructure. Plus, it scales easily and is flexible for developers, which makes it a practical choice for data needs. The client after the proposal of all of the scenarios, opted for this option because it offered the most benefits at the lowest cost, without additional perks that were not needed like the second scenario had.

# 4. EMPIRICAL STUDY

The objective of this chapter is to analyse the empirical study. To do so the chapter is divided into three main parts: the overview of the architecture chosen for the DLH, the Stored Procedures and their transformations to fit the new architecture better, and finally, the Tabulars and PowerBI used to create the relations between tables, their measures, and the reports. The overall architecture of the Data Platform in Azure is based on the serverless scenario proposed previously, as it has the most advantages and the best performance-to-cost ratio of all the proposals. This specific solution involves 3 different areas with the corresponding technologies, as seen in Figure 4.1.

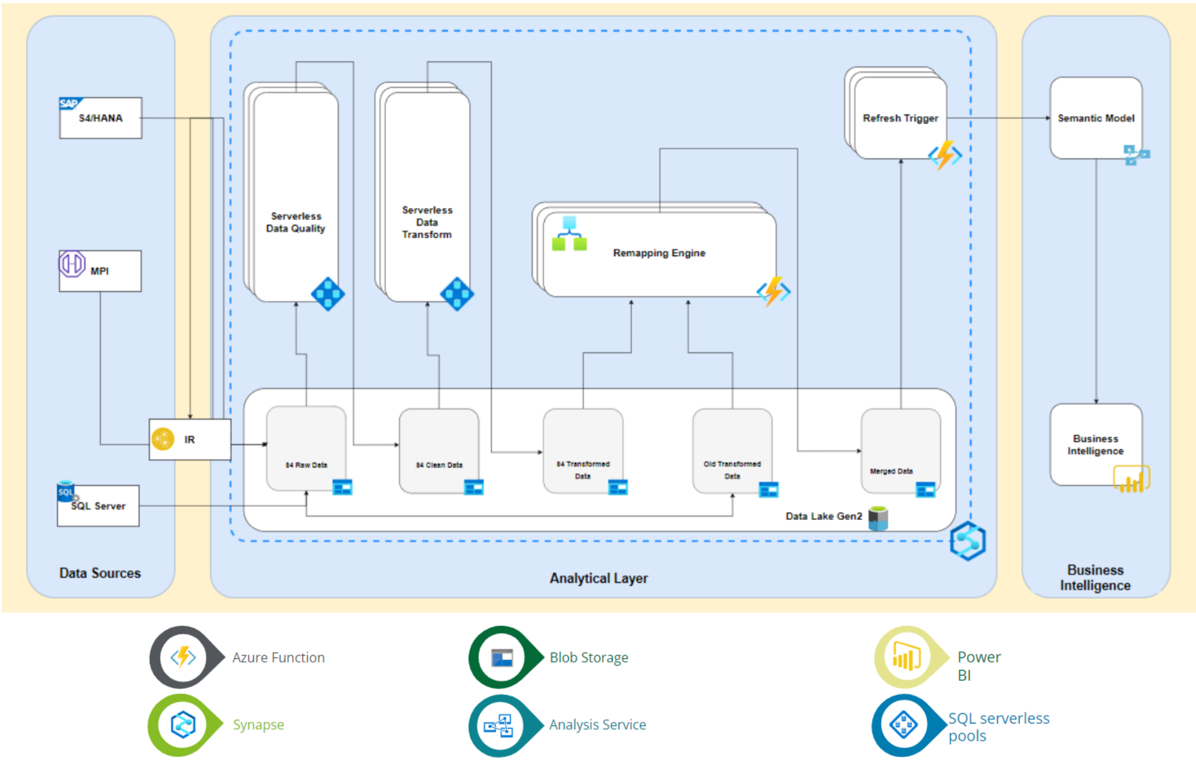


Figure 4.1 – Managed SQL Database Architecture

The Data Sources layer, in which the data are extracted for the BI processes, involves 3 sources:

1. The new SAP s4/HANA which is the new source of data will be generated from now on and it will gradually replace the previous source of information within 3 years. This source involved 2 different parts: CDS Views[10] and Z-CDS Views. Both are Core Data Services of SAP for the extraction of data with the main difference being that the CDS are standard database tables and views already built in SAP while Z-CDS views are custom-made extraction functions, created to accustom the requirements of the client. In total, 132 tables were extracted, of which 81 dimensional tables (identified by the prefix “DIM_”), 34 fact tables (identified by the prefix “FAC_”), and 17 flat tables (tables that were not related to other tables).

[10] Core Data Service (CDS) Views are virtual data models of SAP HANA which allows direct access to underlying tables of the HANA database (Bryteflow, 2024)

2. The SQL-Server on-premises represents the old architecture of the client and will remain operating during the migration period in which both the old and new data sources will be used. This decision was made by the business itself to have a smoother transition due to the complexity and size of the firm and the high amount of offshore involved. Nevertheless, this new requirement will increase the complexity of the solution (like the need for a Merge layer) and the handling of the data.
3. The MPI[11], or Message message-passing interface, provides custom REST[12] Endpoints that allow the extraction of relevant objects within the analysis' perimeter.

In the second layer, the Analytical Layer, all the transformation processes are involved. Here the data from the data source is temporarily stored as a parquet file in the Blob Storage of Azure within Data Lake Gen 2. "The Blob (Binary Large Object) Storage allows storage lets developers to store large amounts of unstructured data in Microsoft's cloud platform" (LogicMonitor, November 2022). During this phase, the data will be processed multiple times to control the data quality, apply the logic behind the SP, and ensure the correct upload into the BI layer with constant performance. The technologies involved in this phase are:

- Storage Account in the form of di Azure Datalake Gen2 named *bisynapsestorage.* It is inside its perimeter in which a vast amount of structured and unstructured data is stored, the majority of which is in the Apache Parquet[13] format.
- Synapse Studio named *bi-synapse-default.* Is the unified workspace for data integration, orchestration, and analysis.
- SQL Pools: computational and storage resources that can be on-demand and serverless or purchased as instances at an hourly rate. Their object is to facilitate the efficiency of data querying, processing, and manipulation inside the Synapse Analytics environment.
- A list of Synapse Pipelines that allowed the creation and execution of a complex data workflow. The main scopes for the pipelines are the following:
    a. DataFlow: "Uses your pipeline code to create an execution graph that represents your pipeline's PCollection and transforms and optimizes the graph for the most efficient performance and resource usage" (Google Cloud, March 2024).
    b. Copy activities: "In Azure Data Factory and Synapse pipelines is used to copy data among data stores located on-premises and in the cloud (…). You can also use the Copy activity to publish transformation and analysis results for business

[11] The Message-Passing interface (MPI) is a standardized means of exchanging messages between multiple computers running a parallel program across distributed memory (S. Gillis & J. Bigelow, 2023)

[12] REST determines how the API looks like. It stands for "Representational State Transfer". It is a set of rules that developers follow when they create their API. One of these rules states that you should be able to get a piece of data (called a resource) when you link to a specific URL (Liew, 2021)

[13] Apache Parquet is an open-source, column-oriented data file format designed for efficient data storage and retrieval (Databricks, 2024a)

intelligence (BI) and application consumption." (Gawande P., et al., October 2023).

c. Synapse Connector (SAP, SQL, WebAPI): The definition for Data Connectors is "Software components that automatically extract data (…) from one or more upstream data sources and land that data in another database" (Incontra, 2023). In our specific case, the data sources are 3: the SAP s4/HANA, the previous SQL Database Server on-premises, and a WebAPI[14].

- Azure Functions: are serverless compute resources that allow the execution of custom logic without worrying about maintaining the infrastructure behind it and reducing the cost of maintaining the servers (Gailey, G., et al., May 2023). In our case is used to run some of the SPs, the remapping engine, and the refresh of the triggers.

The final layer is the one dedicated to Business Intelligence in which the processed data are inserted into a Semantic Model governed by a Tabular model. The Tabular models can be defined as "Databases that run in-memory or in DirectQuery mode, connecting to data from back-end relational data sources (…) that delivers fast access to tabular model objects and data by reporting client applications like Power BI and Excel" (Follis, et al., November 2022). Here additional small transformations are made from the final views located in the Data Lake Gen 2 like the creation of most of the measures needed for PowerBI, the calculated columns, and the subdivision of the information in 6 main areas based on the Business needs. These areas are Controlling (CO), Purchase, Customer Service (CS), Sales Distribution (SD), AR-SD (Accounts Receivable + Sales Distribution), and SD-CO (Sales Distribution + Controlling). These Tabular are alimented by the same data but contain partial information needed only for the specific business unit that they represent. For each of them, there are multiple PowerBI reports connected, with a total of 68 reports, divided into 12 workspaces.

## 4.1. Analysis and Transportation of Stored Procedures

As part of the migration to the new data platform, intervention in the Stored Procedures of the business was imperative. The SP consists of a list of logic custom-made by the client itself to convert the original data extracted from the data sources into a more business-focused table. This is a delicate process that is fundamental for the firm to be able to read and apply its insight from the data. To understand and use this information in the best way possible this logic needs to be applied according to the needs of the client itself. During the project, the SPs needed to be adapted and translated to the new software (Azure Synapse) and remapped according to the new name given to the initial 132 tables in ECC now converted into S4. This crucial passage extracts the tables from the S4 Cleaned Blob storage data, applies the transformation, and then inserts the new information into the S4 Transformed Blob Storage, as shown in Figure 3.1.

An initial analysis was made of the overall SPs involved, to better understand the function of each SP, divide them into different categories, and understand the best way to run them and extract the data. The overall Stored Procedures analyzed were 64, divided into 3 waves of actions based on their importance.

[14] A Web API is an application programming interface for the Web (W3schools, 2024)

### 4.1.1. Challenges from an on-premises database to a serverless one

During this phase, some discrepancies between the Synapse Serverless SQL pool and the original SQL Database were discovered. The main issue encountered was that the single table extracted and inserted into the SP needed to be modified multiple times before its finalization. For example, the output table of many of the SPs was changed many times with the command UPDATE inside the same Stored Procedure before finalizing it. This meant that for each modification a new version of the table needed to be saved and stored in the DB, replacing the original one. This was an easy task for the on-premises DB that just needed to modify the original table by overwriting the new commands. However, this solution, which worked perfectly with a physical DB, could not be applied to a serverless one as the original table was a view of the parquet file inside the DLH and could not modify the source but only generate a new one. To reduce the number of parquet files generated, was decided to divide each of the SPs into multiple subparts, each of them containing one singular transformation of a table. So, for example, if an SP contains 2 UPDATES and one INSERT INTO, 3 blocks will need to be created instead of one unified SP. For each of these new blocks was created an SQL Procedure that through a view of an External Table[15] was able to read the singular block, as seen in Figure 4.2.

```
CREATE OR ALTER PROCEDURE dbo.SP__nameTable_nameSP__1 AS
DROP VIEW IF EXISTS dbo.VW_nameTable_nameSP_1
EXEC dbo.sp_executesql N'
CREATE VIEW dbo.VW_nameTable_nameSP_1 AS
...
DECLARE @sql_drop NVARCHAR(MAX)
SET @sql_drop = '
  IF EXISTS ( SELECT * FROM sys.external_tables WHERE object_id =
OBJECT_ID('nameTable_nameSP_1'') )
  DROP EXTERNAL TABLE dbo.nameTable_nameSP_1
    '

EXEC (@sql_drop)
CREATE EXTERNAL TABLE [dbo].[nameTable_nameSP_1] WITH (
     LOCATION = 's4hana-sp-steps/nameTable_nameSP_1/',
     DATA_SOURCE = [DataSource_SP_Dev],
     FILE_FORMAT = [ParquetFF]
) AS
SELECT
  *
FROM dbo.VW_nameTable_nameSP_1
```

Figure 4.2 – Frame for the block in the Stored Procedure

[15] It enables you to access data in external sources as if it were in a table in the database (Oracle, 2024b)

Other new requirements generated by the serverless one were that not only the SP code needed to be divided, but also basic SQL functions such as UPDATE, INSERT INTO, and INTO, needed to be altered, as well as the name of the tables and columns mentioned with the new names provided by S4. The INSERT INTO was replaced with a UNION ALL (Figure 4.3, A), eliminating the INTO name_table line (Figure 4.3, B) as the entire block will be saved into a view.

```
SELECT *
FROM name_table

UNION ALL
(...)
```

**(A)**

```
SELECT ...,
INTO name_table,
FROM name_table
```

**(B)**

Figure 4.3 – New code for INSERT INTO (A) and elimination of INTO a temporal table (B)

Regarding the UPDATE cases, the most common ones, the fix required was more complex, as the UPDATE command was not supported by the new architecture. To replace it was used the COALESCE function, which takes two columns/sources as input and returns the second one in case the first one is blank. As seen in Figure 4.4, the COALESCE is merging the new logic for the modified rows with the original one thanks to a LEFT JOIN made from the same table.

```
UPDATE
name_column
FROM table
```

**(A)**

```
SELECT X0.*
COALESCE (X1.name_column_1,
X0.name_column) AS name_column_new
FROM table X0
LEFT JOIN
    (SELECT X0.*,
    [Insert new logic here] AS name_column_1
    ) X1
ON ( X0.key_column = X1.key_column)
```

**(B)**

Figure 4.4 – Original form for UPDATE (A) and the new version (B)

Not all the SPs needed these changes, in fact, out of the 64 Stored Procedures, 5 were excluded because out of scope, and 12 contained a too-complex code to be replicated with this approach. The 5 excluded SPs mostly served to extract old data from external Excel sheets, like for example the information for suppliers not present in SAP. It was decided to insert them into SAP rather than keep the additional data source as the Excel sheets were not reliable and could have ended up polluting the DLH with wrong information. The other 12, instead, contained a complex form of calculus named Cursor. "A cursor in SQL is a database object stored in temp memory and used to work with datasets. You can use cursors to manipulate data in a database, one row at a time. A cursor uses a SQL SELECT statement to fetch a rowset from a database and then can read and manipulate one row at a time" (Choudhary, August 2023). To solve this problem the entirety of the SPs were inserted into a staging environment within the Azure Functions, a more costly solution, that allowed them to run smoothly the full Stored Procedure without the need to divide it into blocks or change the code, besides the remapping of the S4 tables.

The final modification to the SPs were the creation of 2 additional, one for a further transformation of the data at the beginning of all the SP, and another at the very end, to ensure the correct merge of the S4 Transformed Data Blob to the Old Transformed Data (Figure 3.1). The 2 new codes are the following:

- PRC_INITIAL_MASTER_DATA: This initial stored procedure (SP) is introduced to perform preliminary transformations, such as adding columns to main tables or performing unions to achieve the desired structure. Within its code, the blocks created for Azure Functions (AF) are also included.
- PRC_FINAL_STRUCTURE: This final SP contains all the FINAL blocks for tables unmodified by intermediate SPs. FINAL blocks are the final version of all the tables that need to be merged with the old table and therefore need to have an exact match of columns name and datatype.

## 4.2. TABULAR AND POWERBI LAYER FOR REPORTING

Inside the 3rd layer of the data model's architecture, named Business Intelligence, are presence 2 tight connected technologies: the Tabular and the PowerBI reports. The second one is the final layer in which the finalized version of the information extracted from all the data is shared with the entire company. Here 68 reports, divided into 12 specific workspaces inside PBI, represent the main information tool for the firm to share, monitories, and understand the major trends. From these areas, we can identify the main ones the following: Purchase, Customer Services, Finance, Controlling, Sales Distributions, and Supply Chain Management. The reports in each workspace are alimented from the same original information but portioned based on the specific information needed to satisfy the reports' objectives. This division is quite like what happens inside a data mart[16], where a partial of the same data is given as input in a specialized DB, however, the source of input for this case is a tabular project of Analysis Services for Microsoft Visual Studio. Tabular models can be defined as "databases that run in-memory or in DirectQuery mode, connecting to data from back-end relational data sources" (Follis, et al., November 2022). It was preferred this alternative instead of using, for example, the naïve Power Query model already built in Power BI reports for different reasons: the power and velocity of the instrument, the possibility to feed to multiple reports the same data, calculated columns, variables, etc. from one unique source, and the simplification in maintaining the reports.

From all the already existing tabular, 6 were taken into consideration for the first switch of data: CO, CS, AR-SD, SD-CO, SD, and Purchase. Each of these tabulars contains different versions of the same information tailored to what the tabular needs to showcase. For example, the AR-SD contains the aggregated information from the Account Receivables and Sales Distribution, represented by a total of 16 different tables, as shown in Figure 4.5.

[16] A data mart is a subset of a data warehouse focused on a particular line of business, department or subject area (IBM, n.d.)

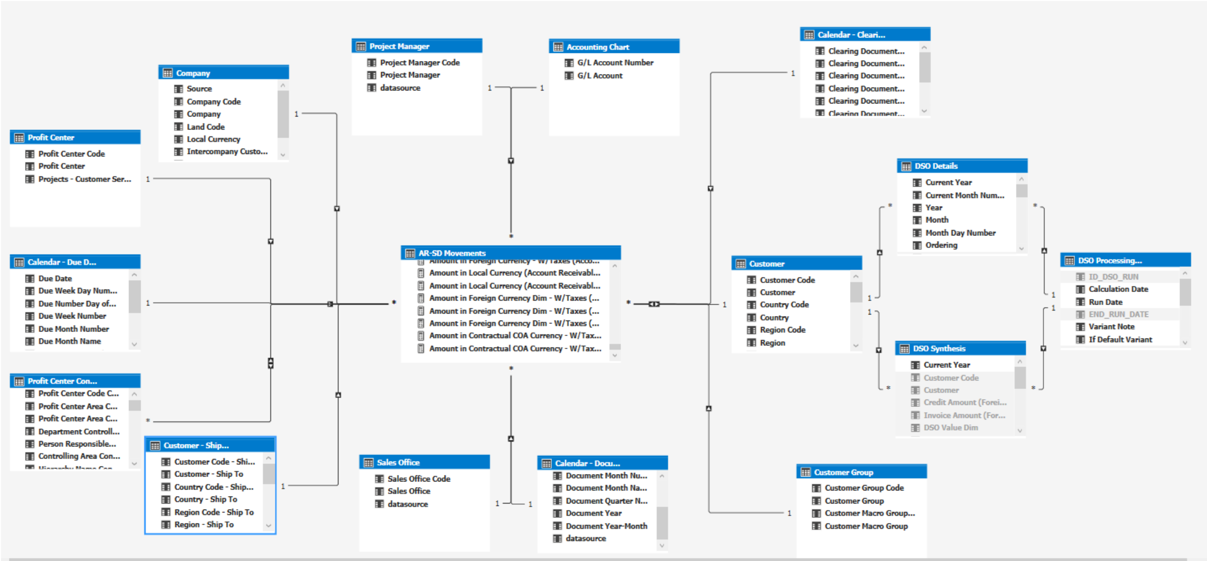


Figure 4.5 – Tabular of AR-SD

This representation, very similar to the one proposed by Power Query, contains a list of dimensional tables connected to the factual table, in this case, AR-SD Movements at the center of the image in Figure 4.5. This tabular, like all the other 5, contains a snowflake schema in which the various dimensional tables are broken down into subdimensions (Databricks, 2024b). During this phase, the main structure of the model was already present, however, the introduction of new data coming from a different source caused a series of problems that needed to be fixed. We can sum up those issues in the following list:

- New requirements: Both IT and Business came with several requests for adding numerous new columns, measures, and logic. One of the most important ones was the Datasource column, fundamental to distinguishing the origin of the data.
- Merge the two data sources: These requests varied a lot but came from the same original problem on how to combine in the same tables data from 2 different origins. One of the first steps done in this direction was the creation of the final layer of the SP of the FINAL block in which the datatype from S4 data was matched exactly to the ECC one. The second step was making sure that the dimensional tables did not have duplicated data and that the new dimensions from S4 were winning against the ECC one. This was possible by taking all the S4 data unified with the ECC one minus the S4 (Figure 3.6). Finally, a high amount of time was spent on making sure of the uniqueness of the Keys of the Views that feed the Tabulars.
- Hierarchy Names: The S4 source generated sets of new hierarchies for the 4 main parts of the database. The new names came from 4 different areas: Hierarchy Costs, Elements, Purchase, and WBS (Work breakdown structure). Each of these areas contained from 5 to 10 leaf-level with a corresponding code and name. Many of these codes changed from ECC to S4, and many kept the same code but a different name or vice versa. The challenge was now making sure that all the filters made with one of these levels were considering both the old and new names. This included not only filters in the

PBI reports but also in the measures, KPIs, and calculated columns created on the tabulars.

```
SELECT *
FROM table
WHERE datasource = 'S4'
UNION ALL
SELECT *
FROM table
WHERE datasource = 'SQL'  AND KeyColumn NOT IN
    (
    SELECT *
    FROM table
    WHERE datasource = 'S4'
    )
```

Figure 4.6 – Merge two data sources

Another challenge was the need to keep refreshing the data generated daily and inserting it into the 6 tabulars. This task was aggravated due to the ongoing parallel project of implementing SAP as a main data source. This meant that much of the data input was not always cleaned and needed to be checked by the BI team and the business itself. The most common problem was the insertion of duplicated or corrupted data that might end up blocking one of the tabulars. To better monitor the everyday situation of each of them, a new PBI report was created. This dashboard could not only instantaneously tell if one of the tabulars was down at the moment but also provide a history of the up and downtime them, as seen in Figure 4.7.


Figure 4.7 – Monitoring report

The monitoring report is composed of 3 parts, from the top to the bottom of the sheet:

- A set of filters such as the Status of the tabular, Name, and the number of days that we want to see the historical trends until now.
- The main traffic light visual to check the status of the tabulars: either OK or KO.
- The Historical trends of the tabular with the days and the duration of the downtimes.

In conclusion, Chapter 3 provides a detailed analysis of the client's existing BI infrastructure. Each section examines different facets of their BI situation, highlighting challenges and proposing practical recommendations for improvement. From scalability issues to governance gaps and usability concerns, the analysis discusses the complexities faced by modern BI environments. Implementing these recommendations will enable the client to strengthen their BI setup, maximizing the utility of their data assets for informed decision-making. Looking ahead, Chapter 5 will present results and discussions, building upon the groundwork laid in this analysis.

# 5. RESULTS & DISCUSSIONS

The modernization of the client's Business Intelligence (BI) infrastructure to a new platform required a complex solution to improve operational insights and decision-making capabilities of the firm. In this chapter, we discuss the outcomes of the migration project, examining the impact on various aspects of the BI landscape. From the conversion of Stored Procedures (SPs) to the optimization of tabular structures, each step of the migration will be analysed to understand its implications and challenges. Through an in-depth analysis, the aim is to extract insights based both on the successes and challenges faced by the migration project, offering recommendations for future improvements as shown below.

## 5.1. IMPACT ANALYSIS

This section will explore how the migration project impacted four key areas of the client's Business Intelligence setup. Each area has its challenges and opportunities, affecting how well the company can operate and make decisions. From changing Stored Procedures to modifying tabular structures, the object will be to look at how each change has affected the business, hoping to better understand the overall impact of the migration project and draw important lessons for future projects.

### 5.1.1. Converting Stored Procedures in Azure Synapse

The transition from on-premises SQL Server to Azure Synapse needed the conversion of Stored Procedures (SPs) to suit the new serverless architecture. This conversion process introduced several complexities and challenges. Initially, SPs were singular entities, containing logic tailored to the on-premises environment, easily identifiable, stored, and maintained. However, in the context of Azure Synapse's serverless model, SPs needed to be decomposed into smaller, modular blocks to align with the platform's architecture.

The decomposition of SPs into smaller blocks led to increased complexity and reduced visibility of the overall scope of the procedures. With an average of 5 substeps per SP, the number of blocks to maintain increased from 64 SPs to 320. The blocks were indeed enumerated and classified in a way that was easier to navigate, by having the one coming from the same SP into one folder and by naming them with the following structure: SP__ +*Name of the Store Procedure* + _ + *Name of the table of the FROM* + __+ #, with # the enumerating number of the same table in the same SP. For example, if in the original SP, there were 2 FROM with table A and one of table B, the blocks will have the following name: SP_*NameSP*_A__1, SP_*NameSP*_A__2, and SP__*NameSP*_B__1.

As imaginable, each block required careful management and coordination to ensure seamless execution and data flow. Additionally, the remapping of SQL commands and functionalities posed challenges, particularly with commands like UPDATE and INSERT INTO, which required alternative approaches in the serverless environment.

Furthermore, the absence of optimization efforts during the migration process made the running of the SP a higher time requirement operation than it should have been. SPs, running sequentially and with interdependencies, resulted in longer processing times since they have to be processed one after the other, meaning each error will cause the run to be stopped. The lack of optimization further compounded these issues, leading to suboptimal performance and increased maintenance overhead.

### 5.1.2. Performance Optimization and Time Efficiency Issues

An important consequence of the migration project was the absence of dedicated efforts towards Stored Procedure optimization. While the serverless architecture offered scalability and flexibility, it also introduced challenges in optimizing SPs for efficient execution. As mentioned in the point before, the sequential execution of SPs and the interdependencies between the processes contributed to increasing the processing times. This will not only lead to the business having the data ready to be analyzed later but also will mean a higher computational requirement for the server and overall an increase in the cost.

The sequential nature of SP execution, combined with the lack of optimization efforts, resulted in prolonged data processing times. SPs running in sequence require considerable time to complete, and this can impact operational efficiency and decision-making timelines. Additionally, the interdependencies between SPs meant that if some errors or delays stopped one process, it would cascade through the next SP, blocking an entire tabular or more from being refreshed with the daily data until the error was solved.

### 5.1.3. Suboptimal Tabular Structure and Integration Challenges

The migration project highlighted the inefficiencies of the tabular structure, particularly in terms of schema design and integration challenges. The existing tabular structures did not present, in fact, a snowflake or a star schema structure but rather a three-leaf one, with numerous bridge tables connecting factual and dimensional tables. These many times were not connected directly to a fact table but to another dimensional table, ending up creating avoidable complexity and reducing the efficiency of the tabular. This suboptimal design impacted query performance and data accessibility, lowering the realization of expected benefits from the migration.

Moreover, maintaining both the old and the modern BI systems ends up introducing numerous integration challenges. Ensuring data compatibility and consistency between the two systems required extensive efforts, including data mapping, schema alignment, and data transformation. The introduction of a FINAL table further complicated integration processes, leading to delays in the timeline of the project due to the additional new work.

### 5.1.4. Transition Challenges and Extended Data Processing Time

The decision to maintain the old BI systems alongside the new architecture introduced transition challenges and extended data processing times. Coordinating the data refresh processes across multiple systems required additional planning and synchronization efforts for the team. The data was, in fact, first needed to be processed by the old BI, then extracted alongside the new data coming from SAP, and only then elaborated by the SP to finally be merged with the old BI one and refreshed by the tabular, so the business was able to visualize it. This was a daily operation that on many occasions created friction between the various stakeholders due to the longer expected time needed to be executed. Delays in data availability post-refresh, and causal underperformances of the regional server due to an overload of requests in rush time, lowered operational efficiency, and delayed decision-making.

The extended data processing window, together with unpredictable server performance, posed operational challenges throughout the entirety of the project. Fluctuations in server performance introduced variability in data processing times, making it difficult to establish reliable processing schedules and meet business demands consistently.

## 5.2. Discussion

The assessment of the current on-premises SQL solution revealed limitations in scalability and operational efficiency, after the decision to migrate to a cloud-based Data Lakehouse (DLH) structure using Microsoft Azure. This transition aimed to use Azure Synapse's serverless capabilities for flexible data processing. The objectives of the project included assessing the current state of the BI infrastructure, designing and implementing the DLH structure, evaluating the migration's impact on efficiency and cost reduction, analyzing encountered challenges, and providing recommendations for future transitions. However, the migration highlighted challenges in converting Stored Procedures (SPs) and optimizing them for efficient execution in the serverless environment. Despite achieving improved scalability and data processing flexibility, the project faced increased processing times and operational costs due to the complexities in SP optimization. The analysis also identified integration hurdles between legacy and modern BI systems, contributing to extended data processing timelines and operational inefficiencies. Recommendations emphasized the importance of comprehensive planning, stakeholder collaboration, and iterative testing to mitigate risks and maximize the benefits of cloud-based BI infrastructures. The five objectives and the main findings per each are summarised in Table 5.1, as shown below.

Table 5.1 – Summary of Impact Analysis

| Objective | Key Findings |
|---|---|
| 1.Assess current state | Identified opportunities included migrating to a cloud-based solution for enhanced scalability and reduced maintenance overhead. |
| | The assessment revealed limitations in scalability and operational efficiency due to the on-premises SQL infrastructure. |
| 2.Design and implement DLH | Successfully implemented a DLH structure on Azure Synapse, leveraging serverless technologies for flexible and scalable data processing. |
| | Challenges included the decomposition of Stored Procedures (SPs) and optimization for efficient execution in a serverless environment. |
| 3.Evaluate the impact on efficiency, cost, scalability | Impact analysis indicated improvements in scalability and data processing flexibility. |
| | However, challenges in SP optimization led to increased processing times and operational costs. |
| 4.Analyse migration challenges | Challenges included SP conversion complexities, integration issues between legacy and modern BI systems, and extended data processing times. |

| Objective | Key Findings |
|---|---|
| | Solutions involved refining SP optimization strategies, redesigning tabular structures, and enhancing integration frameworks for improved performance. |
| 5.Provide recommendations | Recommendations emphasize thorough SP optimization prior to migration, strategic schema redesign for enhanced query performance, and robust integration planning to ensure data consistency across systems. |
| | Importance of clear planning, stakeholder collaboration, and iterative testing to mitigate risks and maximize the benefits of cloud-based BI infrastructures. |

Table 5.1 summarizes the main findings and outcomes based on the objectives outlined in the migration project. Each objective was analyzed to understand its impact on BI infrastructure modernization, highlighting both successes and challenges encountered throughout the process.

# 6. CONCLUSION

The migration project, aimed at upgrading the client's BI infrastructure and reducing costs, encountered significant challenges that potentially compromised its overall success. Opting for a serverless architecture, driven by cost-saving intentions, resulted in increased processing times and integration complexities, underscoring the critical need for meticulous planning and optimization strategies. Furthermore, the project underscored the importance of reassessing existing logic and structures before initiating new BI initiatives; adopting a parallel execution approach for Stored Procedures could have notably enhanced operational efficiency. Similarly, redesigning tabular structures with standardized schemas might have improved query performance and data accessibility, laying a more robust groundwork for future BI expansions.

Looking forward, recommendations for future projects include optimizing current processes, exploring alternative data management strategies, and integrating historical data into platforms like SAP to enhance data processing capabilities. Reflecting on lessons learned, defining project scopes more rigorously, and assessing migration implications comprehensively would have mitigated unexpected challenges and implementation delays. Despite initial setbacks associated with architectural choices, particularly in terms of overcoming inherent limitations, it's important to recognize the context in which decisions were made. The chosen solution, while posing challenges, was necessary in initiating the project and aligning with the client's budget constraints, surpassing other proposed alternatives.

In conclusion, the migration project serves as a valuable learning experience, emphasizing the importance of planning, optimization, and adaptability in facing complex BI projects. By identifying these challenges and working on the lessons learned, the clients can position themselves for sustained success and innovation in this evolving digital workplace. Through collaboration, innovation, and continuous improvement, the client can reach the full potential of modern BI technologies to drive sustainable business value and competitive advantage.

# 7. REFERENCES


Alsufyani, R., & Chang, V. (2015, November). Risk analysis of business intelligence in cloud computing. In 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom) (pp. 558-563). IEEE. DOI:10.1109/CloudCom.2015.84

Blažić, G., Poščić, P., & Jakšić, D. (2017, May). Data warehouse architecture classification. In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1491-1495). IEEE. DOI:10.23919/MIPRO.2017.7973657

Bryteflow (2024). SAP Data Replication: CDS Views in SAP HANA and how to create one. https://bryteflow.com/cds-views-in-sap-hana-and-how-to-create-one/#:~:text=Core%20Data%20Service%20(CDS)%20Views,with%20SAP's%20new%20programming%20model.

Chairungruang, S., Khampuong, P., Rodcharoen, P., Leelitthum, C., & Eak-ieamvudtanakul, P. (2022). Business Intelligence for Data-Driven Decision-Making in Vocational Education. International Journal of Educational Communications and Technology, 2(2), 61-69.

Choudhary, P. K. (2023, August). What is a Cursor in SQL? And when to use a Cursor?. C# Corner. https://www.c-sharpcorner.com/UploadFile/f0b2ed/cursors-in-sql/#:~:text=A%20cursor%20in%20SQL%20is%20a%20database%20object%20stored%20in,one%20row%20at%20a%20time.

Cisco (2024). What is Shadow IT?. https://www.cisco.com/c/en/us/products/security/what-is-shadow-it.html

Curry, E., Cavanillas, J. M., & Wahlster, W. (2016). New horizons for a data-driven economy: A roadmap for usage and exploitation of big data in Europe. Springer Open.

da Costa, P. J. P., & Da Cruz, A. M. R. (2012). Migration to windows azure–analysis and comparison. Procedia Technology, 5, 93-102. DOI:10.1016/j.protcy.2012.09.011

Databricks (2024, March). What is Databricks?. https://docs.databricks.com/en/introduction/index.html#:~:text=and%20IP%20control.-,What%20is%20Databricks%20used%20for%3F,from%20BI%20to%20generative%20AI.

Databricks (2024a). What is Parquet?. https://www.databricks.com/glossary/what-is-parquet#:~:text=What%20is%20Parquet%3F,handle%20complex%20data%20in%20bulk.

Databricks (2024b). What is a Snowflake Schema. https://www.databricks.com/glossary/snowflake-schema#:~:text=A%20snowflake%20schema%20is%20a,data%20marts%2C%20and%20relational%20databases.

Deloitte. (2023a). Founder's Day and 10th Anniversary of Our Wuhan Office. Deloitte. https://www2.deloitte.com/cn/en/pages/about-deloitte/articles/founders-day-n-10th-anniversary-of-our-wuhan-office.html

Deloitte. (2023b). About Deloitte. Retrieved Month Day, Year, from https://www2.deloitte.com/uk/en/legal/about-deloitte.html

Deloitte. (2023c). Digital Controls, Data Analytics. Deloitte. https://www2.deloitte.com/it/it/pages/risk/solutions/digital-controls--data--analytics---deloitte-italy---risk-advisory.html

Diamond, S. (2022). Migrating from a Data Warehouse to a Data Lakehouse: Databricks Special Edition. For Dummies®. John Wiley & Sons, Inc.

Ellison, M., Calinescu, R., & Paige, R. F. (2018). Evaluating Cloud Database Migration Options Using Workload Models. Journal of Cloud Computing, 7, 1-18. DOI: https://doi.org/10.1186/s13677-018-0108-5

ElMalah, K., & Nasr, M. (2019). Cloud business intelligence. International Journal of Advanced Networking and Applications, 10(6), 4120-4124. DOI:10.35444/IJANA.2019.100612

Ereth, J., & Baars, H. (2015). Cloud-based Business Intelligence and Analytics applications–business value and feasibility.

Errami, S. A., Hajji, H., El Kadi, K. A., & Badir, H. (2023). Spatial big data architecture: From Data Warehouses and Data Lakes to the LakeHouse. Journal of Parallel and Distributed Computing, 176, 70-79. DOI:10.1016/j.jpdc.2023.02.007

Fernandes, F., Correia, J., & Pontes, A. (2023, June). Business Intelligence: Trends and the Impact of Emerging Technologies. In 2023 18th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE. DOI: 10.23919/CISTI58278.2023.10211645.

Follis, K., Inbar, P., Coulter, D., Mabee, D. (2022, November). Tabular modeling overview. Microsoft. https://learn.microsoft.com/en-us/analysis-services/tabular-models/tabular-models-ssas?view=asallproducts-allversions

Gailey, G., Wang, W. W., Almeida, T., Skwiers-Koballa, D., Burns, E., Penmatsa, S., Eastbury, W., Shoemaker, C., Urnun, M., Gottipalli, S., Kracher, A., Brockschmidt, K., Dykstra, T., Appel, R., Price, A., Clebsch, S., McSwain, W., Fowler, C., Hei, D., (…), Henderson, M. (2023, May 24). Azure Functions overview. Microsoft. https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview?pivots=programming-language-csharp

Gawande, P., Burchel, J., Hedge, N., Stevenson, A., Gries, W., Xu, Y., Yu, S., Campise, K., Wang, L., Rabeler, C., Jenks, A., Ghanayem, M., Zurita, M., Lo, S., Laudenschlager, D., Pelluru, S. (2023, October 8). Copy activity in Azure Data Factory and Azure Synapse Analytics. Microsoft. https://learn.microsoft.com/en-us/azure/data-factory/copy-activity-overview

Gillis, S., & Bigelow, J. (2023). MPI: A Message-Passing Interface Standard (Version 4.1). University of Tennessee, Knoxville, Tennessee. https://www.mpi-forum.org/docs/mpi-4.1/mpi41-report.pdf

Google Cloud (2024, March). Deploy Dataflow pipelines. https://cloud.google.com/dataflow/docs/guides/deploying-a-pipeline#:~:text=Dataflow%20uses%20your%20pipeline%20code,operations%2C%20such%20as%20data%20aggregations.

Gupta, B., Mittal, P., & Mufti, T. (2021, March). A review on amazon web service (aws), microsoft azure & google cloud platform (gcp) services. In Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India. DOI:10.4108/eai.27-2-2020.2303255

Harby, A. A., & Zulkernine, F. (2022, December). From Data Warehouse to Lakehouse: A Comparative Review. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 389-395). IEEE. DOI:10.1109/BigData55660.2022.10020719

Hlupić, T., Oreščanin, D., Ružak, D., & Baranović, M. (2022, May). An overview of current data lake architecture models. In 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 1082-1087). IEEE. DOI: 10.23919/MIPRO55190.2022.9803717.

Hunges, A. (2024). Stored Procedure. TechTarget. https://www.techtarget.com/searchoracle/definition/stored-procedure#:~:text=A%20stored%20procedure%20is%20a,and%20shared%20by%20multiple%20programs.

IBM (n.d.). What is a data mart?. https://www.ibm.com/topics/data-mart

IBM. (2023). What is a data mart?. IBM. https://www.ibm.com/topics/data-mart

Imperva (2024). Data Lineage. https://www.imperva.com/learn/data-security/data-lineage/#:~:text=Data%20lineage%20is%20the%20process,Data%20lineage%20process

Incontra. (2023). What are data connectors?. https://www.incorta.com/glossary/what-is-data-connector#:~:text=Data%20connectors%20are%20software%20components,data%20science%20applications%20and%20frameworks.

Inmon, W. H. (2016). Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump. Technics Publications.

Jadhav, M. (2021). Data Migration From On Premise Oracle Database To SQL Manage Instance On Azure Cloud Using Azure Data Factory-A Working Approach. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12), 3524-3528. DOI: https://doi.org/10.17762/turcomat.v12i12.8093

Landutama, J. F., & Chowanda, A. (2023). Applied Design Thinking for Kimball Lifecycle to Improve Business Intelligence Dashboard Usability. International Journal of Innovative Computing, Information and Control, 19(4), 1139-1152.

Liew, Z. (2021, May). Understanding And Using REST APIs. https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/

LogicMonitor (2022, November). What is Azure Blob?. https://www.logicmonitor.com/blog/what-is-azure-blob#:~:text=What%20is%20Blob%20Storage%3F,like%20text%20or%20binary%20data).

Marr, B. (2023, September 21). The Role of Data Storage in Accelerating Time-to-Insights. LinkedIn. https://www.linkedin.com/pulse/role-data-storage-accelerating-time-to-insights-bernard-marr/

Microsoft (2024, May). Azure Functions documentation. https://learn.microsoft.com/en-us/azure/azure-functions/

Microsoft (January 2023). Automatic tuning in Azure SQL Database and Azure SQL Managed Instance. https://learn.microsoft.com/en-us/azure/azure-sql/database/automatic-tuning-overview?view=azuresql

N. Taylor, R., Medvidovic, N., & M. Dashofy, E. (2008). Software Connectors. University of California, Irvine. https://ics.uci.edu/~taylor/classes/221/07_Software_Connectors.pdf

Nambiar, A., & Mundra, D. (2022). An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management. Big Data and Cognitive Computing, 6(4). DOI:10.3390/bdcc6040132

Olariu, F., & Alboaie, L. (2023). Challenges in Optimizing Migration Costs from On-Premises to Microsoft Azure. Procedia Computer Science, 225, 3649-3659. DOI:10.1016/j.procs.2023.10.360

Oracle (2021). Data Integration Services. https://ocacs.ocs.oraclecloud.com/

Oracle (2024a). What Is a Data Catalog and Why Do You Need One?. https://www.oracle.com/it/big-data/data-catalog/what-is-a-data-catalog/#:~:text=Simply%20put%2C%20a%20data%20catalog,support%20data%20discovery%20and%20governance.

Oracle (2024b). Database Utilities: External Tables Concepts. https://docs.oracle.com/en/database/oracle/oracle-database/12.2/sutil/oracle-external-tables-concepts.html#:~:text=The%20external%20tables%20feature%20is,indexing%20of%20the%20staging%20table.

Peachman, R. R. (2023, August 16). Meet The World's Best Management Consulting Firms 2023. Forbes. https://www.forbes.com/sites/rachelpeachman/2023/08/16/meet-the-worlds-best-management-consulting-firms-2023/

Rodzi, N. A. H. M., Othman, M. S., & Yusuf, L. M. (2015, October). Significance of data integration and ETL in business intelligence framework for higher education. In 2015 International Conference on Science in Information Technology (ICSITech) (pp. 181-186). IEEE. DOI: 10.1109/ICSITech.2015.7407800.

Sarmah, S. S. (2018). Data migration. Science and Technology, 8(1), 1-10. DOI: Sarmah, S. S. (2018). Data migration. Science and Technology, 8(1), 1-10.

Sawadogo, P., & Darmont, J. (2021). On data lake architectures and metadata management. Journal of Intelligent Information Systems, 56, 97-120. DOI: https://doi.org/10.1007/s10844-020-00608-7

Siddiqa, A., Karim, A., & Gani, A. (2017). Big data storage technologies: a survey. Frontiers of Information Technology & Electronic Engineering, 18, 1040-1070. DOI: https://doi.org/10.1631/FITEE.1500441

Strohbach, M., Daubert, J., Ravkin, H., & Lischka, M. (2016). Big data storage. New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe, 119-141. https://doi.org/10.1007/978-3-319-21569-3_16

Tabular modeling overview. (2022, December 10). Microsoft Learn: https://learn.microsoft.com/en-us/analysis-services/tabular-models/tabular-models-ssas?view=asallproducts-allversions

Vailshery, L. S. (2023, December 7). Public cloud application infrastructure services market 2015-2024. Statista. https://www.statista.com/statistics/505248/worldwide-platform-as-a-service-revenue/

W3schools. (2024). Web APIs – Introduction. https://www.w3schools.com/js/js_api_intro.asp#:~:text=A%20Web%20API%20is%20an,functionality%20of%20a%20web%20server.

Yevge, A., Ghag, P., Solanki, C., & Mishra, A. (2022). Review Paper on Cloud Service Provider–AWS, Azure, GCP (No. 7440). EasyChair.

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa