Designing a Data Engineering lifecycle for the Football analytics on Cloud

Dissertation: MSc Data science

Author: Aidin Miralmasi

Date: August 2024

Student ID: 23053472

Department of Computer Science and Technology

Word count: 1013 (body word count: 7911 (from chapter 1 to the end of chapter 5))

Acknowledgments

First and foremost, I would like to express my deepest gratitude to God for His blessings and guidance throughout this journey.

I am profoundly thankful to my friends and family, especially my father for their unwavering support, encouragement, and love. Their belief in me has been a constant source of motivation.

I would also like to honor and remember my late mother, whose memory continues to inspire me and whose influence remains an integral part of my life.

I am especially appreciative of my supervisor, Professor Ahsan Kazmi, for his invaluable guidance, support, and expertise. His mentorship has been crucial in shaping this work, and I deeply appreciate his dedication and encouragement.

I would also like to extend my heartfelt thanks to the University of the West of England, Bristol, for providing me with a stimulating and supportive academic environment. The resources and opportunities available at UWE Bristol have significantly contributed to my growth and the success of this endeavor.

I extend my heartfelt thanks to the United Kingdom for welcoming me and providing me with an enriching environment to further my studies and personal development. The opportunities to study and shape my career in this lovely city, among such wonderful people, will always be cherished in my sweetest memories.

Additionally, I am proud and grateful to my lovely home country, Iran, and its eternal Persian Gulf, for contributing to my growth and shaping my character and personality. The foundation provided by my homeland has been fundamental to my journey.

Thank you all for your profound contributions to my academic and personal growth.

Abstract

Football has emerged as the most profitable sport globally, driven by its immense popularity and substantial media coverage of international games. The integration of big data and cloud computing is reshaping various industries, including sports, where these technologies are becoming crucial for enhancing performance and profitability. This paper aims to bridge the gap between football and technology by presenting a comprehensive data engineering lifecycle designed to benefit football enthusiasts, whether for professional or personal interests.

The proposed system focuses on three primary goals: effective data collection, optimization of infrastructure for hosting and operations, and the establishment of data sharing protocols to ensure accessibility and usability. Leveraging AWS cloud services, the system documents and reports data to optimize performance and quality in football.

The study starts with defining its research objectives and reviewing related works from other sports industries. It then details the methods and approaches used, followed by an evaluation of the system's strengths, weaknesses, and overall performance. The paper concludes with recommendations to address any identified shortcomings and enhance the system further.

By following best practices, this dissertation explores the potential of integrating technological advancements in football to elevate the quality and efficiency of data management in the sport.

2 minute video of abstraction review available from :
https://uwe.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=15829ce4-fab5-40a0-8c60-b1cd003625ec

Table of Contents

1. Introduction

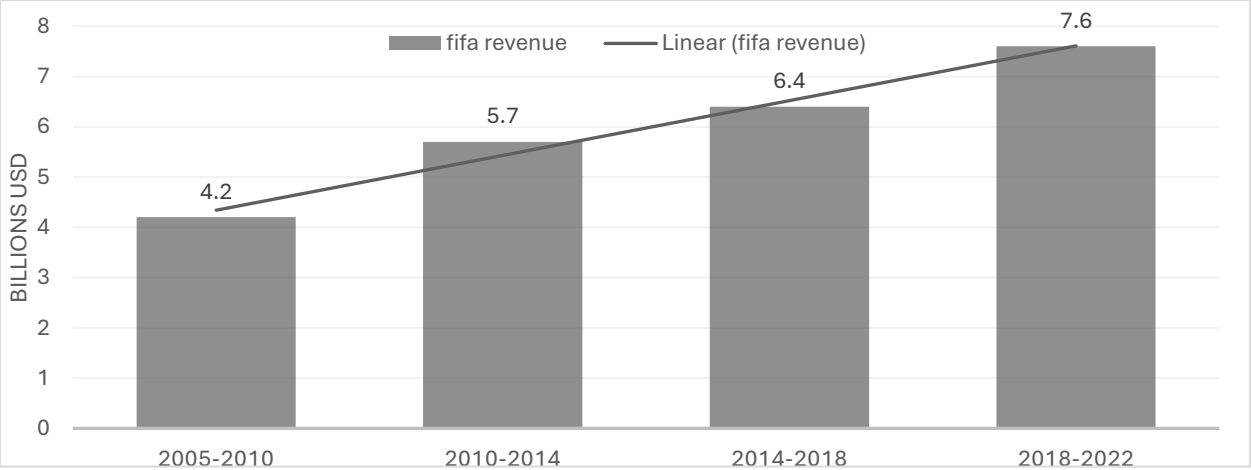
In this chapter, simply, the purpose of the paper is discussed. The chapter is sequentially organized as follows: Initially, background and the problem are raised, followed by motivation. Subsequently, research questions are mentioned. Finally, the structure of the paper is outlined.

1.1 Rational

1.1.1 Revenue

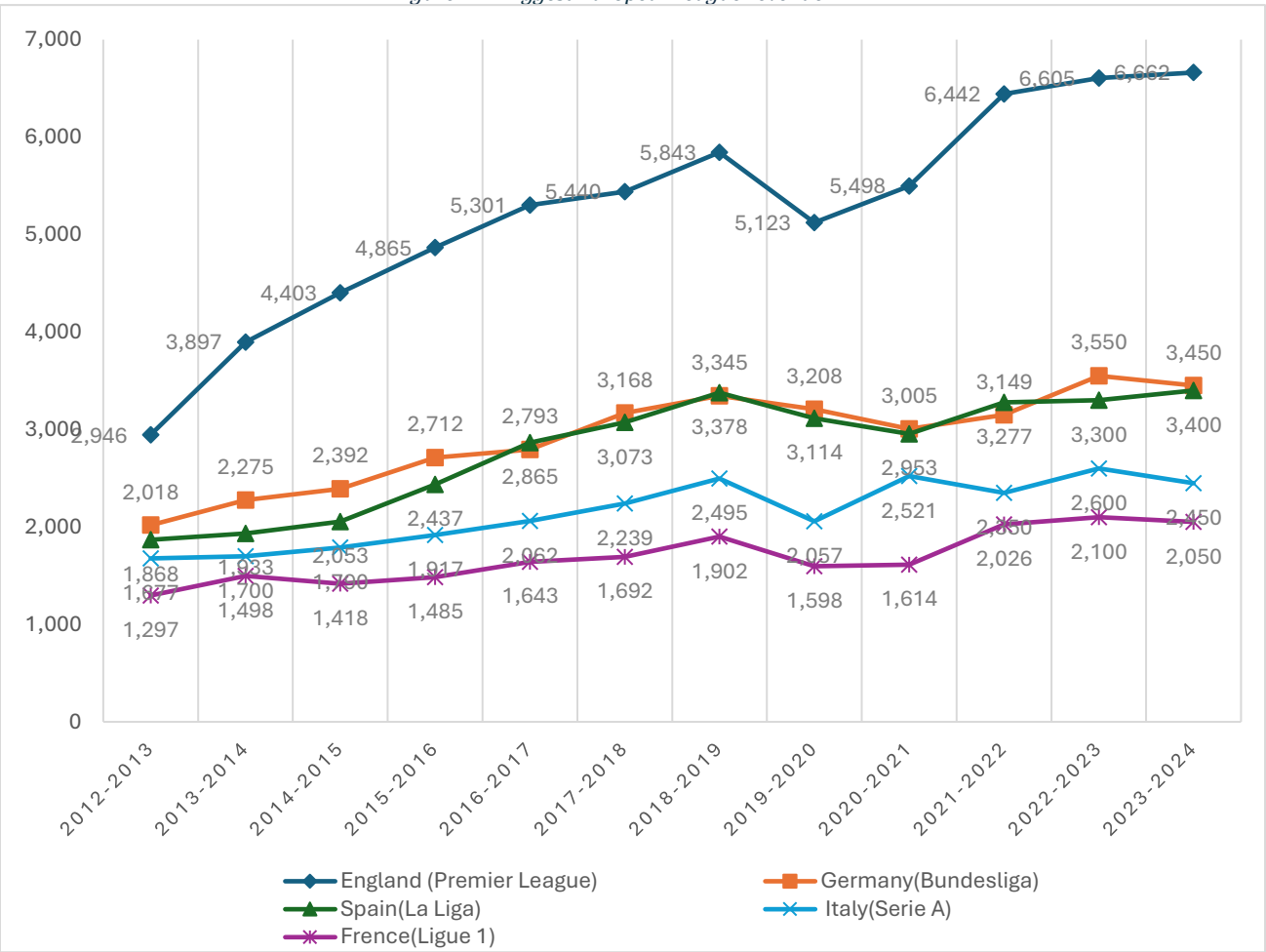
According to a Deloitte report, the Premier League generated approximately €6.6 billion in revenue, while La Liga generated about €3.4 billion, and Ligue 1 around €2 billion during the 2023 season. Additionally, FIFA statistics indicate that the 2022 Qatar Football World Cup generated $6.4 billion, which is significant given the short period compared to entire seasons of European leagues like La Liga or the Premier League.

Figure 1: FIFA revenue on 4 years period in billion USD



(FIFA, 2018; FIFA, 2022)

Figure 2: 5 Biggest European league revenue

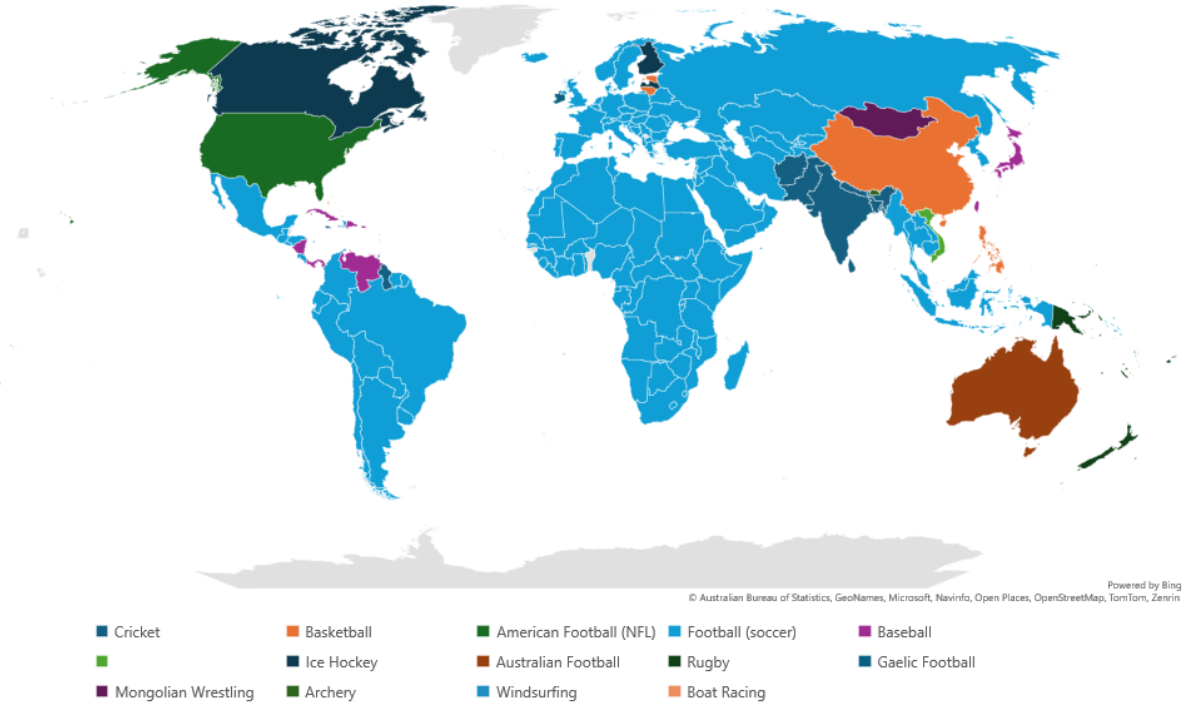


(Deloitte, 2023)

1.1.2 Fans and Society

Local communities are integral to mass sports, which are non-profit and aim to increase people's participation. Authors in the Sport Management Review cite the example of Germany in 2011, where there were more than 90,000 clubs and over 27 million people engaged in sports programs (Wicker, P., et al., 2012). Additionally, according to the World Population Review website, football (soccer) is the most popular sport. The map on the site indicates that soccer is the most favored sport in the most countries. Furthermore, based on FIFA data, the Qatar World Cup 2022 saw more than 3 million tickets sold, with an average stadium occupancy rate of 96%.

Figure 3: most favorite sport in the world



(world population review,2021)

Currently, as I write this dissertation, the Euro Cup is underway. Cafés are bustling with football fans, and numerous programs on the internet and TV are analyzing data alongside football coaches. This environment has inspired me to develop a system that benefits both individuals and clubs alike.

1.1.3 Academic and Scientific Approach

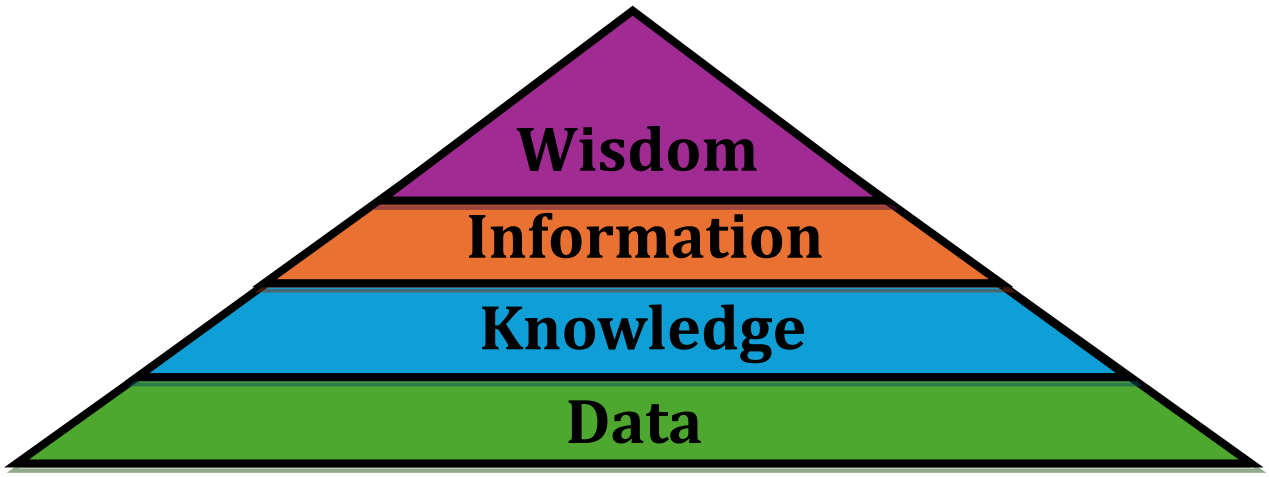
The sports industry is enormous. In 2022, there were about 500 sports management programs in the USA, 30 in Europe, 20 in Australia and New Zealand, and about 30 in the rest of the world (Shonk, D.J. and Weiner, J.F., 2021; NASSM, 2020). Recent innovations in data science and advancements in infrastructure have led to significant evolution and expansion in many fields such as healthcare, retail, manufacturing, cybersecurity, banking, and numerous other areas. The sports industry is no exception, with companies increasingly adopting data-driven approaches to decision-making. (Thakkar and Shah, 2021).

The Oakland Athletics baseball team in 2002 was the first team to implement a data-driven approach (Thakkar and Shah, 2021). In data science, this approach can be implemented in real-time processing for large-scale analysis. For example, in the FIFA World Cup, data can be extracted using wearable devices for movement analytics, or machine learning applications can be used to analyze ticket sales to increase revenue (Xiao et al, 2023).

1.2 state of problem

In today's world, data is continuously produced in various forms such as visual and auditory formats (pictures, videos, voices), as well as traditional formats like text and numbers. Raw data, in its initial state, lacks inherent meaning and value but can be transformed into valuable insights or wisdom through careful management of data quality (Baskarada and Koronios, 2013)

Figure 4: the pyramid of wisdom



(Baskarada and Koronios, 2013)

Data mining plays a critical role in this process by revealing patterns and relationships within data, thereby turning it into meaningful and valuable information. This emphasizes the need to design systems that gather and store data in a desired format, optimized for analysis (Che et al., 2013). However, raw data is inherently unreliable as it may contain noise or inconsistencies. Poor data quality, originating from various sources, significantly impacts data mining applications accuracy (Alasadi and Bhaya, 2017).In the realm of football analytics, matches from major European leagues are streamed continuously each season. It falls upon data scientists to develop applications adept at merging data from diverse sources into a cohesive format suitable for analytics.

1.3 Aims and Scope

1.3.1 objectives

The objective is to manage, format, collect, merge, and store data in a database that allows others to query and retrieve relevant data, ensuring high data quality throughout the process.

1.3.2 research questions

For this goal, several critical questions arise:

1. **How to extract or generate data ensure data quality?**
2. **How to store data and manage infrastructure?**
3. **How to provide access to data?**

1.4 paper structure

We have discussed the justification of the topic. In the next chapter, we will review the related works. In Chapter 3, we will discuss the methodology. In Chapter 4, we will evaluate the results and comment on performance. Finally, in chapter 5 we will see how we can improve or recommend the model.

1.5 Code Accessibility

The code referenced in Chapters 3 and 4 of this dissertation is available on my GitHub repository. The repository, named Dissertation, includes the following files:

- **Index.html:** The front-end code for the web application.
- **Index.zip:** A zip archive of the web application.
- **Test.py:** A script used for performance testing, as discussed in Chapter 4.
- **Lambda_function.py:** The Python code for the Lambda function described in Chapter 3.
- **Lambda_function.zip:** A zip archive containing the Lambda function code and its dependencies.
- **Readme.md:** A markdown file with instructions for setting up and installing the code.
- **Data.zip file:** Contains raw data in Chapter 3 (the Readme.md file provides instructions for its use).
- **SQL file:** this is extracted database
- **Relationaldb-diagram.mwb:** the purposed diagram discussed in chapter 3

For access to these files and further details, please visit the GitHub repository at: https://github.com/Aidin1999/Dissertation.

2. literature review

This chapter reviews academic works on digital sports, focusing on football and basketball. It discusses various approaches, answers key questions, and critiques differences in methods and goals compared to our case study.

2.1 Data generation and management

2.1.1 wearable devices

In this context, there are multiple ways of data generation and gathering. **Xiao et al. (2023)** in their paper "**Review on the Application of Cloud Computing in the Sports Industry**," believe that wearable devices can be a good way of collecting data. The paper explores methods such as using wearable devices for players to track their performance, which can be monitored and analyzed in real-time with cloud infrastructure. Moreover, it discusses the less-explored area of fan data or fan engagement, highlighting how organizations like FIFA or the Olympics can use fan data for better marketing analytics. However, this paper does not delve into how this method can be organized.

This idea is supported and implemented by **Raković and Lutovac (2015)** in their paper "**A Cloud Computing Architecture with Wireless Body Area Network for Professional Athletes Health Monitoring in Sports Organizations – Case Study of Montenegro.**" This paper aims to record and monitor players' activities to prevent injuries using a case study in Montenegro. It employs Wireless Body Area Networks (WBAN) and Wireless Sensor Networks (WSNs) within a cloud infrastructure, providing detailed commentary on the data management of their system. Additionally, it suggests improving data timeliness by transferring data between clubs when a player moves. Overall, it offers in-depth analytics.

2.1.2 Artificial intelligence

Li and Huang (2023) in the paper "**A Comprehensive Survey of Artificial Intelligence and Cloud Computing Applications in the Sports Industry**" also support this idea. They highlight the benefits for player injuries and health, noting that these technologies can prevent injuries by changing players before an accident occurs. Moreover, it adds other ways of data gathering, mentioning that chatbots and social media reviews, with the help of Natural Language Processing (NLP), can extract data on fan engagement.

Chomątek and Sierakowska (2021) also used cognitive AI in their paper "**Automation of basketball match data management**" They suggested using OCR for data gathering and generation. This paper aims to design a system for Polish women's basketball teams to easily extract and manage data. They explain that the usual method of extracting data involved writing down data on tables and saving the tables as pictures. They introduce a new way of extracting data by using AI to read the information on pictures. They use neural

networks with the aim of reaching an accuracy of 70 percent or more. Their application is an OCR (Optical Character Recognition) application that reads data from handwritten notes or signs through pictures.

2.1.3 web scraping

Another method of data gathering could be web scraping and getting data from different sources and saving them in the desired form. **Cao (2012**) in the paper "**Sports Data Mining Technology Used in Basketball Outcome Prediction**" concerns data mining methods for basketball outcome prediction using machine learning applications. The paper used web scraping methods to collect data from different sources, mainly basketball-reference.com. It utilized open-source websites to combine data and achieve high-quality results. The Ruby language with Water WebDriver was the tool used for data collection. The author explained that data was extracted from different sources, cleaned by deleting unnecessary columns, and then merged into a CSV file for different entities.

2.1.4 data inserting

Besides the mentioned methods, which include AI through OCR and NLP or computer vision or wearable devices or even web scrapping, **AL-ASADI et al. (2018)** suggest a more convenient and traditional way. In their paper "**An Online Information System for Football Club Management**," they suggest that data is gathered by staff, either coaches, staff managers, or their assistants. This paper aims to design and implement an information system for managing the data of a football club. It focuses on health, operational, and competition data, and how they are stored and managed. They categorized the data into five databases: gaming, staff, competition, administration, and extra. Although they did not emphasize SQL or NoSQL databases, they illustrate a relational database with possible entities. The data is managed by admins who can retrieve data for different purposes. The data is produced by different sources such as club staff, managers, and players. Moreover, they claim benefits like providing doctors with secure records for players, allowing players to track their performance and penalties, and enabling coaches to perform analytics. They mentioned that all data is stored in a data warehouse and queried using IDs.

2.2 Data storage and system infrastructure

Efficient data management is crucial for deriving valuable insights from the vast amount of data generated in the sports industry. Various studies have explored methods to collect, manage, and analyze sports data using different technologies and approaches.

2.2.1 Relational Database Management Systems

Atasoy and Özer (2021) in the paper "**Database Knowledge Management in Sport**," focus on extracting knowledge from data through knowledge management. They emphasize the necessity of organizing data amid the information explosion, with definitions of data,

information, knowledge, and wisdom, concerning data-based sport managing systems. The paper justifies the importance of data management by advocating for relational models due to their reliability and simplicity. It highlights various types of data in sports, such as performance, competition, and management data. Examples of common SQL databases like MySQL and Microsoft SQL are provided.

Horvat et al (2019) implement the use of MySQL in the paper "**Data-Driven Basketball Web Application for Support in Making Decisions.**" The authors utilized and managed data with a relational database, emphasizing the querying of data in different ways, such as by time periods or different teams. This method offers a more convenient and flexible way of processing and querying data. The web application enhances reliability and allows coaches to easily access data. The web app connects to MySQL, making it easily accessible.

Wong (2001) also used SQL databases for baseball. While the paper emphasizes relational databases, the author adds more detail, such as setting IDs. In the paper "**Implementing a Database Information System for an Electronic Baseball Scorecard**," it discusses an information storage system for an electronic baseball scoreboard. The paper suggests a SQL database matched with a graphical user interface. It proposes that coaches can use this for data-driven analytics to monitor players' performance, and fans can also track their favorite teams. The paper emphasizes SQL databases and relational models. It categorizes the data into five tables, such as game and game event tables, and joins data with specific IDs for each record. It also prefers MySQL for its free mode and limitless data saving. The system is event-driven, meaning that new data records are added to the electronic board during changes or events. Overall, they provide a system that users can query through different tables, benefiting both coaches and fans.

Cao, C., 2012 in the mentioned paper, MySQL was used for its ease of querying and merging data, and the cloud infrastructure AWS was used for this goal. It was mentioned that the EC2 instance was the preferred choice. The use of cloud infrastructure will be discussed further.

2.2.2 NoSQL Database Management Systems

On the other hand, NoSQL could provide distinct advantages for different requirements

Hafizul et al (2019) have a different approach. In the paper "**Data Management System for A Series of Kinematic Movement in Football Analytics**," they discuss the right format of data that should be used, focusing on document-oriented databases for real-time analytics. It offers a cloud solution and compares cloud databases to MySQL databases, stating that due to accessibility and flexibility, cloud options are better for real-time analytics, which require higher processing capabilities and lower latency. The paper also covers methods for a web application to connect to databases and display results, supporting the upload of data to web applications.

Similar to the last paper which used NoSQL databases in the cloud, **Foschi (2021)** also implements NoSQL databases for its goal. In the paper "**Design of a Data Management System for Value Bet Detection and Soccer Performance Analysis**," it concerns the extraction of betting value in football for AI applications. In this context, betting values mean higher chances of winning. The paper explores possible methods and choices, explaining SQL with ACID transactions and NoSQL with BASE transactions, ultimately choosing NoSQL for its scalability and flexibility. The NoSQL database allows for real-time processing. It used a graph database on AWS for scalability and agility. The infrastructure is hosted on AWS, using AWS services. The database used is Neo4j for this goal. IDs play a crucial role in this scenario, as each match played gets a new ID to show the new match. The paper also explained using EC2 for hosting Neo4j, S3 for storage, and Lambda functions for processing.

2.2.3 Cloud Infrastructure

Similar to mentioned papers, cloud infrastructure can be a good option for hosting the infrastructure. **Xiao et al. (2019)** in their report mentioned in the last heading discuss the potential benefits of cloud computing in the sports industry. It highlights strengths such as cost-effectiveness, scalability, and reliability. Additionally, it addresses critical points and challenges like data privacy, security, and dependency, offering solutions for these issues. The paper addresses where and how to host and store data, providing strong reasons for its recommendations. It suggests that cloud computing is a good option for processing and saving data due to its scalability, cost-effectiveness, availability, and simplicity. Similar to this, **Raković and Lutovac (2015)** also address data management by discussing hosting and saving data on the cloud and provide more details in terms of cloud architecture.

Li and Huang (2023) also advocate for using cloud infrastructure. They in the paper review the current applications of cloud computing and AI in sports, investigating their potential benefits. It explores different aspects of sports and the advantages that AI and cloud computing can bring. Like **Xiao et al. (2019)** they emphasize that, due to cloud scalability, availability, flexibility, and cost-effectiveness, the cloud is an excellent environment for implementing AI applications. The paper also addresses threats like data privacy, security, and transparency, citing that the NBA uses Azure, and the CBA uses Alibaba Cloud

2.3 Data sharing applications and protocols

After discussing the methods of data generation, storage, and ingestion, we can now delve into data sharing protocols and applications.

Atasoy and Özer (2021) argue that web-based databases ensure data quality management. They illustrate the significance of e-governance in managing sports clubs and

activities through a real scenario from Turkey. This highlights the importance of having a centralized, accessible platform for data management and sharing.

This concept was implemented by **Horvat et al. (2019)** in their cited paper, which aims to create an online information processing system web-based application to assist basketball coaches in decision-making. By leveraging web-based applications, they enable real-time access to data, making it easier for coaches to analyze and utilize data for strategic purposes.

Hafizul et al. (2019) also add more detail to this method. In their cited paper, they concern sharing data with users and providing an easier way to upload and download data. Their aim was to provide a web application with an architecture that handles user requirements. This involves creating an intuitive interface for users to interact with the database, ensuring that data can be easily uploaded, accessed, and downloaded as needed.

Overall, all these papers emphasize the importance of having a backend infrastructure connected to cloud infrastructure to facilitate data sharing. This backend infrastructure typically includes APIs (Application Programming Interfaces) that allow different applications to interact with the database, ensuring seamless data queries and updates. This approach ensures that users can effectively access and manage data, leveraging the scalability and reliability of cloud services to maintain data integrity and availability.

2.4 Chapter Summery

This chapter discusses academic works and answers to questions from the previous chapter through various research papers on digital sports, particularly football and basketball. The review includes papers with similar goals but differing approaches, solutions, and scopes, addressing key questions and comparing them to our case study.

Data generation methods such as wearable devices, AI, web scraping, and manual data entry by staff are explored. Technologies for managing large sports data volumes are examined, highlighting the pros and cons of SQL and NoSQL databases. SQL databases like MySQL are noted for reliability and simplicity, while NoSQL databases are valued for scalability and flexibility in real-time applications. Data sharing protocols and web-based applications for managing sports clubs are also discussed.

2.4.1 Preferred Methods

Data Generation: Due to limitations of wearable devices and AI solutions, an event-driven architecture with manual data insertion is chosen for accuracy and completeness. Post-match data ingestion is feasible, while machine learning is preferred for real-time data ingestion.

Data Infrastructure: Cloud infrastructure is recommended for scalability, cost-effectiveness, and reliability, with strategies including data redundancy, automated functions, and load balancers.

Database: SQL databases are ideal for event-driven architectures needing extensive querying and data quality, while NoSQL databases suit stream-based architectures with lower latency.

Sharing Protocols: A web-based information system is optimal, with detailed artifacts connecting the system to the infrastructure, such as an API gateway to the cloud.

Table 1: papers ordered by years of publication, recaps the papers reviewed

Author(s)	Year	Sport field	Data generation	Data infrastructure	Data sharing
Wong	2001	Baseball	-	Relational databases (MySQL)	-
Cao.	2012	Basketball	Web scraping	Relational databases (MySQL) On Cloud (AWS EC2)	-
Raković and Lutovac	2015	Football (Soccer)	WBAN and WSNs	Cloud	-
AL-ASADI et al.	2018	Football (Soccer)	Data insert	Data warehouses	-
Xiao, L et al.	2019	Broad fields	Wearable devices	Cloud	-
Horvat et al.	2019	Basketball	-	Relational databases (MySQL)	Web based application
Hafizul et al.	2019	Football	-	NO-SQL (Document Oriented) On cloud	Web based application
Atasoy and Özer.	2021	Broad fields	-	Relational databases (MySQL)	Web-based information system
Chomątek and Sierakowska.	2021	Basketball	Artificial Intelligence (Computer Vison)	-	
Foschi	2021	Football (Soccer)	-	Graph DB (noj4) On Cloud (AWS EC2, s3, lambda)	-
Li and Huang.	2023	NBA and CBA (basketball)	Wearable devices, NLP	Cloud	-

3. Methodology

Having explored other papers' solutions, we have preferred methods that align with our requirements. Now, we can proceed to implement these methods tailored to our specific needs.

3.1 Data Characteristics

The initial data is sourced from Kaggle and is licensed under the Open Data by Open Knowledge. The database is in SQLite format and aggregates data from various sources, including attributes from FIFA games by EA Sports. The database consists of seven entities:

- **Country (id, name):** Represents 11 European countries.
- **League (id, country_id, name):** Contains the leagues within these European countries.
- **Match (115 attributes including league, country, team IDs, and match details):** Records every match event and detail, containing data for 25,884 matches.
- **Player (7 columns including player id, name, height, weight, and birthday):** Details the physical attributes of players and has 11075 records.
- **Player Attributes (42 attributes including id of records and performance records):** Shows player performance for each season, such as finishing ability and overall record, containing about 184000 records.
- **Team (5 columns including team names and short names):** Lists team names with 299 records.
- **Team_Attributes (25 columns of all teams):** Displays team performance and attributes such as build-up play speed, speed class, and dribbling, with 1,458 records.

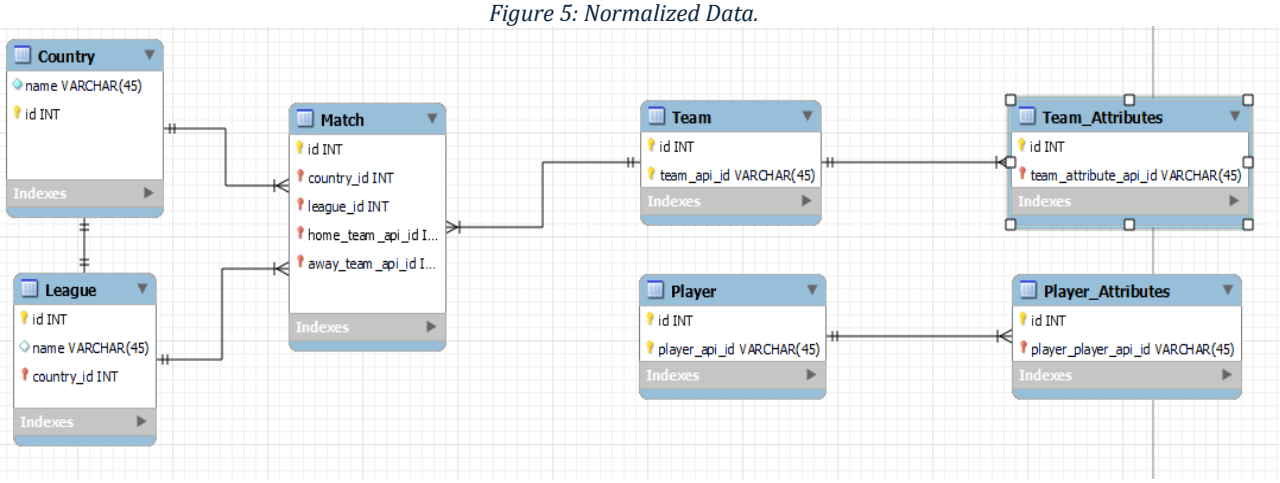
The SQLite file is read using DB Browser for SQLite. The script and data are extracted and, with minimal changes, set up in the data warehouse.

3.1.1 Normalization

In the process of data normalization, our goal is to structure the database to minimize redundancy and ensure data integrity. By utilizing unique identifiers (IDs) in normalization, we establish relationships between different entities such as matches, leagues, countries, teams, and players. Normalization involves breaking down the data into entities, achieving atomicity and consistency while avoiding redundancy. After identifying entities, we can define relationships that depict how data is interconnected. This approach proves beneficial and timesaving across various query methods like insert, update, or delete operations. Overall, these relationships illustrate dependencies and provide a clear data model, especially useful for queries and joins (Kumar and Azad, 2017).

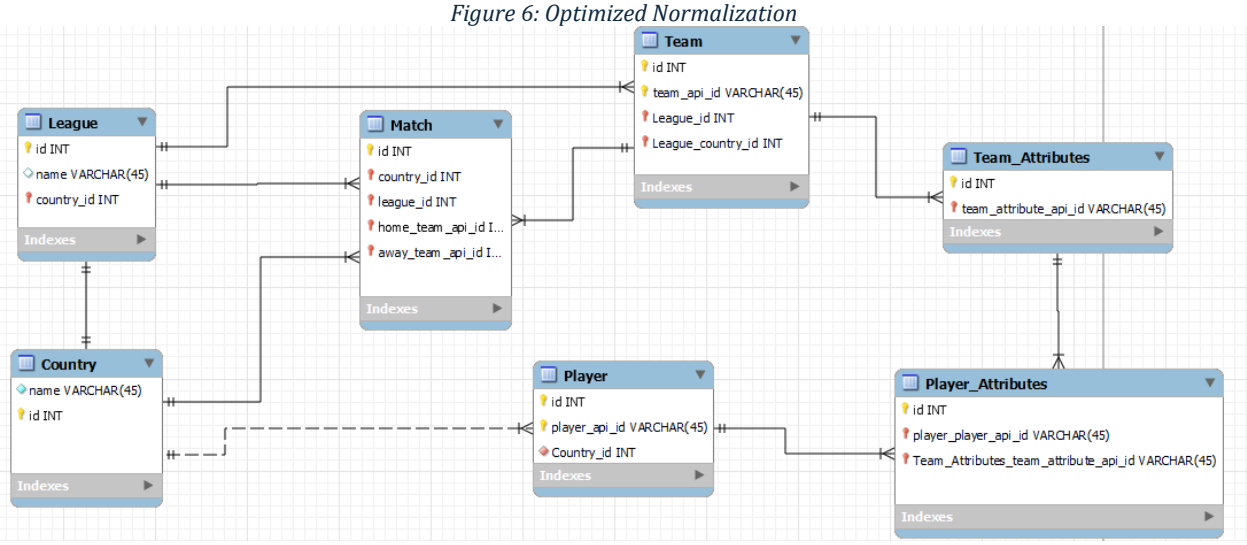
The diagram below, created using MySQL Workbench, exemplifies this third-level normalization

Figure 5: Normalized Data.



To enhance the relationships in the database schema, we will add a 'country_id' field to the Player table to link players nationality to establish a one-to-many relationship between Country and Player. The Team table will link teams to leagues through a 'league_id' field, creating a one-to-many relationship between League and Team. Additionally, 'Team_attrinute_id' was also added to Player Attributes to link players to teams. This setup allows for querying players by their country and teams, accommodating the fact that players can play in different European countries regardless of their nationality.

Figure 6: Optimized Normalization



3.2 Data quality

Data quality and governance are key points to success in any data lifecycle system. Batini et al. (2009) aim to enhance data quality and introduce methods to improve it. They suggest both data-driven and data-process strategies to improve quality.

Data-Driven Strategies

- **Acquisition of New Data:** Sourcing data from high-quality providers.
- **Standardization and Normalization:** Ensuring all data inputs conform to predefined standards.
- **Schema Integration:** Creating a unified schema to manage and access data efficiently.

Data-Process Strategies

- **Data Governance:** Establishing policies and procedures to manage data quality.
- **Reliability:** Ensuring data is trustworthy.
- **Cost Optimization:** Implementing cost-effective methods to maintain data quality.

and highlight several dimensions of data quality:

- **Accuracy:** Ensuring data values reflect the real-world entities they represent. accuracy refers to the correctness of the data.
- **Completeness:** completeness indicates the extent to which data attributes are fully populated. Minimizing missing values in data attributes.
- **Consistency:** Ensuring data logically aligns and coherence, such as age values being between 0 and 120.
- **Timeliness:** addresses the need for regular updates to maintain relevance Keeping data updated and relevant for current needs.

Sidi et al. (2012) support and expand upon the ideas presented by Batini et al. (2009) emphasizing the importance of data quality and more to this like Process-Driven Strategies which focusses on managing the data processing system to improve data quality:

- **Processing Control:** Implementing controls to monitor and manage data quality during processing.
- **Process Redesign:** Redesigning processes to improve data quality systematically.

They say data quality depends on customer and client needs They mention 32 quality dimensions and reference 15 dimensions from other papers we can refer most suitable one to our application as

- **Duplication:** Minimizing redundant data entries.

- **Sufficient Data Quantity:** Ensuring enough data is available to meet analytical needs.
- **Effectiveness and Usability (Relevancy):** Ensuring data is effective for its intended use and easy to utilize.

Haug et al. (2011) also expands the idea and categorize data into three types:

1. **Master Data:** Basic data that changes infrequently, such as customer names and addresses.
2. **Transactional Data:** Data related to transactions, such as sales.
3. **Historical Data:** Past data that is stored for reference and analysis.

They recommend a trade-off approach for investing in data quality improvements. This approach underscores the importance of balancing quality and cost, emphasizing that while some data quality improvements are essential, others may not provide sufficient benefits to justify the expense. The figure below illustrates the trade-offs between achieving decent quality data, the effects of poor-quality data, the profits from improving data quality, and the total costs involved.

Figure 7: The trade of accusing good data quality and profit and loss of bad data quality.

(Haug et al., *2011)*

3.2.1 Data Quality Enhancements

As we reviewed the papers, we can categorize data into different groups based on their importance:

- **Master Data**: This includes leagues, teams, countries, and players. These entities are the most important and foundational elements, and they are unlikely to change frequently.
- **Historical Data**: This category involves attributes of teams and players. These attributes are updated once a season and contain crucial details like overall performance.

- **Transactional Data**: This includes match data, which is updated after every match. While voluminous, this data is considered the least important of the three categories.

Data quality trade-off: As discussed earlier, the trade-off between data quality and cost can be optimized by adding new columns and IDs to each entity. This effort is worth the investment. For instance, adding connections between players, countries, and teams, as well as linking team attributes, can enhance data integration. Additionally, a "coaches" entity could be included and connected to its attributes, further enriching the dataset. While other data points, such as fan engagement (e.g., attendance at each match) or the nationality of coaches or managers, could be implemented, they may not be worth the additional effort. Similarly, defining a stadium entity with attributes like capacity, location, and name could be beneficial but might not provide sufficient value relative to the cost of implementation.

- **Timeliness**: The data spans from the 2007 to 2016 league seasons, making it quite outdated. This is the weakest point of quality. Improving timeliness would require updating the data by inserting records for the most recent games.
- **Completeness**: Master data quality is good, but historical data has some missing values, mostly under 10% (except for one attribute, dribbling, in team attributes). Match data, however, ranges from 0 to 45% null values.
- **Relevancy**: While the database contains detailed information on teams and players, the lack of normalized attributes reduces its efficiency.
- **Accuracy**: Data gathered from FIFA EA games is likely accurate, especially for master data. However, performance metrics can be subjective.
- **Duplications and Consistency**: Some IDs are missing but given the large size of historical and match data, this can be considered a minor issue. Overall, consistency is good.

Data-Driven Strategies: Data is sourced from EA games, a reliable source for team and player performance, and transaction data appears accurate despite some nulls. The data insertion process follows a schema managed by a data warehouse, ensuring consistency.

Data-Process Strategies: When data is inserted by experts and SQL constraints are in place, inconsistency issues are minimized. SQL constraints prevent problems such as duplicate IDs or incorrect data types. This approach is cost-effective compared to other data gathering methods.

- **Processing Control**: the system allows for monitoring and upgrading data flow through pipelines, akin to process control. Furthermore
- **Process Redesign**: as client bases expand, scaling the infrastructure becomes pivotal in managing data effectively.

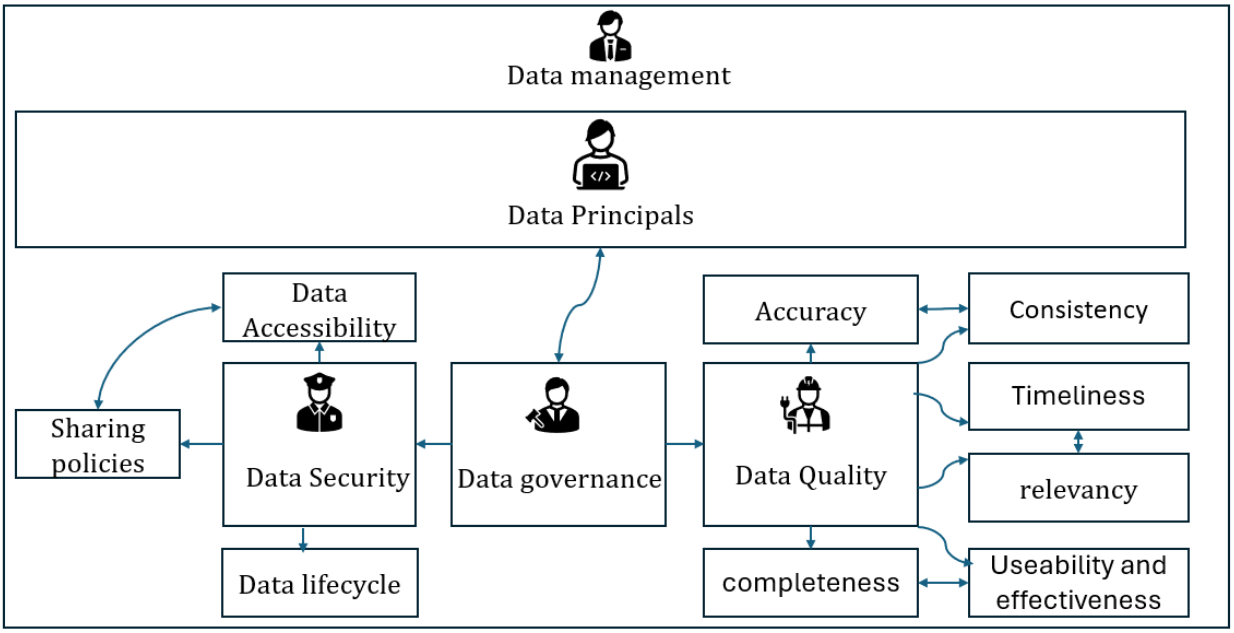
3.3 Data Governance

Data quality, data governance, and data management are closely related. Data management is the method for managing, including hosting, generating, and sharing data. Data governance involves the rules and actions that should be considered to ensure data quality, which is key for optimizing the system. Just like data quality, data governance does not have a unique definition and should be defined based on users' needs. Data governance is a framework for decision-making and duties for managing data, ensuring that data is in the right format. The main difference between data management and data governance is that governance focuses on the decisions and rules for managing data efficiently. The data principles are the goals, and other aspects ensure data principles are met. These aspects include data quality, metadata, data access, and the data lifecycle (Alhassan et al., 2016)

3.3.1 Data Governance implementation

In our frame, Data management overarching role involves data principles which is the goals or objectives for managing data and under that data governance. One aspect of governance is security, which includes data accessibility, sharing protocols, and the lifecycle. Another aspect is data quality. The figure below illustrates the management structure:

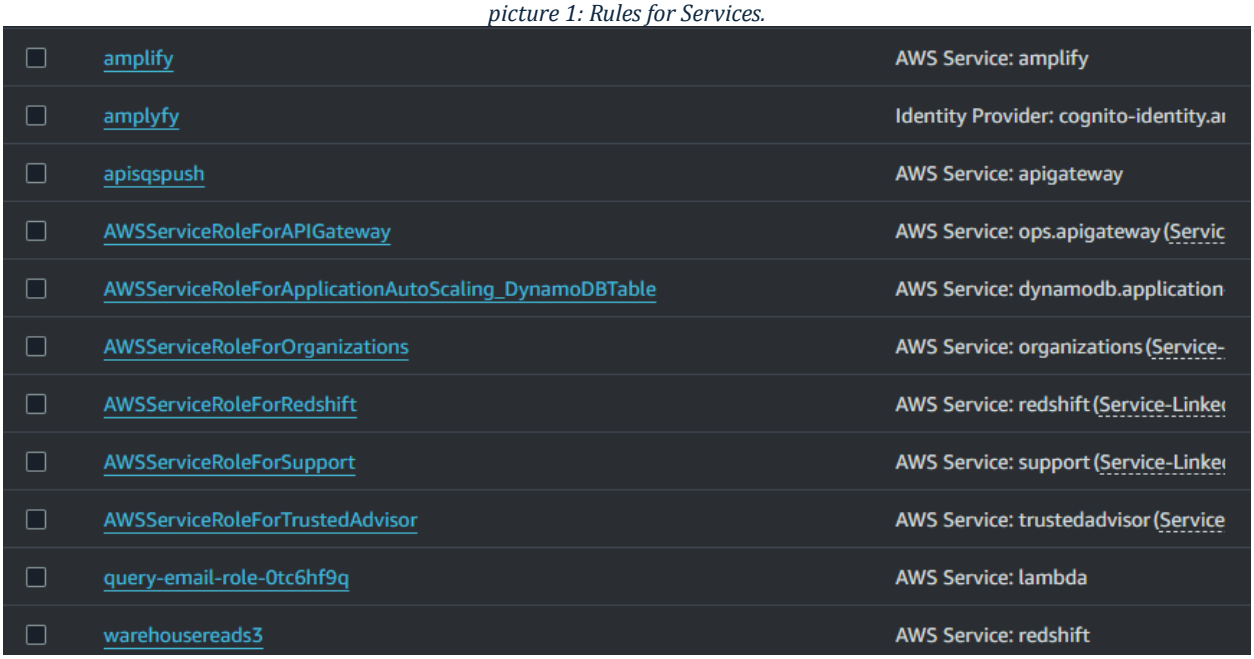
Diagram 1: The Data Management Diagram.



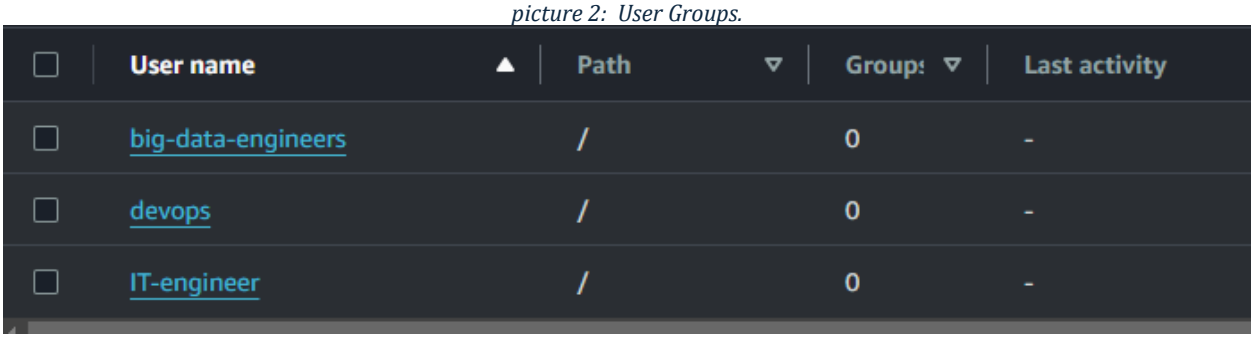
Data security is ensured through data governance rules, which stipulate that data accessed by users should only be accessible to them and retained for a specified period. Users should have access only to the specific data they request, ensuring they do not have indiscriminate access to all data. In this study case, these rules can be implemented for services and users. Lambda, a serverless service, handles backend requests, queries the data warehouse, and saves the results in a NoSQL database with query and message permissions. The data

warehouse has permissions for storage access to perform ETL processes. The system involves three key roles: a Big Data Engineer, who handles the data warehouse, NoSQL database, and storage; a DevOps Engineer, who manages the serverless functions; and IT Engineers, who handle APIs and web app configurations. These users have permissions only to perform their tasks. The VPC and security groups have rules that restrict access through the internet, allowing connections only from aligned apps.

picture 1: Rules for Services.



picture 2: User Groups.



3.4 Data management

Cloud computing can be beneficial for data storage and data transmissions. Cloud services can be referred to as SaaS, PaaS, and IaaS. The main goal of cloud computing is virtualization, which can be deployed in three modes: public cloud, private cloud, and hybrid cloud. Cloud computing can increase collaboration opportunities and reduce the cost of infrastructure (Masrom & Rahimli, 2014). Additionally, cloud services provide on-demand service and wide access due to their elasticity, allowing resources to be measured and monitored effectively. Cloud services also offer sharing pools for sharing with customers (Lupşe et al., 2012) In this case study, the data warehouse is a core component,

as suggested by literature reviews for hosting large datasets. The data warehouse, with its elasticity, supports on-demand access and sharing pools, and is ideal for OLAP (online processing applications) (Verma, 2013). Data warehouses are highly capable of ETL (extract, transform, load) from different sources, suitable for many applications, and can support backend processes for data-driven applications, data exploration, reporting, and more (Rehman et al., 2018). Furthermore, Function as a Service (FaaS) is a popular and effective serverless choice, reducing aggregation calls and workloads, and is perfect for debugging. For example, AWS Lambda is serverless, auto-scaling, resource-efficient, and ideal for ETL (Lynn et al., 2017). DynamoDB is a fast, scalable, and convenient data storage solution, perfect for storing data on SSDs. This NoSQL database is fast, predictable, and has efficient metrics like logs for monitoring purposes, making it a perfect match for API calls (Niranjanamurthy et al., 2014).

3.4.1 Key Components:

To provide a comprehensive view of the data architecture and its components, here's a refined and structured summary of the architecture and its elements, leveraging AWS for its advantages and services. This architecture aims to facilitate data sharing and convenience for users while ensuring high data quality. AWS is preferred for its cost-effective services, including a $300 free tier for Redshift and extensive free tier advantages.

1. **Data Warehouse (Redshift)**:
    - **Role**: Central component for handling and hosting large volumes of data, up to terabytes.
    - **Advantages**: More scalable and faster compared to traditional RDBMS.
2. **Serverless Functions (Lambda)**:
    - **Role**: Code execution for the system (IaaS).
    - **Advantages**: Serverless, meaning engineers do not manage servers; cost-effective for running code.
3. **API Gateways (API Gateway)**:
    - **Role**: Entry point for accessing the system (PaaS).
    - **Advantage**: Only accepts encrypted requests, preventing direct internet access to the backend and decouples the internet from the core backend services.
4. **Queue Service (SQS)**:
    - **Role**: Manages and orders incoming requests. Usually, connects requests to services, such as Lambda, efficiently (PaaS).

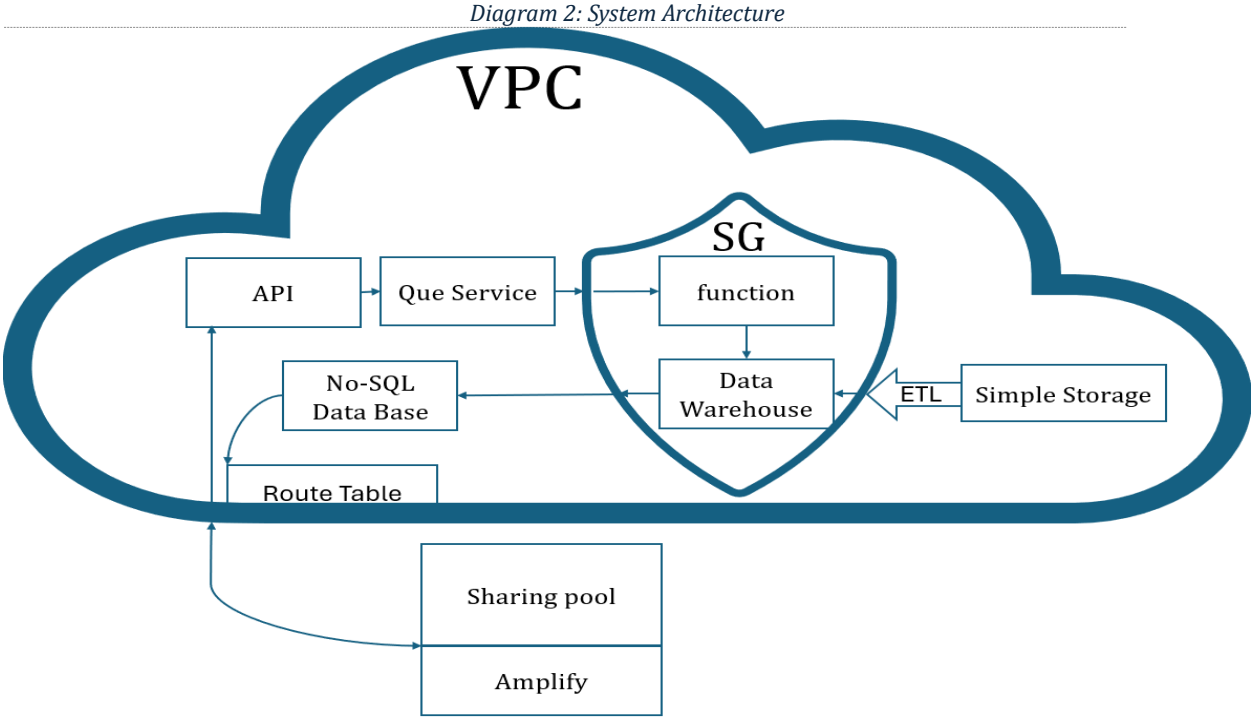
- **Advantages**: Serverless services decouple services and reduce the stress on functions and the warehouse.

5. **Web App and Sharing Pools (Amplify and Cognito Pools)**:
    - **Role**: User interaction platform. Sends requests through the backends such as API Gateway or data bases and retrieve the answer (PaaS).
    - **Advantages**: Effortless way to share data and simple access.
6. **NoSQL Database (DynamoDB)**:
    - **Role**: Stores queried results as key-value pairs usually use for frequent referred data (IaaS).
    - **Advantages**: Scalable, cost-effective, and more organized than simple storage.
7. **Simple Storage Service (S3)**:
    - **Role**: simple storage for infrequent data which is crucial to have and should be durable (IaaS).
    - **Advantages**: Very cost-effective, scalable, and integrable with almost any service.

3.4.2 Architecture

Users send requests with an ID through a web interface hosted on AWS Amplify, utilizing AWS Cognito for secure access control. The API Gateway receives these requests and forwards them to the Simple Queue Service (SQS), which manages and orders the requests. This triggers serverless functions (Lambda) that query the data warehouse (Redshift) and store the results in a NoSQL database (DynamoDB) as key-value pairs, using the ID as the key. Users can then enter their ID to view results on screen and download the data as a CSV file. The architecture ensures efficient data handling and user interaction, with services connected via IDs assigned by a VPC, while security groups filter requests sent to each service.

Diagram 2: System Architecture



3.4.3 The Web-application

As shown in the picture below, this architecture provides a straightforward way for users to interact with the system, worth to mention, creating a detailed website was beyond the scope of this project. The initial version of the website includes a bar for entering the query ID or writing queries. This web interface is developed using XML, focusing on demonstrating the core functionality and goals of the site.

Picture 3: Initial lunch of the website.

picture 4: the result of entering the query and id for retrieving the data.

Retrieve Data and Download as CSV

Send Query

select * from country limit 3 Send Query

Assigned ID: 341856

Retrieve Data by ID

341856 Get Data

Item Details:

id	name
1	Belgium
1729	England
4769	France

Download as CSV

4.Evaluation

In this design, data can exist in three modes, essential for creating an interactive information system. **Data at Rest** includes data stored in the data warehouse and S3, available for querying and selection by users. **Data in Motion** refers to data being transferred from S3 to the data warehouse and during active queries, involving transformation and movement processes. **Data in Serve** encompasses data stored in DynamoDB, ready for immediate access and download. These modes enable a comprehensive assessment of the system, ensuring efficient data handling, smooth interaction, and reliable performance.

Data at rest and data in serve, such as those stored in temporary and permanent databases, can be assessed and monitored through administrative roles and logs, while data in motion can be tracked and monitored primarily through logs.

Interactive information systems allow users to clean, visualize, and build models quickly through iterative, session-based processes. Users continuously manipulate data subsets. Evaluation metrics include scalability, usability and convenience, and latency (Jiang et al., 2018). Additionally, costs, performance, durability, availability and fault tolerance are important metrics for these systems.

4.1 Metrics

4.1.1 Scalability

The system is integrated with cloud infrastructure, offering significant scalability. During peak usage periods, it can scale horizontally by adding more instances to manage increased loads. Although some information services have fixed infrastructure, they can still achieve scalability by using load balancers to distribute traffic evenly and auto-scaling to adjust the number of active servers based on demand, ensuring reliable performance.

4.1.2 Usability and Convenience

This information system provides numerous features and demonstrates high data maturity. By optimizing identifiers (IDs) as suggested in the methodology, usability and maturity can be further enhanced. Data scientists can easily write queries, join tables, and select from a wide range of attributes to meet their research objectives. The system's design is intuitive, allowing users to efficiently perform complex data analyses, making it a valuable tool for researchers.

4.1.3 latency

The system is integrated into the London availability zone, providing minimal latency for British users. However, European users experience slightly higher latency compared to UK users. This setup is tailored for a European audience, but for other regions, alternative

websites can be implemented to route requests to the API infrastructure, optimizing performance globally.

4.1.4 Costs

Since all services are hosted in the cloud, leveraging its flexibility and payment options is beneficial. The cloud provides pay-as-you-go models and upfront payment plans for one or three years, allowing users to choose between on-demand or reserved instances based on their usage patterns (Mazrekaj et al., 2016).

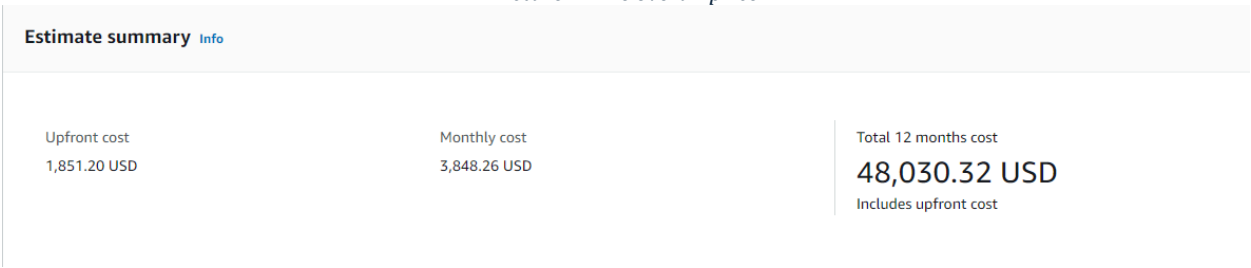
In this case study, I utilized the free tier, spending under £10 overall while using a $300 free tier for Redshift. As the website grows, costs may increase, but the owner can reserve plans for significant savings. The estimated monthly cost is $3,848, with an upfront payment of $1,851.20, totaling $48,030 for 12 months.

Table 2: The Costs analytic table

Service	Upfront Costs	Monthly Costs	First 12 months total Costs	Configuration
Amazon Redshift	0	2734.75$	32817.00$	32 RPU for 6 hours run
Amazon API Gateway	0	116$	1392.00$	10 million requests of average 34 kb size
DynamoDB provisioned capacity	1851.2$	317.13$	5656.76$	100 GB monthly data and 10kb average size
AWS Lambda	0	0	0.00	1M free requests per month and 400,000 GB-seconds of computing time per month.
Amazon Simple Queue Service (SQS)	0	0	0.00	one million requests
Public IPv4 Address	0	21.9$	262.80$	three subnets
S3 Standard	0	24.88$	298.56$	1 Terabyte of data with 100gs of transformation
Data Transfer	0	0	0.00	
Amazon Cognito	0	514.25$	6171.00$	10k users and enhanced security

AWS Amplify	0	119.35$	1432.20$	100gb of transmission and 500ms average request

Picture 5: The overall price

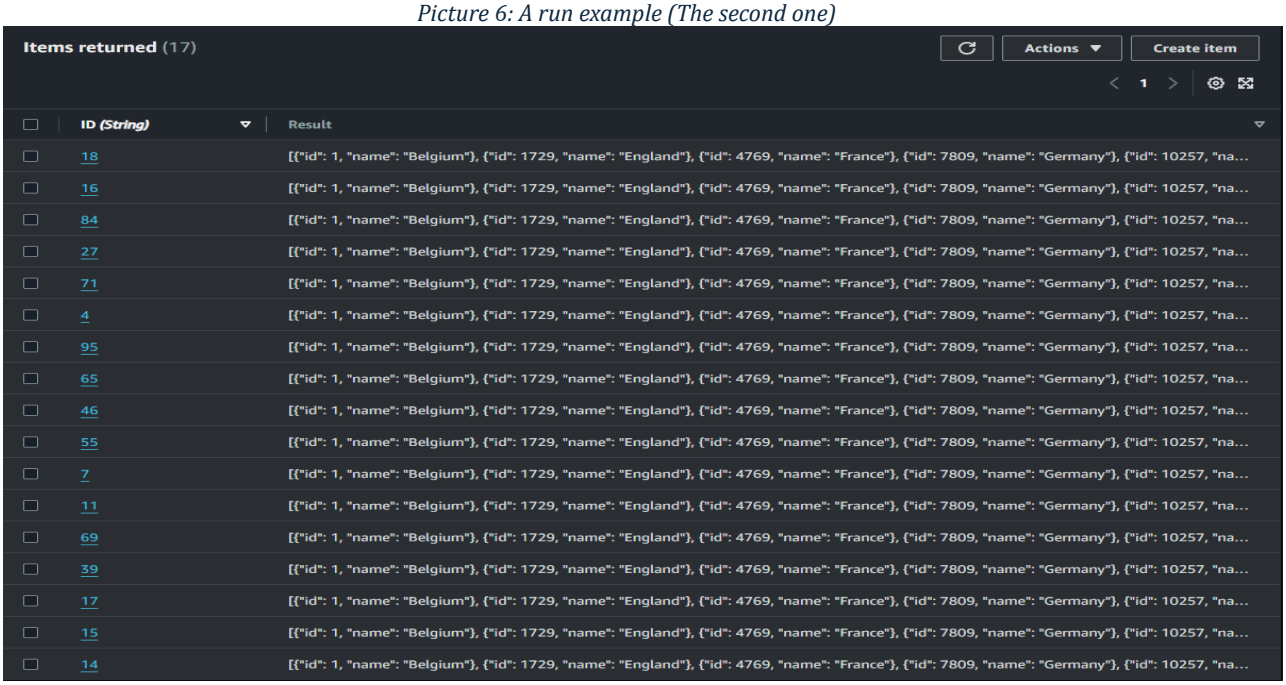


4.1.5 Performance Testing and Simulation

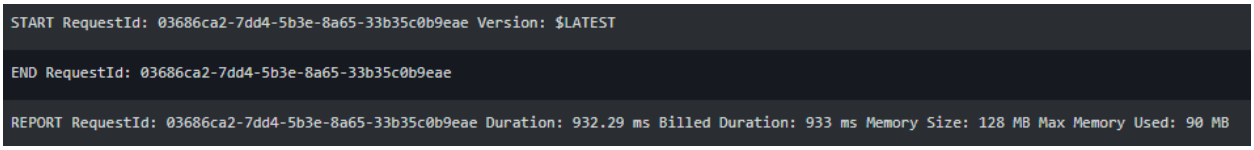
To evaluate the performance of the system, a Python application is utilized to simulate requests. This application sends queries with 11 rows and 2 columns and monitors the entire process from the first request to the completion of the last query step-by-step. It's worth mentioning that the Lambda function can log the queries, enabling us to monitor the process through logs.

The system was tested by running 400 queries in four batches of 100 records each, yielding 26, 17, 10, and 14 records saved per batch, with an average of 16.75 records saved. The transaction times ranged from 932 milliseconds to 1800 milliseconds. At peak, the system saved up to 26% of concurrent queries successfully, indicating that during rush hours, if 100 users query simultaneously within 2 seconds, only 26% will have their data saved. These results highlight the system's concurrency limitations and the need for enhanced scalability and robustness to improve performance and reduce failure rates.

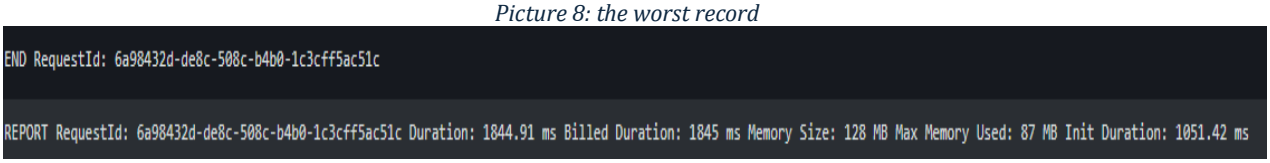
Picture 6: A run example (The second one)



Picture 7: the best record



Picture 8: the worst record



Given that other components are more scalable than the Lambda function, the most common scenario involves Lambda failures. Therefore, it is crucial to focus on testing and monitoring the Lambda function. This approach allows us to identify and address any performance bottlenecks or issues within the Lambda function, ensuring the overall robustness and reliability of the system. Although in this test the DynamoDB was also involved.

4.1.6 Durability

The data gathered for data science is critical and requires meticulous effort to generate and maintain. Therefore, ensuring its durability is paramount. The main component is the data warehouse, with Amazon Redshift implemented in their availability zones. The system spans three availability zones in Ireland, London, and Paris, safeguarding against disasters and data loss by replicating data across these locations. Similarly, the secondary database, MongoDB, is established in three availability zones, ensuring durability and redundancy.

4.1.7 Availability

Cloud providers guarantee infrastructure availability through Service Level Agreements (SLAs). AWS promises 99.99% availability, equating to approximately 364.96 days per year. This means downtime is less than one hour annually, which is a significant advantage for the system, ensuring low system downtime and reliable performance.

4.1.8 Fault Tolerance

The system is designed with decoupled components to ensure that the failure of one does not affect others. For instance, the data warehouse is isolated from direct public internet access. During high demand, requests are optimized for performance. If a site goes down, primary storage remains secure. Additionally, if a Lambda function fails to execute a request, it does not impact other requests, maintaining overall system reliability.

4.1.9 Security

The system adheres to over 20 security standards in the cloud (Hendre et al., 2015) focusing on both knowledge and operational issues. Security is enhanced through a shared responsibility model, where cloud providers secure their services. Data governance is critical, controlling who can access specific data and for how long. Security groups and access controls restrict unauthorized users, while data governance rules prevent the deletion or caching of critical data.

4.1.10 Data Privacy

Data privacy is a crucial aspect of data security. Users do not register or provide personal information on the website, ensuring anonymity. In public use cases, registration or payment methods are avoided, allowing for a pool of users while protecting sensitive data from exposure.

4.1.11 Retention and Recovery

User custom data is stored in MongoDB, ensuring durability and allowing users to retrieve their data using assigned IDs. In a released version of the site, data retention can be managed based on contracts or user accounts, enabling owners to save their data for specified periods. Additionally, initial data is saved on S3, providing backup and recovery options in case of downtime.

4.2 Overcoming the Limitations of The System

In the era of big data, many information systems are hosted in Kubernetes clusters to address connectivity issues and ensure services are loosely coupled. For example, Lambda functions can be problematic; if one fails, the entire process fails, requiring users to retrieve data again, which increases the likelihood of further failures. Additionally, the integration of services like SQS and Lambda means a problem in SQS can cascade, affecting the whole system. Lambda functions are also restricted to executing code within 15 minutes, making it unsuitable for querying large datasets, thus limiting user data retrieval capabilities. Moreover, heavy reliance on Lambda can cause system vulnerabilities, especially during high-traffic events like sports matches. To solve these issues, using Kubernetes with containerized services (software as a service) can be advantageous. This approach not only addresses the limitations of Lambda but also provides benefits in infrastructure management by shifting from PaaS and IaaS to SaaS, thus leveraging the advantages of serverless computing.

5. Recommendations and Conclusion

In our previous discussions, we covered the aims, scope, problem identification, solution implementation, and result evaluation. Now, it is essential to consider ways to enhance the system further. The following recommendations aim to increase system capabilities, optimize performance, and ensure scalability.

5.1 Increase System Capabilities

Enhancing system capabilities involves improving the performance, efficiency, and scalability of the core components. This can be achieved through various strategies, such as optimizing the data warehouse, enhancing API gateways, improving queue and email services, and refining Lambda functions.

5.1.2 Data Warehouse

The data warehouse is a critical component of the system, primarily involving data storage and retrieval processes. To improve its performance:

1. **Vertical Scaling**: Increase the data warehouse's vertical capabilities, scaling from 32 Read Processing Units (RPU) to 512 RPU. This substantial increase in RPUs will expedite query execution, reducing query times by up to 1/16 of the current duration. The current query execution time ranges from 0.9 seconds to 1.9 seconds.
2. **Memory Efficiency**: Optimize the data warehouse to be more memory efficient, which will help in smoother performance and faster query processing.

5.1.3 API Gateways

Currently, there is a single API and SQS handling requests through one gateway. To improve this setup:

1. **Multiple APIs**: Implement multiple APIs attached to the system. This will allow requests to be sent via different gateways.
2. **Load Balancing and Autoscaling**: Use load balancers and autoscaling services to distribute the requests efficiently across multiple APIs. This will optimize performance and ensure that the system can handle varying loads effectively.

5.1.4 Queue Services and Email Services

At present, the system saves up to 26 of queries of 100 queries in 1 seconds. To enhance this:

1. **Dead Letter Queues (DLQ)**: Enable dead letter queues to save unsent or failed messages. These messages can be retried and sent during low load periods, improving overall system reliability.
2. **Email Services**: Integrate email sending services to notify users about the status of their queries or system updates. This requires setting up an organizational email domain, which has not been done yet due to the lack of an own email domain.

5.1.5 Lambda Functions

The current Lambda function setup involves three main roles: receiving the request, querying the warehouse, and saving the results in a secondary database. To improve this:

1. **Function Separation**: Separate these roles into three distinct Lambda functions. This modular approach will allow each function to be optimized and managed independently.
2. **Scalability**: While Lambda functions are inherently serverless and scalable, ensuring that each function is optimized for its specific task will enhance overall system performance. AWS manages scalability, but configuration optimizations can still be applied.

By implementing these recommendations, the system will become more efficient, scalable, and capable of handling increased loads and complex queries. These improvements will lead to better performance and a more robust infrastructure.

5.2 Superior Approach: Kubernetes

To further enhance the system's fault tolerance and operational efficiency, consider using Kubernetes for container orchestration. Kubernetes provides a serverless option managed by services, which can be optimized for various goals and interpreted as microservices.

1. **Microservices Architecture**: Containerizing services and running them as microservices can reduce management and operational tasks. Microservices allow each component to be developed, deployed, and scaled independently.

2. **Operational Capabilities**: Kubernetes supports resilience and elasticity, providing features such as self-healing, load balancing, and automatic scaling (Medel et al., 2018).

3. **Adoption and Industry Standards**: Approximately 60% of large organizations implement Kubernetes, indicating its reliability and industry acceptance (Jeffery, A., et al., 2021).

4. **Improved Availability and Scalability**: Using Kubernetes as the orchestration tool can lead to better availability, healing, and autoscaling capabilities (Poniszewska-Marańda and Czechowska, 2021).

By implementing these recommendations, the system will become more efficient, scalable, and capable of handling increased loads and complex queries. These improvements will lead to better performance and a more robust infrastructure.

5.3 Conclusion

The paper aims to provide an effective data engineering lifecycle or information system beneficial for football enthusiasts, whether for professional or personal interest, while considering academic resources and research. The system achieves its primary goal of gathering and generating data, optimizing infrastructure for hosting and operations, and establishing sharing protocols to ensure accessibility. Initially, the paper defined its aims and reviewed related work in other sports industries. Subsequently, it detailed the suggested methods and efficient approaches implemented. Finally, it evaluated the system's strengths, weaknesses, and performance, proposing solutions to address any shortcomings. Overall, the system successfully meets its objectives by creating a rational environment for users.

6. refences

FIFA, 2018. *FIFA Financial Report 2018, p.15*. [pdf] Available at: https://digitalhub.fifa.com/m/337fab75839abc76/original/xzshsoe2ayttyquuxhq0-pdf.pdf [Accessed 17 July 2024]

FIFA, 2022. *2019-2022 Cycle in Review: 2022 Financial Highlights*. [online] Available at: https://publications.fifa.com/en/annual-report-2022/finances/2019-2022-cycle-in-review/2022-financial-highlights/ [Accessed 17 July 2024]

Deloitte, 2023. *Annual Review of Football Finance 2023, p.8*. [pdf] Available at: https://www.deloitte.com/content/dam/assets-zone2/uk/en/docs/services/financial-advisory/2024/deloitte-uk-annual-review-of-football-finance.pdf [Accessed 17 July 2024].

Wicker, P., Breuer, C. and Hennigs, B., 2012. Understanding the interactions among revenue categories using elasticity measures—Evidence from a longitudinal sample of non-profit sport clubs in Germany. *Sport Management Review*, 15(3), pp.318-329. Available at: https://www.sciencedirect.com/science/article/abs/pii/S1441352311000957 [Accessed 17 July 2024].

Shonk, D.J. and Weiner, J.F., 2021. *Sales and revenue generation in sport business*. Human Kinetics, p.2. Available at https://books.google.co.uk/books?id=TatJEAAAQBAJ&lpg=PR1&ots=rYW6hJchRg&dq=sport%20revenue&lr&pg=PA2#v=onepage&q=sport%20revenue&f=false

Thakkar, P. and Shah, M., 2021. An assessment of football through the lens of data science. *Annals of Data Science*, pp.823-835. Available at: https://link.springer.com/article/10.1007/s40745-021-00323-2 [Accessed 17 July 2024].

Xiao, L., Cao, Y., Gai, Y., Liu, J., Zhong, P. and Moghimi, M.M., 2023. Review on the application of cloud computing in the sports industry. *Journal of Cloud Computing*, 12(1), pp.1-10. Available at: https://link.springer.com/article/10.1186/s13677-023-00531-6 [Accessed 17 July 2024]

Baskarada, S. and Koronios, A., 2013. Data, information, knowledge, wisdom (DIKW): A semiotic theoretical and empirical exploration of the hierarchy and its quality dimension. *Australasian Journal of Information Systems*, *18*(1), pp.1-20. Available from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2304010

Che, D., Safran, M. and Peng, Z., 2013. From big data to big data mining: challenges, issues, and opportunities. In: *Database Systems for Advanced Applications: 18th International Conference, DASFAA 2013, International Workshops: BDMA, SNSM, SeCoP*, Wuhan, China, April 22-25, 2013. Proceedings 18. Springer Berlin Heidelberg, pp. 1-15.

Alasadi, S.A. and Bhaya, W.S., 2017. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12(16), pp.4102-4107.

Xiao, L., Cao, Y., Gai, Y., Liu, J., Zhong, P. and Moghimi, M.M., 2023. Review on the application of cloud computing in the sports industry. *Journal of Cloud Computing*, 12(1), pp.1-20. Available at: https://link.springer.com/article/10.1186/s13677-023-00531-6 [Accessed 17 July 2024].

Raković, P. and Lutovac, B., 2015. A cloud computing architecture with wireless body area network for professional athletes' health monitoring in sports organizations—Case study of Montenegro. In: *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, pp. 387-390. IEEE.

Li, A. and Huang, W., 2023. A comprehensive survey of artificial intelligence and cloud computing applications in the sports industry. *Wireless Networks*, pp.1-12.

Chomątek, Ł. and Sierakowska, K., 2021. Automation of basketball match data management. *Information*, 12(11), p.461, (pp.1-15).

Cao, C., 2012. Sports data mining technology used in basketball outcome prediction, pp.1-16.Available at: https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1040&context=scschcomdis [Accessed 17 July 2024]

Al-Asadi, M.A., Taşdemir, Ş. and Tezcan, B., 2018. An online information system for football club management. In: *Kongre Kitapçığı/Congress Proceedings Book*, pp. 46-50.

Atasoy, B. and Özer, U., 2021. Database knowledge management in sport. *International Journal of Sport Culture and Science*, 9(1), pp.37-53.

Horvat, T., Havas, L., Srpak, D. and Medved, V., 2019. Data-driven basketball web application for support in making decisions. In: *icSPORTS*, pp. 239-244.

Wong, T.M., 2001. Implementing a database information system for an electronic baseball scorecard. Dartmouth College Department of Computer Science Thesis (Undergraduate), pp.1-23.Available at: https://digitalcommons.dartmouth.edu/cgi/viewcontent.cgi?article=1013&context=senior_theses [Accessed 17 July 2024].

Hafizul, N.A.U.B.M., Ghani, M.A.A. and Sainan, K.I., 2021. Data management system for a series of kinematic movement in football analytics. *Borneo Engineering and Technology Journal*, 3(1), pp.12-20. Available at: https://bep.uitm.edu.my/fkm/images/pdf/vol3bep22.pdf [Accessed 17 July 2024]

Foschi, F., 2021. Design of a data management system for value bet detection and soccer performance analysis. Doctoral dissertation, Politecnico di Torino, pp. 1-37.

Kumar, K. and Azad, S.K., 2017. Database normalization design pattern. In: 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), October. IEEE, pp. 318-322. Available at: https://ieeexplore.ieee.org/abstract/document/8251067 [Accessed 18 July 2024].

Batini, C., Cappiello, C., Francalanci, C. and Maurino, A., 2009. Methodologies for data quality assessment and improvement. *ACM Computing Surveys (CSUR)*, 41(3), pp.1-52. Available at: https://dl.acm.org/doi/abs/10.1145/1541880.1541883 [Accessed 18 July 2024].

Sidi, F., Panahy, P.H.S., Affendey, L.S., Jabar, M.A., Ibrahim, H. and Mustapha, A., 2012. Data quality: A survey of data quality dimensions. In *2012 International Conference on Information Retrieval & Knowledge Management* (pp. 300-304). IEEE. Available at: https://ieeexplore.ieee.org/abstract/document/6204995 [Accessed 18 July 2024].

Haug, A., Zachariassen, F. and Van Liempd, D., 2011. The costs of poor data quality. *Journal of Industrial Engineering and Management (JIEM)*, 4(2), pp.168-193. Available at: https://www.econstor.eu/handle/10419/188448 [Accessed 18 July 2024].

Alhassan, I., Sammon, D. and Daly, M., 2016. Data governance activities: an analysis of the literature. *Journal of Decision Systems*, 25(sup1), pp.64-75. Available at: https://doi.org/10.1080/12460125.2016.1187397 (Accessed: 18 July 2024).

Masrom, M. and Rahimli, A., 2014. A review of cloud computing technology solution for healthcare system. *Research Journal of Applied Sciences, Engineering and Technology*, 8(20), pp. 2150-2153. Available at: https://www.cabidigitallibrary.org/doi/full/10.5555/20153285980 (Accessed: 18 July 2024).

Lupşe, O.S., Vida, M.M. and Tivadar, L., 2012, April. Cloud computing and interoperability in healthcare information systems. In: The first international conference on intelligent systems and applications, pp. 81-85. Available at: https://www.researchgate.net/profile/Vasile-Stoicu-Tivadar/publication/267781309_Cloud_Computing_and_Interoperability_in_Healthcare_Information_Systems/links/54bcb6200cf253b50e2d5380/Cloud-Computing-and-Interoperability-in-Healthcare-Information-Systems.pdf (Accessed: 18 July 2024).

Verma, H., 2013. Data-warehousing on cloud computing. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(2), pp. 411-416. Available at: https://d1wqtxts1xzle7.cloudfront.net/53834292/data_warehousing-libre.pdf?1499840478=&response-content-

disposition=inline%3B+filename%3DData_warehousing_on_Cloud_Computing.pdf&Expires=1721323625&Signature=CVj3CkWd12sB8PE4E3xWqzbjX8OfJegD-MkSZbBi9-sDdMmzlw3cpUYLd3L5tXgylC8RCCw8s1iZgZ0pnlkzS~gmSfqD5upVt5dC9sngg5QkhLttXMWHP73LdGtu7KzUkigD1tSFuwWlXHgscuOhI53t5Con9Z9U4RN5C6NMO~S8qeb4MEz2Zg3elkpzkzRpTmcasYdNNrB2-jrYK8lnaHx3OtXfmyhJrXRna14ULoYf6W6x0RroT74XPr8tLArCHsMimgBqLG40aVt1F~BAEXAQdD1qrbwI3eAo4I-tarPdf4oxFw-GaXYpRnrb0dgW4El8nHbXY1FzBMlBVnZfiQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (Accessed: 18 July 2024).

Rehman, K.U.U., Ahmad, U. and Mahmood, S., 2018. A Comparative Analysis of Traditional and Cloud Data Warehouse. *VAWKUM Trans. Comput. Sci.*, 6, pp. 34-40. Available at: https://d1wqtxts1xzle7.cloudfront.net/91355822/8e6e0868f8ea05a3dde2c4822e62f27cf0e9-libre.pdf?1663788944=&response-content-disposition=inline%3B+filename%3DA_Comparative_Analysis_of_Traditional_an.pdf&Expires=1721324170&Signature=I0Nu6o-qvmYkMXd0Oh~YQAUGPuIxAXqUjQIki6l6sgP2yQkcqEuXqSBOFx6yFjU4f4Ai5wzz4n1iElOhmeoRjI2MjIiOhUElosobeOyni9ZJVHecI9W89D7vzZrbsDFpinoTcMV5h7iUQNhfvdRzxJQYRHQdJKdEVkmxFbTc2~rGdJbHiIxW0yzLT6Iwl5eqwfCQabvzBwXy-VfbiSOFrf7hj3e335iAGjbroVjcIzR3Cso0fKKsiE1nabthsDdKycS6Wrm5DnBY7vjY79pi4Fv~ouA-uedy86IVsCPz20TBZhAjJFtHpCF61GTibwqH0ZQ4g2drWfAinNrySo82JQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (Accessed: 18 July 2024).

Lynn, T., Rosati, P., Lejeune, A. and Emeakaroha, V., 2017, December. A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms. In: 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 162-169. IEEE. Available at: https://d1wqtxts1xzle7.cloudfront.net/91355822/8e6e0868f8ea05a3dde2c4822e62f27cf0e9-libre.pdf?1663788944=&response-content-disposition=inline%3B+filename%3DA_Comparative_Analysis_of_Traditional_an.pdf&Expires=1721324170&Signature=I0Nu6o-qvmYkMXd0Oh~YQAUGPuIxAXqUjQIki6l6sgP2yQkcqEuXqSBOFx6yFjU4f4Ai5wzz4n1iElOhmeoRjI2MjIiOhUElosobeOyni9ZJVHecI9W89D7vzZrbsDFpinoTcMV5h7iUQNhfvdRzxJQYRHQdJKdEVkmxFbTc2~rGdJbHiIxW0yzLT6Iwl5eqwfCQabvzBwXy-VfbiSOFrf7hj3e335iAGjbroVjcIzR3Cso0fKKsiE1nabthsDdKycS6Wrm5DnBY7vjY79pi4Fv~ouA-uedy86IVsCPz20TBZhAjJFtHpCF61GTibwqH0ZQ4g2drWfAinNrySo82JQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (Accessed: 18 July 2024).

Niranjanamurthy, M., Archana, U.L., Niveditha, K.T., Jafar, S.A. and Shravan, N.S., 2014. The research study on DynamoDB—NoSQL database service. *Int. J. Comput. Sci. Mob. Comput.*, 3(10), pp. 268-279. Available at: https://d1wqtxts1xzle7.cloudfront.net/35162087/V3I10201487-

libre.pdf?1413494148=&response-content-disposition=inline%3B+filename%3DThe_Research_Study_on_DynamoDB_NoSQL_Dat.pdf&Expires=1721324468&Signature=W-4ii99fGczHEXOnJx9616gNvN07gWbBuUU129LKM49PBupLC2U9fxWno7Bt8ZnNQT7zeoerkE0UE806gsNHn9g4KB6jfsY3ydGoIdiyyJTBNkl36iaQbrlJsI347LQ6ywPfSyp8LmAllNK9fDWrt1OrAl196ikidX9HTpnCdnsespXtyUrastsMNBIAmzv-t31qpUqpnqDXPW-CAvJhv7-G2IZvkAygiMu6zJvkn7Dp~ymLOyNqlCJbS8LVAq16DSodigcrY6R7uaBcC~s1GhIw3rLYDYJY0inGT2GdismSAKbYRXpDEvtSHwke518sKmoi39PRFA07eGM4HcK7XA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (Accessed: 18 July 2024).

Jiang, L., Rahman, P. and Nandi, A., 2018, May. Evaluating interactive data systems: Workloads, metrics, and guidelines. In: *Proceedings of the 2018 International Conference on Management of Data*, pp. 1637-1644. ACM Digital Library. Available at: https://dl.acm.org/doi/abs/10.1145/3183713.3197386 (Accessed: 18 July 2024).

Mazrekaj, A., Shabani, I. and Sejdiu, B., 2016. Pricing schemes in cloud computing: an overview. *International Journal of Advanced Computer Science and Applications*, 7(2), pp. 80-86. Available at: https://d1wqtxts1xzle7.cloudfront.net/86779253/Paper_11-Pricing_Schemes_in_Cloud_Computing_An_Overview-libre.pdf?1654018674=&response-content-disposition=inline%3B+filename%3DPrishtina_Republic_of_Kosovo.pdf&Expires=1718735657&Signature=gTT~7BrO1rqsvyksgDBkEX41Y7~p0zshbABzE3FaQWD248ySDer8KXNzV6qjLYlQFt61k2R0Fy9bhJwpSYAiM296IjRgu6po0YHmAY4wei~rkbI94qVB6ps6zwgvDdwOpIgsALsItTjHS07OR94XNLSzfw0FZ~jlr6pCzAonoM9zwhaEDEMix1y17nQCVTfkY2dh55l7ww2TLHAx0R2mg6aarIA9u~VZr-w4t0Nf-ktk1JevCy5P6w0Qy7IH8faKKweIcByrRsrV~GLvr052tlsgaALZwNF5I9nVsCB4yvFRbI0J0cZoVDr4OZtZCxl~66nu1xsjyXXZWGauca8WBA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (Accessed: 18 July 2024).

Hendre, A. and Joshi, K.P., 2015, June. A semantic approach to cloud security and compliance. In: *2015 IEEE 8th International Conference on Cloud Computing*, pp. 1081-1084. IEEE. Available at: https://ieeexplore.ieee.org/abstract/document/7014763 (Accessed: 18 July 2024).

Medel, V., Tolosana-Calasanz, R., Bañares, J.Á., Arronategui, U. and Rana, O.F., 2018. Characterising resource management performance in Kubernetes. *Computers & Electrical Engineering*, 68, pp. 286-297. Available at: https://www.sciencedirect.com/science/article/pii/S0045790617315240 (Accessed: 18 July 2024).

Jeffery, A., Howard, H. and Mortier, R., 2021, April. Rearchitecting Kubernetes for the edge. In: *Proceedings of the 4th International Workshop on Edge Systems, Analytics and*

Networking, pp. 7-12. ACM Digital Library. Available at: https://dl.acm.org/doi/abs/10.1145/3434770.3459730 (Accessed: 18 July 2024).

Poniszewska-Marańda, A. and Czechowska, E., 2021. Kubernetes cluster for automating software production environment. *Sensors*, 21(5), p. 1910. Available at: https://www.mdpi.com/1424-8220/21/5/1910 (Accessed: 18 July 2024).

7. List of Illustrations

7.1 Figures

7.2 Tables

7.3 Diagrams

7.4 Pictures