Standardization of ETL Process

Rashmi Phalnikar
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT WorldPeace University Pune, India

Utkrist Agrawal
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT WorldPeace University Pune, India

Prabhat Panwar
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT WorldPeace University Pune, India

Mehul Pansari
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT WorldPeace University Pune, India

Nikunj Padia
School of Computer Engineering and Technology
Dr. Vishwanath Karad MIT WorldPeace University Pune, India

Abstract- The ETL (Extract, Transform, Load) standardization project aims to develop an integrated solution that enables efficient management, processing, and analysis of large volumes of data specific to company's data requirements. The project includes researching and developing an ETL application along with standardization module for interconversions of various data formats, leverages ETL tools and technologies to extract data from various sources, transform it into a format suitable for analysis, and load it into a data warehouse. The project also includes the development of a user-friendly data visualization tool that provides intuitive and interactive access to the data warehouse's contents. The project's objective is to enable the company to make informed decisions based on accurate and up-to-date data, leading to improved business performance and maintain data coming from various sources. The Proof of Concept and research is aimed at arriving at an all-in-one application similar to other Business Intelligence tools available in the market, but is lightweight and specific to organization's technological and data requirements. The project's scope covers the entire ETL, data warehousing, and visualization process, including requirements gathering, design, development, testing, and deployment. The project utilizes an Agile development methodology to ensure flexibility and adaptability to changing requirements and stakeholders' feedback. Overall, the ETL, visualization, and data warehousing project addresses the critical need for effective data management, processing, and visualization in today's data-driven business landscape.

Keywords- ETL, Data Warehouse, standardization, Business Intelligence, visualization

I. INTRODUCTION

ETL stands for Extract, Transform, Load. It is a data integration process that involves extracting data from various sources, transforming it into a consistent format, and loading it into a target database or data warehouse for analysis.

The Extract phase involves pulling data from different sources such as databases, spreadsheets, or web APIs. The data can be stored in various formats, and it may be structured or unstructured. The Transform phase involves cleaning and processing the extracted data into a format that is suitable for analysis. This phase can include several steps such as data cleansing, normalization, aggregation, and filtering. The goal of this phase is to ensure that the data is in a consistent format and is suitable for analysis. The Load phase involves loading the transformed data into a target database or data warehouse.

This phase may involve defining data schema, mapping source data to target data, and optimizing data load performance.

Overall, the ETL process is critical for organizations that need to integrate data from multiple sources to gain valuable insights. By following a systematic approach to data extraction, transformation, and loading, organizations can ensure that their data is consistent and accurate, making it easier to analyze and draw insights from.

Data standardization is the process of transforming data into a consistent format that conforms to a set of predefined rules or specifications. This process involves identifying and resolving differences in data formats, values, and semantics to ensure that the data is accurate, reliable, and comparable.

Standardization is crucial when dealing with data from multiple sources, as data may be stored in different formats, using different units of measurement, or with different codes or abbreviations. Without standardization, it can be challenging to compare data across different sources or to analyze data consistently. Data standardization typically involves several steps, including data cleansing, normalization, and formatting. Data cleansing involves removing or correcting invalid, incomplete, or inconsistent data. Normalization involves converting data into a common format or structure, such as converting dates to a standardized format. Formatting involves ensuring that data is presented consistently, such as using the same units of measurement or abbreviations. Overall, data standardization is critical for ensuring data consistency, accuracy, and comparability, which are essential for effective data analysis and decision-making. By standardizing data, organizations can make better-informed decisions.

Data visualization and reporting are crucial components of ETL (Extract, Transform, Load) processes. Once data has been extracted, transformed, and loaded into a database, it needs to be visualized and reported in a way that is easy to understand and analyze. Data visualization involves presenting data in a graphical or pictorial format, such as charts, graphs, or maps. The goal of data visualization is to make it easier for users to understand complex data and identify patterns, trends, or outliers. By using the right visualization and reporting tools, organizations can create a data-driven culture, improve data quality, and optimize their decision-making processes.

This project aims to build an Extract, Transform, and Load (ETL) application using ReactJS and Django, and perform standardization of given data file into user-specified data format. The application will be used to extract data from different sources, transform it, and load it into a target data store. The application enables the user to import data from any data source such as online or offline data files, process the data and convert it into standard format as per user requirement/format such as JSON, Excel, CSV, etc., all in one application for company's internal data requirements.

The project will have two parts: a front-end built using ReactJS and a back-end built using Django. The front-end will provide a user-friendly interface for users to configure and run ETL jobs. Users will be able to specify the source data, target data store, and any necessary transformations using a checkbox interface. The back-end will be responsible for executing the ETL jobs specified by the user. It will handle the extraction of data from various sources, perform the necessary transformations on the data, and load the transformed data into the target data store. The back-end will also provide an API for the front-end to communicate with it, and download the data files after all the operations are successfully completed, followed by visualization module to create interactive dashboards for uploaded data.

The domain for creating an ETL application along with data standardization module followed by visualization application would be software engineering, web application development, data engineering and data analytics. Data analytics involves the process of examining and interpreting large data sets to derive insights and make informed decisions. In this project, the ETL application would be used to extract data from various sources, transform it, and load it into a target data store. The visualization application would then be used to analyze and present the data in a user-friendly format. The project would involve dealing with data in different formats and from various sources, performing data cleaning and transformation operations, and storing the transformed data in a database.

II. LITERATURE REVIEW

Various research papers are being studied to arrive at Proof of Concept (POC) for ETL standardization application, whose topics were ranging from research on ETL tools to Data warehouse and data loading after standardization.

[1] S.Sajida, Dr.S. Ramakrishna in their paper entitled "A Study of Extract–Transform– Load (ETL) Processes", has stated In Warehouse environment, ETL processes constitute the integration layer which aims to pull data from data sources to targets, via a set of transformations. This paper focused on ETL, the backstage of DW, and presents the research efforts and opportunities. [2] In a paper published by Manish Manoj Singh, he discusses various ETL Tools Available in the Market. A huge piece of BI frameworks is a well-performing implementation of the ETL process, and focuses on the best ETL Tools and which tool can be the best for the ETL process, and performed comparison of different ETL tools.

[3] About the Performance comparison between Talend-Data Integration and other ETL tools, the authors in their paper "Overview of ETL Tools and Talend-Data Integration" declared that In BI an ETL tool helps to extract the data from one or more sources, cleanse it and loads the date into data warehouse. In data integration techniques, the ETL method is important.[4] The paper "Data Integration in ETL Using TALEND" describes the various steps involved in integrating data from various sources using the ETL process, how the Talend Open Studio acting as a Data Integration and ETL tool helps in transforming heterogeneous data into homogeneous data for easy analysis. In his research, the author Sergio Luján-Mora on [5] A UML Based Approach for Modeling ETL Processes in Data Warehouses in the year 2003 proposed an approach involving using UML diagrams, such as activity diagrams, use case diagrams, and class diagrams, to model different aspects of ETL processes.[6] Esmail Ali observed the role and importance of data warehousing in today's business landscape. The most popular data model for a DW is a multi-dimensional model, which can exist in the form of a star schema, a snowflake schema, or a fact constellation schema . [7] The Real-Time Data Warehouse Loading is focused on four major areas: Data warehouse schema adaptation; ETL loading procedures; OLAP query adaptation; and DW database packing and reoptimization. The paper proposes a methodology of creating a replica of each table in the data warehouse, which is initially empty and has no constraints.[8] An ETL Strategy for Real-Time Data Warehouse by Zhou, H., Yang, D., Xu, Y. explains the components of RTDW, including real-time behaviour and data warehousing, and highlights the importance of ETL in establishing and maintaining the data warehouse. The paper also discusses the challenges of capturing changed data in real-time and provides examples of message queues, database triggers, or streaming technologies.[9] The objective of the research paper "JSON Integration in Relational Database Systems" by Dušan Petković is to explore the integration of JSON data format into relational database systems. The paper aims to investigate the challenges and benefits of incorporating JSON data into a RDBMS, such as MySQL, Oracle, etc.

[10] The objective of this text is to highlight the challenges posed by the increasing volume and complexity of data on substances and materials properties, and to propose a set of solutions based on Big Data technology that can help to integrate diverse resources belonging to different organizations and states. While [11] paper is designed to allow continuous data processing, allowing data to be analysed in real-time as it arrives, rather than being processed in batches at predetermined intervals. The platform consists of three main components: a data collector, a data transformer, and a data analyser. These components work together to collect data from a variety of sources, transform the data to a format. The paper [12] "Modelling ETL Processes of Data Warehouses with UML Activity Diagrams" presents a case study of the proposed methodology applied to a real-world data warehouse. The authors use UML Activity Diagrams to model the ETL process of the data warehouse, including data extraction, transformation, and loading. In [13], a case study of the EMD methodology applied to a real-world data warehousing scenario. It introduces the EMD methodology, which is based on a graphical notation for representing ETL processes using entities, attributes, and relationships. Demonstrates how EMD can be used to automate the ETL process, while [14] "Conceptual data warehouse modeling" by Panos Vassiliadis describe the process of designing a conceptual schema that includes facts, dimensions, and hierarchies.[15] In "Research on Extract,

Transform and Load (ETL) in Land and Resources Star Schema Data Warehouse (2013)", authors provide an overview of the Land and Resources Star Schema data model and the requirements for the ETL process. The paper discusses the challenges associated with the ETL process, such as data consistency, data accuracy, and data security. The [16] to proposes a graphical conceptual model called the Dimensional Fact model, and a semi-automated methodology to build it from pre-existing ER schemes or RDBMS, for designing data warehouse (DW) systems.

III. PROJECT REQUIREMENTS

The scope of creating an ETL application and a visualization application can be defined as follows:

1. ETL Application: The ETL application will be responsible for extracting data from various sources, transforming it, and loading it into a target data store. The scope of the ETL application includes identification and integration of data sources, i.e. The application will need to be able to connect to various data sources such as databases, APIs, and file systems, and integrate the data into a unified format, data transformation to perform various operations such as cleaning, filtering, and aggregating the data to ensure that it is consistent and accurate and finally data loading into a data store such as a database or data warehouse.

2. Data Standardization: The ETL standardization application would allow the user to convert the data file into specified data format such as JSON, CSV, Excel through automated ETL pipeline, and download the newly formatted file into local machine.

3. Visualization Module: The visualization application will be responsible for presenting the transformed data in a user-friendly format. The scope of the visualization application includes data exploration using various interactive visualizations such as charts, graphs, and tables, and customization of the visualizations to select the data they want to display.

The scope of the project includes the integration of various functionalities such as live data connection using APIs and importing large amount of data using different techniques. The project will involve selecting and implementing appropriate technologies as well as libraries for the ETL and visualization applications, designing and implementing user interfaces, and testing and debugging the applications. The ETL standardization application is bounded by several limitations for the current scope of the project which includes software and hardware limitations, data integrity and security issues, dynamic requirements of the organization and time constrains.

3.1 Software Requirements

ETL (Extract, Transform, Load) visualization applications are designed to help users visualize and analyze data during the ETL process. These applications typically require the following software requirements:

1. Programming Languages: The given ETL standardization application requires knowledge of programming languages like SQL, Python, and Frontend programming language like ReactJS to manipulate data, write scripts, or develop custom solutions. The project is built using Django as backend language and ReactJS and its various libraries as Frontend programming language.

2. Database Management Systems: ETL standardization application require a database management system (DBMS) to store and manage data. Popular DBMSs include Oracle, SQL Server, MySQL, and PostgreSQL.

3. Business Intelligence Tools: ETL visualization module may require business intelligence tools to generate reports and visualizations based on the extracted data. Popular business intelligence tools include Tableau, QlikView, and Power BI. For our application, we have built a separate visualization tool using ReactJS, thus covering the need of using any third-party BI tool.

4. Data Warehousing Tools: ETL visualization applications may require data warehousing tools to create and manage a data warehouse. Popular data warehousing tools include Snowflake, Amazon Redshift, and Google BigQuery. For the current scope of the project, we have used MySQL database to load the data and use it for visualization purpose. The application may require data quality tools to ensure the data being extracted, transformed, and loaded is accurate and consistent.

3.2 Hardware Requirements

The hardware requirements for an ETL visualization application project will depend on several factors, such as the size and complexity of the data being managed, the number of users accessing the application, and the performance requirements of the application. Here are some general hardware requirements to consider:

1.ETL processes can be CPU-intensive, so it is recommended to have a multi-core CPU with a clock speed of at least 2 GHz or higher. The amount of RAM required will depend on the size of the data being managed and the number of users accessing the application. A minimum of 8GB of RAM is recommended, but larger datasets may require 16GB or more.

2.The storage requirements will depend on the size of the data being managed. Consider using fast SSD drives for improved read/write performance. Implementing a backup and recovery strategy to ensure data is protected against hardware failures or other disasters is also an important requirement.

3.A fast and reliable network connection is essential for efficient data transfer between systems, performing effective ETL process and load data quickly into the warehouse. Also, If the ETL visualization module may include complex visualizations or 3D graphics, a dedicated graphics card with at least 2GB of VRAM would be required.

3.3 Functional Requirements

The ETL standardization application uses Django as the backend programming language, ReactJS as the front-end language, Microsoft SQL for data loading and SQLite for application data storage. The application will enable users to select the data source of their choice, extract it, and then transform it into a format that

can be easily loaded and analysed. The definition of the source connection will be stored and saved for later use. The transformed data will also need to be stored in the database. The React front-end will be used for interacting with the data and performing various analysis and visualization tasks using ReactJS chart APIs, while the Django back-end handles the data processing and storage.

3.3 Non-functional Requirements

The application allows users to easily manage and organize their data, making it a valuable tool for company that need to extract insights from any amount of data, and perform basic validation on input data sources. The application needs to be scalable, secure and able to handle large volumes of data of any available format. The data can be imported in two ways: Batch-import or Real-time stream import from online/offline sources. The application must be interactive, secure and suitable for company's data requirements and flexible enough to adjust to changing requirements in future.

IV. SYSTEM ARCHITECTURE

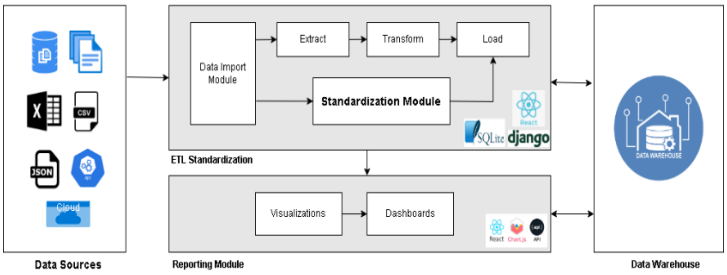

Fig 1. Architecture Block Diagram for ETL Standardization Application

The block diagram for the ETL standardization application Fig. 1 includes various components working in sequence corresponding to Data sources (include databases, files, APIs, web services, and other sources of data. Data sources may be located on-premises or in the cloud), Extract component(extracts data from the various data sources and prepares it for transformation), Transform component(it applies business rules, data cleaning, data enrichment, and other data processing operations to the extracted data), Load component(loads the transformed data into the target system, such as a data warehouse, data lake, or other analytical system)and the Reporting module(provides a user interface for users to interact with the data and generate reports, dashboards, and other visualizations). Various UML diagrams to support the design and development of the application are being ascertained and revied to rule out any disparity while developing the application.

V. IMPLEMENTATION DETAILS

5.1 Technological Details

The discussed ETL application is being developed using specific programming languages and their libraries based on the requirements setup by the stakeholders of the application and technological feasibility study conducted to ascertain the best performance and useability of the application in the business domain, and efforts are being made to make the application most cost effective and scalable. The following tools and technologies are being used in development of the ETL application: -

• Django: Django is a popular web framework written in Python that can be used for building ETL application. Django comes with built-in support for interacting with various data sources and the Django framework can be used in conjunction with libraries such as Pandas, NumPy, and SciPy to perform various data transformation operations. Django comes with built-in support for various databases, such as PostgreSQL, MySQL, SQLite, and Oracle. Developers can also use Django ORM (Object-Relational Mapping) to interact with databases and load transformed data. Overall, Django can be a good choice for building ETL applications due to its ease of use, flexibility, and scalability.

• ReactJS: ReactJS is a popular JavaScript library for building user interfaces. It can be used in conjunction with other technologies to build ETL and data visualization applications. ReactJS provides a rich set of UI components that can be easily customized and combined to build rich and responsive user interfaces, and it can be used in conjunction with data visualization libraries such as D3.js and Chart.js to create interactive data visualizations. The state management system of ReactJS can be used to manage the state of the ETL process and the data visualization components.

• MS SQL Server: It is a popular relational database management system that can be used for data loading in ETL processes. MS SQL Server is designed to be scalable and can handle large amounts of data. It can be used to build ETL pipelines that can handle high volumes of data, and it provides built-in support for encryption, user authentication, and authorization.

• SQLite: SQLite is a lightweight and popular database management system that is widely used in web development projects. It's also one of the default database engines supported by Django, which is a popular Python-based web framework.

• APIs: Chart.js is a popular open-source library for creating charts and graphs using JavaScript. It's easy to use and highly customizable, making it a great choice for data visualization in ReactJS applications.

5.2 Methodology

The implementation details of an ETL (Extract, Transform, Load), visualization, and data standardization application may have several pre-requisites to be considered. Firstly, to create front-end of the application, we have created UI/UX prototype of the application using the Balsamiq wireframing tool available online. After creating wireframes, the work on frontend of the web application started using ReactJS along with developing the backend of the application for ETL pipeline and data standardization using Django, keeping in mind the UML diagrams and workflow of the user's interaction while using the final product. Followed by this, the data loading module was developed and tested on the MS SQL server for various data sources of varying formats, and then the visualization module was created using Chart.js API for ReactJS, allowing the user to create interactive visualizations for the loaded

data and saving them for future use. Here are some common implementation details that can be considered for the project:

1.ETL Implementation:

• Design the ETL workflow: Plan and design the ETL workflow based on the data sources, data types, and data volumes.

• Extract phase: The Extract phase of the ETL (Extract, Transform, Load) process involves retrieving data from various sources and bringing it into a central location, such as a database or data warehouse, for further processing. The Extract phase is crucial to the ETL process because it sets the foundation for the rest of the data processing. The first step in the Extract phase is to identify the data sources that need to be extracted. These sources could be databases, files (CSV, Excel, JSON), APIs, or web services. Once the data has been extracted, it is important to validate and cleanse it to ensure its accuracy and consistency. This involves checking for errors, duplicates, and missing data, and correcting any issues that are identified and finally setting the stage for the Transform and Load phases of the ETL process

• Transform phase: The Transform phase of the ETL (Extract, Transform, Load) process involves cleaning, enriching, and reshaping the extracted data into a format that can be easily loaded into the target system. Some basic transform operation performed by our application includes selecting specific columns for consideration, removing duplicate values, sorting the data, handling the missing values by statistical method, and case conversion of specific columns.

• Loading phase: The Load phase of the ETL process involves loading the transformed data into the target system, here it is MS SQL database, but data warehouse can be considered for future scope. The first step in the Load phase is to define the target schema, which is the structure of the target system where the data will be loaded. This involves creating tables, defining fields, and specifying data types. After the target system has been set up, the next step is to load the transformed data into the target system. Test the ETL workflow to ensure that it is accurate, efficient, and meets the security aspects of web application development.

2. Data Standardization Implementation:

• Define the data standardization requirements based on the company's needs and data sources. Plan and design the data standardization workflow based on the data sources and data types.

• Test and refine the data standardization workflow: Test the data standardization workflow to ensure that it is accurate, efficient, and meets the organization's needs.

3. Data Visualization Implementation:

• Develop the visualization module using ReactJS as frontend programming language and leveraging the benefits of Chart.js API in creating the charts.

• Connect the data visualization tool to the database: Configure the data visualization tool to connect to the database and retrieve the required data.Design and develop the data visualizations.

5.3 Performance Metric

Criteria	Summary	Results
Data completeness	Measures the percentage of data that was successfully extracted, transformed, and loaded into the target system.	The application worked perfectly fine for considerable amount of data (~100mb)
Data accuracy	This metric measures the degree to which the data in the target system reflects the original source data	The transformed data is accurate up to the changes made manually by the user.
Data consistency	Measures the degree to which the data in the target system is consistent across different sources and time periods	The standardization module successfully performs conversions to different data formats of varying size.
Processing time	The time it takes to complete each phase of the ETL process, from data extraction to data loading.	Loading large amount of data takes some time as connection has to be maintained with database server.
Visualization response time	The time it takes for the system to generate and display visualizations based on the data.	Charts are generated quickly from the loaded data in the database.

Table 1. Performance Metrics

VI. RESULTS AND DEPLOYMENT

The final results of the ETL standardization project can provide numerous benefits to stakeholders and company after incorporating almost all the future prospects and use cases as already mentioned in the requirements, following are some of the benefits the ETL application would provide to the end-users once deployed:

1. Accurate Data: The ETL application extracts data from various sources, transforms it into a standard format, and loads it into a database, ensuring that the data is accurate, consistent, and reliable.

2. Improved Data Analysis: The data standardization component of the application ensures that the data is comparable and can be easily analyzed. The data visualization tools enable stakeholders to view the data in a way that is easy to understand, identify trends and patterns, and make informed decisions.

3. Enhanced Efficiency: The ETL application automates the data processing, reducing manual effort and errors. The data standardization component eliminates the need for manual data cleaning, saving time and resources. The data visualization tools enable stakeholders to quickly access the data they need and make informed decisions faster.

4. Better Decision Making: With accurate, standardized data and powerful visualization tools, stakeholders can make more informed decisions. The data can be used to identify trends, predict outcomes, and optimize operations, resulting in better business outcomes.

Deploying the given application on-premise/cloud environment includes the consideration of various factors such as choosing a deployment environment, which can be on-premise, cloud-based, or a hybrid model,

ensuring that the data sources are properly configured and that the application can connect to them, deployment involves setting up the ETL processes, data standardization, and loading the data into the data warehouse environment, and creating visualizations after connecting the module to stored data. It is essential to test the deployment thoroughly to ensure that the application is functioning correctly and meeting the organization's needs. Once the deployment is complete, it is essential to maintain and monitor the application regularly. This includes monitoring performance, ensuring data accuracy, and making updates and changes as needed.

VII. CONCLUSION AND FUTURE PROSPECTS

In conclusion, an ETL (Extract, Transform, Load) application project that includes data standardization and visualization components can be a valuable tool for organizations to gain insights from their data. The project involves extracting data from various sources, transforming it into a useful format, standardizing the data, and loading it into a database. With this foundation, powerful data visualizations and reports can be created to help users better understand and analyze the data. Successfully implementing this project requires a deep understanding of data processing, database management, data standardization techniques, and visualization tools. It also involves careful planning, task prioritization, and efficient execution to ensure that the project achieves its goals. Following are some of the future scopes the project:

• Real-time Data Processing: As the demand for real-time data processing continues to grow, the ETL application can be enhanced to support streaming data, enabling company to analyze large volume of online data in real-time.

• Cloud Computing: The ETL application can be deployed in the cloud, allowing company to scale their data processing and storage needs on-demand and reducing the costs associated with maintaining on-premise infrastructure.

• Automated Data Quality Checks: The ETL application can be enhanced to include automated data quality checks, ensuring that data is consistent, accurate, and complete, reducing errors and inconsistencies in analysis.

• Incorporating more transformation operations and providing more dynamic visualization options to the user for creating interactive dashboards.

• The data source security validation and access control is under consideration for the application’s security needs.

Overall, an ETL, visualization, and data standardization project has the potential to provide significant value to the company by helping them make better-informed decisions based on their data. With the right approach, tools, and skills, organizations can successfully build a robust ETL application that meets their specific needs, empowers their users to gain valuable insights from their data, and ensures that their data is accurate, reliable, and comparable.

VIII. REFERENCES

1] S.Sajida, Dr.S.Ramakrishna, 2015, A Study of Extract Transform Load (ETL) Processes, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCACI

2] Manish Manoj Singh,2022,Extraction Transformation and Loading (ETL) of Data Using ETL Tools, IJRASET, ISSN : 2321-9653

3] Kraetz, D., Morawski, M. (2021). Architecture Patterns—Batch and Real-Time Capabilities. In: Liermann, V., Stegmann, C. (eds) The Digital Journey of Banking and Insurance, Volume III. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-030-78821-6_6

4] J. Sreemathy, R. Brindha, M. Selva Nagalakshmi, N. Suvekha, N. Karthick Ragul and M. Praveennandha, "Overview of ETL Tools and Talend-Data Integration," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1650-1654, doi: 10.1109/ICACCS51430.2021.9441984.

5] J. Sreemathy, I. Joseph V., S. Nisha, C. Prabha I. and G. Priya R.M., "Data Integration in ETL Using TALEND," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 1444-1448, doi: 10.1109/ICACCS48705.2020.9074186.

6] Trujillo, J., Luján-Mora, S. (2003). A UML Based Approach for Modeling ETL Processes in Data Warehouses. In: Song, IY., Liddle, S.W., Ling, TW., Scheuermann, P. (eds) Conceptual Modeling - ER 2003. ER 2003. Lecture Notes in Computer Science, vol 2813. Springer, Berlin, Heidelberg.

7] Vyas, Dr Sonali & Vaishnav, Pragya. (2017). A comparative study of various ETL process and their testing techniques in data warehouse. Journal of Statistics and Management Systems. 20. 753-763. 10.1080/09720510.2017.1395194.

8] M. Golfarelli, D. Maio and S. Rizzi, "Conceptual design of data warehouses from E/R schemes," Proceedings of the Thirty-First Hawaii International Conference on System Sciences, Kohala Coast, HI, USA, 1998, pp. 334-343 vol.7, doi: 10.1109/HICSS.1998.649228.

9] Shaker H. Ali El-Sappagh, Abdeltawab M. Ahmed Hendawi, Ali Hamed El Bastawissy, A proposed model for data warehouse ETL processes,Journal of King Saud University - Computer and Information Sciences,Volume 23, Issue 2,2011

10] M. Mrunalini, T. V. S. Kumar and K. R. Kanth, "Simulating Secure Data Extraction in Extraction Transformation Loading (ETL) Processes," 2009 Third UKSim European Symposium on Computer Modeling and Simulation, Athens, Greece, 2009, pp. 142-147, doi: 10.1109/EMS.2009.111.

11] A. Simitsis, P. Vassiliadis and T. Sellis, "Optimizing ETL processes in data warehouses," 21st International Conference on Data Engineering (ICDE'05), Tokyo, Japan, 2005, pp. 564-575, doi: 10.1109/ICDE.2005.103.

12] Muñoz, L., Mazón, JN., Pardillo, J., Trujillo, J. (2008). Modelling ETL Processes of Data Warehouses with UML Activity Diagrams. In: Meersman, R., Tari, Z., Herrero, P. (eds) On the Move to Meaningful Internet Systems: OTM 2008 Workshops. OTM 2008. Lecture Notes in Computer Science, vol 5333. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88875-8_21

13] aisel.aisnet.org/hicss-50/st/big_data_engineering/2/

14] Abdeltawab M.A. Hendawi and Shaker H. Ali El-Sappagh, EMD: entity mapping diagram for automated extraction, transformation, and loading processes in data warehousing, Published Online:May 25, 2012pp 255-272, https://doi.org/10.1504/IJIIDS.2012.047003

15] Ankorion, Itamar,Change Data Capture Efficient ETL for Real-Time BI, www.proquest.com/openview/ebb6b945abb5673c78aa513ff8c9489c/1?pq-origsite=gscholar&cbl=51938

16] Ricardo Jorge Santos, Jorge Bernardino Real-time data warehouse loading methodology, IDEAS '08: Proceedings of the 2008 international symposium on Database engineering & applications

17] Esmail Ali, F.S. (2014). A Survey of Real Time Data Warehouse and ETL. International Scientific Journal of Management Information Systems, 9 (3), 03-09