Scalable ETL Processes with Change Data Capture (CDC) and Monitoring Using Apache Superset

Aneta Trajkovska[1], Violeta Manevska[1] and Kostandina Veljanovska[1]

[1] *University "St. Kliment Ohridski" – Bitola, Faculty of Information and Communication Technologies, Bitola, ul. Partizanska bb, Bitola, Republic of Macedonia*

aneta.trajkovska@uklo.edu.mk;violeta.manevska@uklo.edu.mk;kostandina.veljanovska@uklo.edu.mk

Abstract:

In an increasingly data-driven world, the ability to monitor and respond to real-time changes is critical across various domains, including marketing, transportation, and other dynamic sectors. Timely access to up-to-date information enables organizations to make informed decisions and adapt quickly to changing conditions. This paper presents a study on the implementation of a real-time data monitoring solution that leverages Change Data Capture (CDC) techniques for capturing updates across data sources. Furthermore, it explores the integration of real-time visualization tools, such as Apache Superset, to provide immediate insights and improve situational awareness. The proposed approach facilitates continuous tracking of data changes and supports decision-making processes through intuitive and dynamic dashboards. The research highlights the architectural components, key technologies, and practical benefits of deploying such systems in modern data environments.

Keywords:

ETL(Extract, Transform and Load), Azure services, CDC(Change data Capture), Apache Superset, Databricks.

1. Introduction

As modern systems become increasingly dynamic and data-intensive, the ability to access and respond to real-time information is becoming increasingly vital across a wide range of industries. From marketing and customer engagement to transportation and logistics, decision-makers rely heavily on timely data to adapt to rapidly changing environments. Delayed or outdated data can lead to missed opportunities, inefficiencies, or even critical system failures. As such, there is a growing need for data systems that not only collect, and process information continuously but also provide immediate insights through real-time monitoring and visualization [1].

Traditional data pipelines, which operate on batch processing principles, often fail to meet the demands of modern applications that require low-latency or near-instantaneous data updates. To address this limitation, modern data engineering has introduced techniques such as Change Data Capture (CDC) — a method of identifying and capturing changes in data sources in real time. CDC enables continuous data flow from operational systems to analytical platforms without the need for full data refreshes, thereby enhancing performance and reducing load [2],[3].

This paper investigates the implementation of a real-time ETL (Extract, Trans-form, Load) pipeline using CDC mechanisms to capture and process data changes as they occur. The research further explores how such changes can be visualized effectively through the integration of Apache Superset, an open-source platform for data exploration and interactive dashboards. Superset plays a critical role in this architecture by enabling users to monitor key metrics, observe patterns, and detect anomalies as they emerge. By combining real-time data capture with intuitive visualization, this approach supports proactive decision-making and im-proves operational transparency. The implementation is relevant for a variety of sectors, including but not limited to digital marketing—where customer behavior must be tracked in real time—and transportation systems—where logistics and fleet data change continuously and must be acted upon immediately [4].

The remainder of this paper is organized as follows: Section 2 provides back-ground on real-time data processing, CDC methodologies, and Apache Superset. Section 3 outlines the system architecture and technologies used in the proposed solution. Section 4 presents the significance of data cleaning in preparing data for analytics and visualization . Section 5 outlines the analytical results obtained and explores the potential of Apache Superset as a tool for continuous monitoring and performance evaluation. Section 6 presents advantages of the proposed solution. The limitations detected are presented in Section 7, while Section 8 concludes the paper with recommendations for future research and development.

2. Foundations of Real-Time Data Engineering: CDC Methodologies and Visualization Tools

The shift from batch-oriented data systems to real-time data processing has redefined how organizations collect, manage, and analyze information. As data sources become more dynamic and business decisions increasingly rely on im-mediate insights, the need for efficient and scalable real-time data engineering solutions has grown substantially. This section explores the foundational technologies that enable real-time data workflows, focusing specifically on CDC as a method for detecting and propagating data changes, and Apache Superset as a platform for real-time data visualization and monitoring [5].

The ETL process is a fundamental framework in data engineering, responsible for the systematic integration of data from multiple sources into a centralized data repository, such as a data warehouse or data lake. During the extraction phase, raw data is collected from diverse, often heterogeneous sources, including relational databases, APIs, and file systems. In the transformation phase, the data undergoes cleaning, normalization, aggregation, and enrichment to ensure consistency, accuracy, and compatibility with analytical models. Finally, the loading phase involves inserting the processed data into the target system, where it be-comes available for querying, reporting, and advanced analytics.

CDC plays a critical role in modern data pipelines by capturing insertions, up-dates, and deletions from source systems and streaming them to downstream consumers with minimal latency. It eliminates the need for frequent full-table scans or scheduled batch jobs, thereby improving system performance and ensuring fresher data availability for analytics. Common CDC implementations rely on tools such as Debezium, Kafka Connect, or native database features, which support integration with distributed data platforms and stream processing engines. In parallel, visualization tools such as Apache Superset provide users with interactive dashboards and visual analytics that reflect the most current state of the data. Superset supports a wide range of data sources and allows non-technical stake-holders to monitor key metrics and trends in real time through customizable visualizations [6], [7].

Together, these technologies form a powerful architecture for building responsive, insight-driven systems that meet the real-time demands of modern data eco-systems. The following subsections delve deeper into the principles of CDC, key components of real-time data processing, and the role of Apache Superset in enabling effective data observability and decision-making.

3. System Architecture and Technology Stack of the Proposed Real-Time Monitoring Solution

This section presents the architecture and components of the proposed real-time data monitoring system. The system is designed to capture data changes from operational databases using CDC, process the data through a real-time ETL pipeline, and visualize metrics through Apache Superset for actionable insights. The architecture prioritizes low latency, scalability, and modularity, making it suitable for use cases such as real-time analytics, anomaly detection, and dynamic reporting, shown in Figure 1.

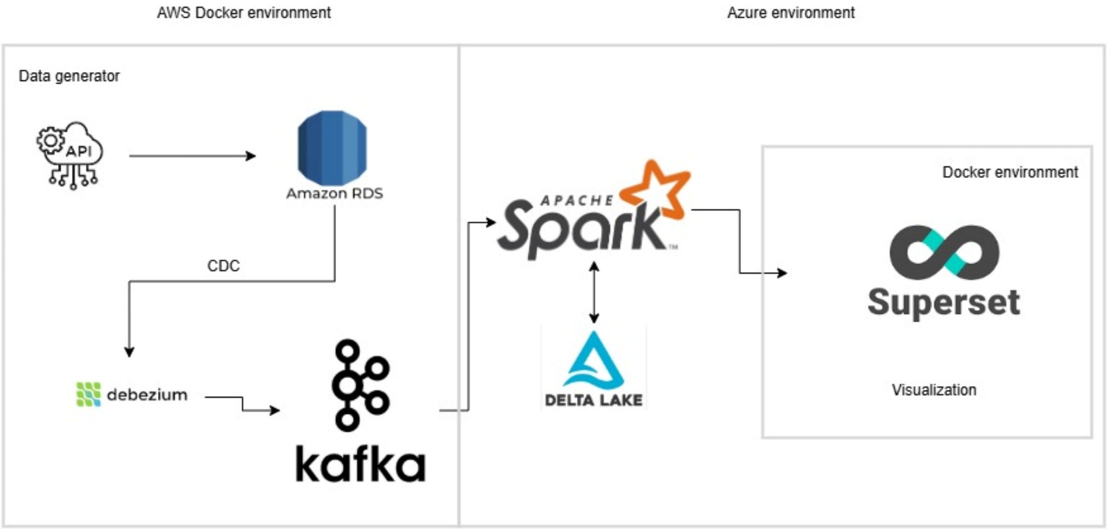


Figure 1: High-level architecture diagram

3.1. Overview of the System Architecture

The proposed architecture adopts a modular, event-driven design to ensure scalability, maintainability, and efficient resource utilization. The underlying computing infrastructure is provisioned and managed using Infrastructure as Code (IaC) principles, implemented through Terraform. This approach enables automated and repeatable deployment of cloud resources using a declarative configuration language, reducing the risk of manual errors and ensuring consistent environment management. Within this architecture, dedicated modules are developed for deploying the Kafka environment in the us-east-1 region and provisioning the Amazon RDS database, both hosted on Amazon Web Services (AWS). [8].

The subsequent component of the architecture involves integrating data from the Kafka consumer group into the Azure environment, specifically targeting Databricks tables to facilitate real-time data stream processing [9]. An Apache Superset instance is deployed on a virtual machine using Docker, enabling seamless connectivity to the Databricks tables which are part of the Delta Lake in Azure. This configuration allows for the dynamic retrieval of streaming data and the generation of intuitive and interactive data visualizations, presented in Figure 2.

3.2. Data Flow Description

The Apache Kafka cluster is configured to capture and log all changes occurring within the transactional tables of a relational database. Deployment strategy was centered on Amazon Elastic Container Service (ECS), leveraging containerization to ensure scalability and portability. Container images, including the Python-based data generation API, the Debezium Kafka connector, the Confluent Schema Registry, and Confluent Kafka components, were retrieved from AWS Elastic Container Registry (ECR). These components collectively facilitated the provisioning and orchestration of the application environment using Docker, enabling efficient management of data ingestion and schema evolution within the streaming architecture [9].

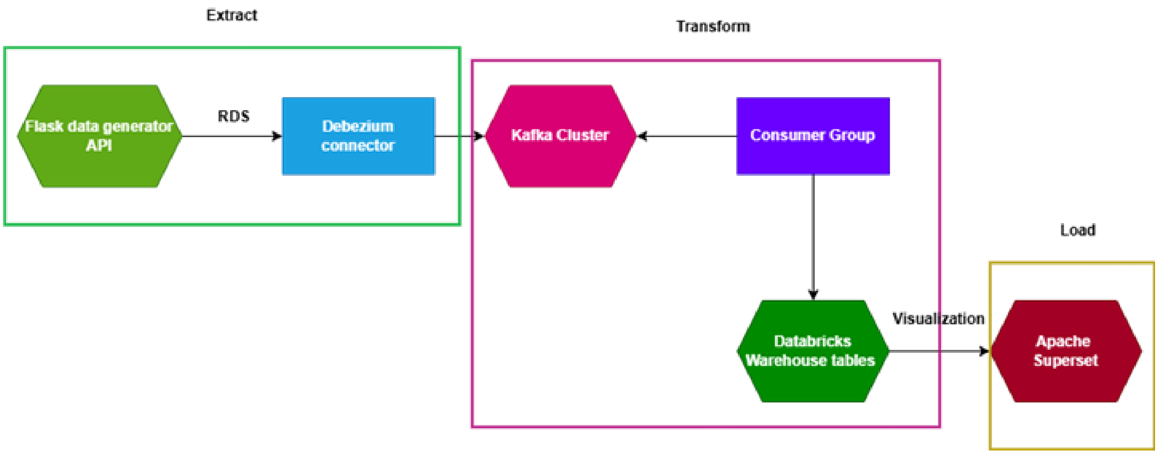


Figure 2: ETL Data flow description diagram

Debezium is employed as a CDC tool to continuously monitor the write-ahead logs (WAL) of the source relational database. It captures all data modification operations, including inserts, updates, and deletes, in real time. These changes are then serialized and published to designated Kafka topics, enabling downstream processing. To consume and handle the streamed data, custom application logic is implemented using Kafka consumers that subscribe to the relevant topics and process the data accordingly for further transformation or analysis [10].

Dashboards may include key performance indicators (KPIs), change frequency visualizations, alerting thresholds, and pipeline health metrics. Superset supports scheduled refreshes, SQL-based metrics, and real-time filters, enhancing observability.

4. Importance of Data Cleaning for Data Analytics and Visualization

Data cleaning, also referred to as data preprocessing, is a crucial step in the data analytics lifecycle, ensuring the accuracy, reliability, and interpretability of analytical results. Raw data collected from various sources is often incomplete, inconsistent, duplicated, or contains errors, which can lead to misleading conclusions if used directly for analysis or visualization. Uncleaned data introduces bias and noise into the analysis process, reducing the validity of insights and potentially leading to incorrect decision-making.

The process of cleaning data plays a central role in enhancing data quality, which is essential for meaningful analytics and effective visual representation. Accurate and consistent data allows for the generation of reliable models and visualizations, enabling researchers and decision-makers to better understand patterns, trends, and relationships within the dataset. For example, missing values, duplicate records, and formatting inconsistencies must be addressed to avoid skewed statistical results and inaccurate graphical representations. Furthermore, standardized data formats improve interoperability between analytics platforms and visualization tools such as Apache Superset, ensuring seamless integration and performance. Furthermore, standardized data formats improve interoperability between analytics platforms and visualization tools such as Apache Superset, ensuring seamless integration and performance [11].

From a visualization perspective, data cleaning is equally significant. Poor-quality data can distort visual outputs, making dashboards difficult to interpret and reducing their usability for decision-making. Clean and well-prepared datasets enable the creation of clear, precise, and interpretable visualizations that effectively communicate complex information to diverse audiences. This is especially important in research and business environments where stakeholders rely on visual insights for strategic planning [12].

In this research, data cleaning was conducted as a preliminary step before importing the dataset into Superset for analysis and visualization. The cleaning process involved handling missing data, removing duplicate entries, standardizing categorical values, and restructuring the dataset to meet the requirements of the visualization platform. By ensuring high data quality, the analytics process was optimized, resulting in accurate visual representations and trustworthy insights, Figure 3.

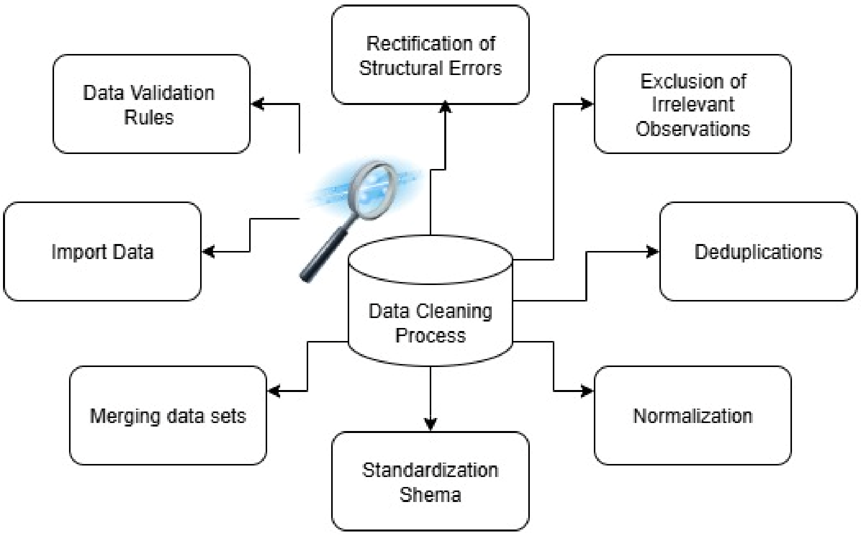
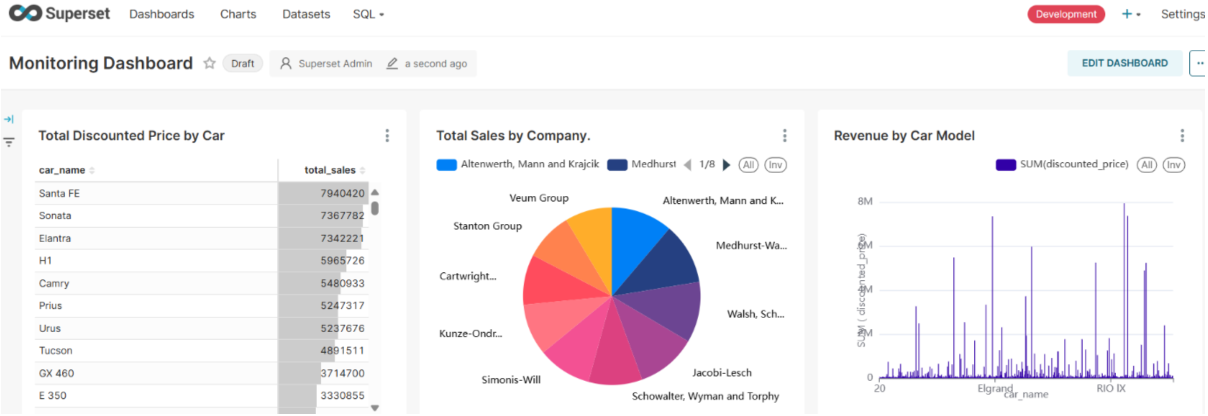


Figure 3: Data cleaning components

5. Analysis with Apache Superset

Apache Superset was utilized as the primary tool for the analysis and visualization of the datasets. Superset is an open-source, modern business intelligence (BI) platform designed to handle large datasets and provide interactive and highly customizable visualizations [13], [14].

Prior to initiating the analysis, the data was loaded into a structured relational database in Databricks and connected to Superset. This connection allowed seamless access to raw and processed data while leveraging SQL queries for dynamic exploration. Superset's SQL Lab module was particularly valuable, enabling the creation of custom queries to join and aggregate data from multiple sources, such as the Sales, Customers, and Companies tables, shown in Figure 4. This process ensured that the analysis could be tailored to the specific research objectives and that only relevant subsets of data were retrieved, improving performance and clarity.

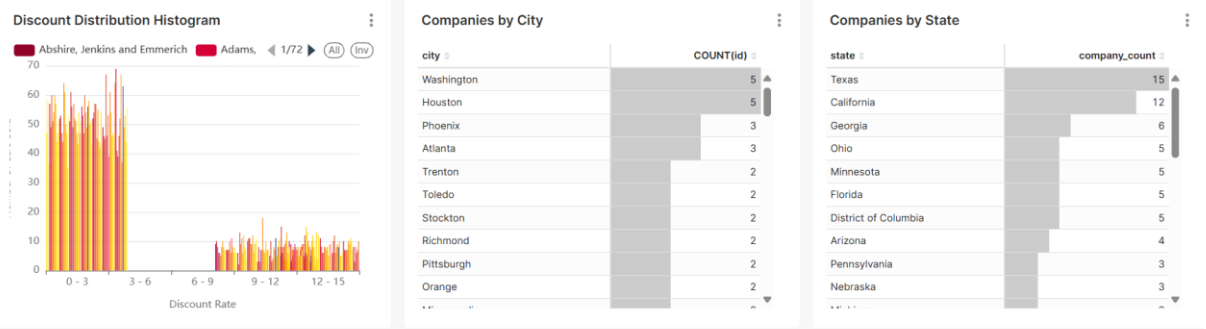


Total Discounted Price by Car

car_name	total_sales
Santa FE	7940420
Sonata	7367782
Elantra	7342221
H1	5965726
Camry	5480933
Prius	5247317
Urus	5237676
Tucson	4891511
GX 460	3714700
E 350	3330855

Figure 4: Monitoring dashboard implementation

For visualization, Superset provides a diverse set of chart types, including bar charts, histograms, line charts, pie charts, scatter plots, and geospatial maps. These were employed to extract actionable insights from the dataset. For example, bar charts were used to compare the total number of companies across different states, while histograms were applied to examine the distribution of discounts applied to vehicle sales, Figure 5. Geographical maps were employed to illustrate the spatial distribution of companies based on state and city, providing a clear overview of regional patterns [6].



Companies by City

city	COUNT(id)
Washington	5
Houston	5
Phoenix	3
Atlanta	3
Trenton	2
Toledo	2
Stockton	2
Richmond	2
Pittsburgh	2
Orange	2

Companies by State

state	company_count
Texas	15
California	12
Georgia	6
Ohio	5
Minnesota	5
Florida	5
District of Columbia	5
Arizona	4
Pennsylvania	3
Nebraska	3

Figure 5: Visualization of data from diverse data sets

Utilization of Superset enhanced the efficiency and clarity of the data analysis process. By combining the flexibility of SQL-based querying with advanced visualization options, Superset served as a powerful tool for both exploratory analysis and the presentation of results. The visualizations generated through this process provided valuable insights into the dataset, forming a foundation for further interpretation and discussion of trends in sales, customer demographics, and company distribution.

6. Advantages of the Proposed Solution

The proposed architecture integrates modern cloud-native technologies to achieve real-time data ingestion, processing, and visualization. It provides a scalable and flexible framework for handling dynamic and heterogeneous data streams, which is particularly relevant in applied data-driven systems. The main advantages are outlined below:

- **Reproducible and automated deployment:** terraform scripts enable automated, consistent, and replicable infrastructure provisioning, ensuring transparency and facilitating reproducible research experiments [15].
- **Real-Time visualization:** Processed data is continuously loaded into target tables and visualized through Apache Superset, offering interactive, near real-time dashboards for decision-making.
- **Low-Latency data ingestion**: Kafka and Debezium enable scalable, event-driven streaming with change data capture (CDC), ensuring timely updates without batch delays.
- **Scalable data transformation**: Azure Databricks provides a unified platform for streaming and batch data processing, auto-scaling to handle variable workloads, and enabling direct integration with machine learning models for advanced transformations.
- **End-to-End integration:** the fully integrated ETL pipeline minimizes latency between data collection, processing, and visualization, supporting time-sensitive research and adaptive decision-making.

7. Limitations while using the proposed solution

While the proposed architecture offers significant advantages in terms of real-time data processing, change tracking, and visualization, it also comes with several limitations.

- **Latency and throughput trade-offs:** although the system is designed for real-time or near-real-time performance, actual latency depends on multiple factors, including:
  -The frequency of changes in the source database
  -The performance of the Kafka cluster
  -Processing complexity in the stream processor
  -Query load on the analytics backend
- **System complexity and maintenance overhead:** the solution relies on multiple interdependent components (Debezium, Kafka, Databricks, Apache Superset, etc.), each

requiring configuration, monitoring, and scaling. This increases the system's operational complexity and may require a dedicated team with expertise in distributed systems, data streaming, and cloud infrastructure.

- **Apache superset limitations:** while Apache Superset offers robust visualization features, it has some limitations. Real-time streaming visualization is limited; data must be periodically refreshed from the analytical store. High-frequency dashboard queries can strain the backend data store.
- **Limited use for low-volume or static datasets:** the system is best suited for high-change, high-volume environments. For static datasets or infrequent data updates, the overhead of real-time infrastructure may not be justified and could lead to unnecessary resource consumption.

8. Conclusions

The results obtained from the practical solutions real-time data monitoring solutions is becoming increasingly vital in modern data-driven environments where rapid response and continuous insight are key to maintaining operational efficiency and competitive advantage. This paper has presented a comprehensive architecture that leverages CDC for low-latency data extraction, Apache Kafka for scalable data transport, and Apache Superset for dynamic data visualization. The proposed solution enables end-to-end visibility of data changes and empowers stakeholders to make informed decisions based on up-to-date information.

The integration of modular components such as Debezium, Kafka Connect, and stream processors offers flexibility and scalability. Additionally, Apache Superset provides an accessible platform for non-technical users to explore and monitor real-time metrics through interactive dashboards. With continued advancements in stream processing and data observability, real-time analytics will play a central role in the evolution of data engineering practices.

Future work will focus on extending the analytical framework developed with Apache Superset to enhance its scalability and integration of advanced machine learning (ML) and predictive analytics. While Superset excels at descriptive and diagnostic analytics, the limitations of advanced data modelling could be overcome by coupling it with ML models hosted in external platforms such as Databricks, AWS SageMaker, or Azure Machine Learning and that way enabling predictive and prescriptive insights.

By integrating Superset with automated reporting tools, alerting systems, and audit mechanisms, organizations could transition from static analytics toward a proactive, continuously monitored environment.

References:

[1] A.S.Rao, M. Radanovic, Y. Liu, S. Hu, Y. Fang, K. Khoshelham, M. Palaniswami and T. Ngo. "Real-time monitoring of construction sites: Sensors, methods, and applications" Elsevier Automation in Construction , Vol. 136 (April 2022). doi.org/10.1016/j.autcon.2021.104099.

[2] Dhamotharan Seenivasan and Muthukumaran Vaithianathan. "Real-Time Adaptation: Change Data Capture in Modern Computer Architecture", ESP-IJACT, Vol. 1, Issue 2, (2023). pp.49-69, doi: 10.56472/25838628.

[3] Ankoriion, Itamar." Change Data Capture Efficient ETL for Real-Time BI" Scholarly Journal, Vol. 15, Issue 1, New York, (2005).

[4] D. M. Tank, A. Ganatra, Y. P. Kosta and C. K. Bhensdadia, "Speeding ETL Processing in Data Warehouses Using High-Performance Joins for Changed Data Capture (CDC)", IEEE, 2010 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, India, 2010, pp. 365-368, doi: 10.1109/ARTCom.2010.63.

[5] Denny, I. P. M. Atmaja, A. Saptawijaya and S. Aminah, "Implementation of change data capture in ETL process for data warehouse using HDFS and apache spark," 2017 International Workshop on Big Data and Information Security (IWBIS), Jakarta, Indonesia, 2017, pp. 49-55, doi: 10.1109/IWBIS.2017.8275102.

[6] G. H. Soares and M. A. Brito, "Business Intelligence Over and Above Apache Superset," 2023 18th Iberian Conference on Information Systems and Technologies (CISTI), Aveiro, Portugal, 2023, pp. 1-6, doi: 10.23919/CISTI58278.2023.10211907.

[7] Darshan M. Tank. “Reducing ETL Load Times by a New Data Integration Approach for Real-time Business Intelligence”, International Journal of Engineering Innovation & Research Vol. 1, Issue 2, 2012, ISSN : 2277 – 5668

[8] M. K. Bali, A. Mehdi and M. Singh, "Implementation of AWS Cloud Infrastructure," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-8, doi: 10.1109/ICCCNT56998.2023.10307993.

[9] A. Trajkovska, B. Ristevski, T. Trajkov, N. Rendevski and K. Veljanovska, "From Data to Decisions: Real-Time Analytics and ML with Kafka and Databricks," *2025 60th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, Ohrid, North Macedonia, 2025, pp. 1-4, doi: 10.1109/ICEST66328.2025.11098429.

[10] A. Sayar, Ş. Arslan, T. Çakar, S. Ertuğrul and A. Akçay, "High-Performance Real-Time Data Processing: Managing Data Using Debezium, Postgres, Kafka, and Redis," *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Sivas, Turkiye, 2023, pp. 1-4, doi: 10.1109/ASYU58738.2023.10296737.

[11] Hosseinzadeh, M., Azhir, E., Ahmed, O.H. *et al.* Data cleansing mechanisms and approaches for big data analytics: a systematic study. *J Ambient Intell Human Comput* **14**, 99–111 (2023). https://doi.org/10.1007/s12652-021-03590-2

[12] D. Junaydullaev, S. Tursunov and A. Rashidov, "An Approach Based on Data Profiling at the Preparing a Dataset for Cleaning," *2025 International Russian Smart Industry Conference (SmartIndustryCon)*, Sochi, Russian Federation, 2025, pp. 578-583, doi: 10.1109/SmartIndustryCon65166.2025.10986179.

[13] Ayesga Nazir. “Data Visualization Tolls for Big Data Analysis: Enhancing Insight and Decision-Making”. Vol. 7 No. 02 (2024): Computer Science Bulletin , Research Articles (2024), ISSN: 2959-5347.

[14] Giacomo Galliano. The importance of data visualization tools in modern enterprises. Cost-effective solutions and empowering of an open-source project. Rel. Paolo Garza. Politecnico di Torino, Corso di laurea magistrale in Ingegneria Informatica (Computer Engineering), 2023.

[15] Venkata Ramana Gudelli. “Cloud Formation and Terraform: Advancing Multi-Cloud Automation Strategies”, JIRMPS, Vol. 11, Issue 2, 2023, ISSN:2349-7300, pp.1-10.