Report for the Degree of Master of Computer Science

**End to End Automation in ETL Pipeline**



**Pikesh Maharjan**

**(LC00015002358)**

**Masters in Computer Science (MCS) IIMS college**

**Computer Science Department**

**Lincoln University, Malaysia**

**January 2025**

Research project for the Degree of Master of Computer Science

**End to End Automation in ETL Pipeline**

**Supervised by Prof. Sudan Jha, Ph.D.**

A report submitted in partial fulfilment of the requirements for the

degree of Master of Computer Science

**Pikesh Maharjan**

**(LC00015002358)**

**Masters in Computer Science (MCS) IIMS college**

**Computer Science Department**

**Lincoln University, Malaysia**

**January 2025**

List of Figures

[*Fig. 1.1 Illustration of ETL* 5](#_Toc186292381)

[*Fig. 1.2 Illustration of Advanced ETL* 6](#_Toc186292381)

[*Fig. 3.1 Proposed Methodlogy for ETL automation* 18](#_Toc186292381)

Table of Contents

Contents

**No table of contents entries found.**

CHAPTER 1: INTRODUCTION

In the age of digital transformation data has emerged as a very useful as well as critical asset for organizations which helps organization in driving decision-making and providing visualization into customer behavior and market trends. However, the large amount of information generated through different sources leads to challenges in analyzing, transforming and using this information effectively. This is where the Extract, Transform and Load (ETL) process comes into a play. ETL refers to the process of extracting data from multiple sources, transforming it into usable format and loading it into centralized repository such as a data warehouses, data lakes and so on. This process ensures that businesses can take proper data driven decisions by relying on consistent, accurate and updated data for analytics/reporting.

Extract, Transform and Load (ETL) is a process in data management and integration that plays an important role in digital and modern businesses [9]. It is used to develop and improve applications for analysis It involves three primary steps:

Diagram of a diagram of a process

Description automatically generated

*Fig. 1.1 Illustration of ETL [6]*

Extracting data from various sources, transforming it into a suitable format and loading it in a central repository such as a data warehouse for analysis and decision-making [9]. This phase involves gathering data from diverse sources like databases APIs files or cloud storage. These sources can differ in structure such as relational databases spreadsheets or unstructured web data.

Next step includes transformation where raw data is cleaned formatted and standardized based on the business requirements and the repositories where loading is going to take place. Transformations can include removing duplicates, handling missing values and converting data into a format that matches with business requirements.

The last step of the ETL is loading where the processes load the transformed data into a system such as data warehouse or data lake where it becomes accessible to the users that are assigned for advanced analysis, visualization. This process ensures that the data from various sources are ready for analysis, visualization through which various dashboards, visual reports are generated and thus helping in data driven Decision Making Processes [9]. The figure depicting the use of ETL is presented below.

A screen shot of a diagram

Description automatically generated

*Fig. 1.2 Illustration of Advanced ETL [6]*

* 1. Background

In today’s world where every second a huge amount of data is being produced ETL pipelines play a vital role in the capture, transfer and load process. These pipelines are crucial in the field of data management and analysis as they enable the movement and transformation of raw data that is then loaded into databases, data warehouses or other storage systems. This transformed data can be converted into actionable insights that are essential for decision-making processes in any data-driven organization.

Traditional ETL workflows however are often manual, requiring human intervention in various processes such as schema detection classification and classification. Manual involvement makes the process time-consuming and prone to errors. Manual ETL processes can lead to inconsistencies, issues and delays which can affect the correct and timely generation of the relevant information from the data.

With the exponential growth in the volume and variety of data there is a pressing need to automate ETL pipelines to reduce the reliance on human effort. Automated ETL pipelines can handle large amounts of data with numerous fields with minimal human intervention and provide fast reporting to both IT and financial users. This automation reduces time delays and minimizes issues that could arise due to human errors. Automating ETL pipelines has become a necessity to improve efficiency, scalability and reliability.

Automated ETL processes use advanced technologies such as machine learning and artificial intelligence to perform tasks traditionally required human attention. For example, machine learning algorithms can detect and adapt to changes in data schemas automatically. They categorize data types and map data fields accurately. This not only speeds up the ETL process but enhances the accuracy and consistency of data being processed.

Automated ETL pipelines have been shown to be more flexible and adaptable to changing business requirements. They can easily scale to accommodate increasing data volumes and can be quickly adjusted to integrate new data sources or adapt to changes in existing ones. The flexibility is crucial in today's fast-paced business environment where organizations need to be agile and responsive to stay competitive.

In summary, automation of ETL pipelines is essential for modern data management and analytics. It addresses the challenges posed by traditional manual processes such as time consumption errors and scalability issues.

1.2 Statement of the Problem

The manual creation ETL processes has been a persistent challenge in data-driven industries especially with the exponential growth of data. Traditional ETL workflows often require extensive human intervention to handle a variety of tasks. These tasks include file ingestion, where raw data files are introduced into the system; metadata recognition that involves identifying and understanding the structure and properties of the data; mapping fields from source to destination ensuring that data is correctly aligned between different systems.

This dependency on manual processes leads to significant inefficiencies in both time and resource management. Human intervention is not only time-consuming but also introduces a higher probability of errors particularly when dealing with the data in various formats and varying levels. This can result in data inconsistencies that compromise the integrity of the data and the insights derived from it. Another time-consuming process that exists in manual ETL processes is identifying data types in different datasets. Rule-based methods that rely on predefined rules to classify data often lack flexibility to adapt new and evolving data types. These methods require significant maintenance effort to keep up with changes in data structures and formats which makes them less efficient over many years.

While machine learning approaches offer a promising solution to these challenges, they are still underutilized in the ETL domain. The perceived complexity of implementing machine learning models for tasks such as schema detection, data type categorization and transformation logic definition has hindered their widespread adoption. As machine learning technologies continue to advance and become more accessible, there is a growing opportunity to leverage these tools to automate and improve ETL processes.

Another critical issue lies in the lack of dynamic configuration systems that allow seamless integration with user-defined transformation logic. Each time a new dataset is introduced the pipeline code frequently needs to be modified. This process is not only time-consuming but also significantly limits the flexibility of the ETL system. The need to manually adjust the code for each new dataset introduces delays and increases the workload for data engineers making the system less agile and more responsive to changing data requirements.

The result is that there is a significant gap in automating the final loading stage into target systems such as SQL Server. This stage often involves repeated manual validations and correction to ensure that the data is correctly loaded and meets the required standards. These manual interventions can lead to inconsistencies and errors further delaying the availability of data for analysis and decision making. This lack of automation not only slows down the overall ETL process but also increases the likelihood of data inconsistencies which can compromise the quality and reliability of the insights derived from the data.

The present landscape of data management is the heavy dependency on specific ETL (Extract, Transform, Load) tools for executing ETL processes. Most organizations rely on commercial ETL software tools that come with limitations such as high costs limited customization options and vendor lock-in. These tools often require specialized knowledge and training which can be a barrier for smaller organizations or those with limited resources.

The reliance on these tools can also make it difficult to adapt to new data sources or changing business requirements, further limiting the flexibility and scalability of ETL processes. For instance, when a new data source is introduced or an existing one changes, significant modification can be required to the ETL pipeline. This often involves reconfiguring the tool, updating scripts and ensuring compatibility with the new data formats which can be both time-consuming and resource intensive [8].

Furthermore, the high costs associated with commercial ETL tools can be prohibitive for smaller organizations or startups. These costs include not only the initial price or subscription fees but also ongoing maintenance support and training expenses [8]. The financial burden can limit the ability of these organizations to invest in other critical areas like data analytics and business intelligence thereby hindering their overall growth and competitiveness.

Vendor lock-in is another critical issue associated with dependability on specific ETL tools. Once an organization has invested heavily in a particular tool or software, switching to another solution can be challenging and costly. This lock-in can stifle innovation and prevent organizations from adopting more advanced or suitable technologies as they become available. It also limits the ability to customize and optimize ETL processes to meet specific business needs as organizations are often constrained by the capabilities and limitations of the chosen tool.

This research seeks to address these issues by automating the ETL processes without the need of ETL Tools, implementing Machine learning for data type identification, dynamic configuration for user-driven transformation and loading data into the SQL Server.

1.3 Research Questions

The study seeks to address the following questions:

1. How can machine learning methods be integrated to identify data types in ETL pipelines?
2. Is ETL possible without using ETL tools?
3. How can ETL be carried out with minimum human interaction?

1.4 Objectives

General Objective:

1. To design and implement an automated ETL pipeline capable of handling ASCII delimited datasets while ensuring accuracy.

Specific Objectives:

1. Develop mechanisms for data type recognition using ML methods.
2. Create a configuration file to allow human intervention when necessary.
3. Automate the data loading process into SQL Server.

1.5 Significance of the Study

The automation of the end-to-end ETL pipeline will help to transform the traditional methods of data engineering. This research seeks to minimize the necessity for human interventions in data processing and thus minimizing human errors and increasing data integration for decision-making purposes. By automating these processes, the study aims to develop a framework for handling delimited file formats, which involves accurately identifying data types, and implementing configurable transformation logic. These functionalities can be applied in practical scenarios and used in current market trends thus concluding that the research is both relevant and impactful.

Furthermore, this study aims to make a substantial contribution to the easiness and efficiency of ETL operations. As organizations increasingly generate and stores large volumes of data, the capability to manage this data with enhanced reliability and fast processing of the data becomes crucial. Automated ETL pipelines offer faster and more precise data processing compared to manual techniques since the human errors are minimized due to the less human intervention, ensuring that data is accessible for analysis, reporting and visualizations. This improved efficiency will help organizations to make quicker, data-driven decisions, thereby providing them with a competitive advantage in their respective fields.

This research aims to enhance efficiency, providing significant advantages to data engineers, analysts, and businesses by optimizing workflows and improving the overall quality and accessibility of data. Data engineers will be able to allocate more time to strategic initiatives that contribute value to the organization, as they will spend less time on repetitive manual tasks. Analysts will benefit from access to cleaner and more reliable data, which will enable them to produce more accurate insights. Businesses will experience enhanced data quality and expedited decision-making processes, ultimately resulting in improved business outcomes. Also, the need of learning different ETL tools and their dependencies for the ETL process will be removed and thus engineers can focus on the other tasks.

Furthermore, the study intends to examine the current reliance on specific ETL tools, which often entail high costs, limited customization, and vendor lock-in. By creating an automated ETL framework that is both flexible and adaptable, this research will offer organizations a more cost-effective and customizable alternative. This approach will alleviate the financial strain associated with commercial ETL solutions and empower organizations to tailor their ETL processes to suit their unique requirements.

1.6 Scope and Limitations of the Study

The focus of this research is on the creation and implementation of an automated end-to-end ETL (extraction, transformation, load) pipeline. This extensive study includes several essential elements aimed at establishing an efficient ETL process. Initially, it entails the development of a script where we can provide various inputs such as path of the file, Sql server connection details.

Furthermore, the project will integrate machine learning techniques to identify data types, thereby enhancing accuracy and reliability of the data type recognition process and minimizing the need the need of human effort in this step. Various files have large numbers of fields and thus recognizing data type for all these fields requires human effort and consumes huge time which will be eradicated through this scope of the study.

An important component of the research involves the creation of a configuration file that defines metadata, including headers, delimiters, and transformations logics (which would be manual if required). This configuration file will offer a flexible and customizable scope for specifying the required transformations and mappings for the data. To guarantee the accuracy and validity of the configuration, the system will incorporate features that allow for human intervention, enabling users to validate or modify the configuration as necessary. This functionality will empower users to review and adjust the configuration, ensuring the correct processing of the data.

The study offers a thorough solution for the automation of ETL processes focusing on structured file formats, particularly delimited files. The investigation into semi-structured or unstructured data formats will be very less, as the main objective is to create a solution for managing structured data. Additionally, the research will not include real-time data ingestion and processing, as the pipeline is designed for batch processing scenarios.

To uphold the validity and reliability of the research, synthetic data will be utilized throughout the study. This strategy will eliminate the use of sensitive or proprietary data, which could potentially influence the generalizability of the results in specific industry contexts. There will be no User Interface and inputs would be given through the script directly due to the time constraints.

CHAPTER 2: LITERATURE REVIEW

A literature review investigates the existing knowledge related to the study and projects carried out for automation of ETL (Extract, Transform, Load) pipelines. This examination is useful for studying the theoretical frameworks, methodologies, and practical applications presented in previous literature. Multiple studies have been carried out for the eradicating the obstacles and progress in automating data integration and transformation processes, focusing on the increasing significance of scalable and efficient ETL systems within data-centric industries.

This paper concentrates on automating ETL process, where the study seeks to reduce manual intervention throughout the entire process i.e., extracting, transformation and loading. Additionally, this proposal addresses the demands for identifying data types of the multiple sources present in the source file by applying machine learning algorithms. The discourse encompasses various research efforts related to ETL automation, emphasizing the importance of creating efficient ETL pipelines capable of handling data processing in short interval of time without or minimum human intervention.

The work by Embley et al. [10] highlights the potential of ontological conceptual modeling in addressing the challenges of unstructured data on the Web. By focusing on data-rich documents, their approach provides a framework for automating the extraction and structuring of information, which is particularly valuable in domains where data is abundant but difficult to query using traditional methods. This research contributes to the broader field of data extraction by offering a systematic and efficient method for transforming unstructured Web data into structured formats, enabling more effective data utilization and analysis [10]. Another Paper [1] suggests the role of the machine learning in automating the ETL pipeline. Machine learning based data pre-processor is used to pre-process the data more rigorously. It is reducing the processing time significantly and produces a good quality of data from the data warehouse [1]. An architecture is designed to address the challenge faced in the practical application of near real-time ETL processing [1]. Though it has suggested an architecture for automating ETL processes through the help of machine learning techniques, it hasn’t suggested exact methods and machine learning algorithms that can be used for ETL automation processes.

The paper by [11] highlights the significance of improving ETL process system data flows to enhance business investment returns. It emphasizes the role of enterprise-level scheduling solutions that are user-friendly and capable of managing heterogeneous environments, paving the way for automation in ETL processes [11]. Moreover, the study envisions the development of future ETL tools with enhanced support for command-based or script-based automation, offering end-to-end process handling. By leveraging these advanced technologies, enterprises can achieve more efficient data integration, improve operational efficiency, and reduce manual errors in ETL execution [11].

Another paper [3] by Chiara Van der Putten focuses on automating the creation of workflows in SQL Server Integration Services (SSIS) to reduce manual efforts in data loading and integration processes. The research outlines a multi-stage approach, including the development of custom SSIS components, the creation of databases to store user automation needs and workflow modules, and the integration of advanced artificial intelligence models. A question-answering system leverages embedding-based methodologies and generative AI to interpret natural language user requests and generate tailored ETL workflows. The proof of concept demonstrates how AI-driven automation can streamline ETL processes, enabling organizations to handle complex data integration tasks more efficiently while allowing employees to focus on strategic initiatives [3].

The study by Pham [4] provides a valuable reference for understanding the challenges and approaches in automating ETL pipelines, particularly in addressing manual processes and enhancing data extraction capabilities. This thesis builds upon such insights by focusing on automating the entire ETL pipeline with minimal user intervention [4].

The work by Akisetti et al. [12] explores the integration of CI/CD pipelines into ETL workflows to address the inefficiencies of traditional ETL processes in machine learning applications. The study underscores the transformative potential of automating data preparation and processing to ensure a reliable and repeatable pipeline, facilitating seamless collaboration between data engineers and scientists. Similarly, this thesis builds on these principles by automating end-to-end ETL processes while addressing specific challenges such as error handling, real-time processing, and adaptability to user requirements [12].

Previous research has proposed the integration of SQL-based database systems for loading transformed data, with many emphasizing the use of SQL Server for its robust support for batch data processing and reporting capabilities. Additionally, best practices in ETL pipeline design, including modularity, scalability, and error handling, have been extensively documented.

In summary, the literature underscores the importance of designing ETL pipelines that are not only automated but also adaptive, efficient, and user-friendly. This review identifies gaps in practical implementations, such as handling dynamic configurations, using machine learning for data type identifications, automating without need of using ETL tools or any other CI CD pipeline which this research aims to address.

CAHAPTER 3: METHODOLOGY

**3.1 Research Design**

This research follows a quantitative approach to evaluate the effectiveness of automating ETL pipelines using Python, Pandas, and machine learning for data type identification. The study is structured into three phases: system development, data processing, and performance evaluation.

1. System Development – This phase involves designing and implementing an ETL automation framework using Python and Pandas. A machine learning model is integrated to classify and identify data types from various sources, enabling dynamic schema generation and transformation.
2. Data Processing – The implemented system extracts raw data, applies transformations based on the detected data types, and loads structured data into the target system. The effectiveness of the ML model is assessed using a labeled dataset to validate its predictions.
3. Performance Evaluation – The system’s accuracy and efficiency are measured using standard metrics such as precision, recall, and execution time. An iterative approach is followed to refine the workflow, ensuring optimal automation and minimal manual intervention.

3.2 Tools and Frameworks

The primary tools used in this research are Python and Pandas, with machine learning techniques to automate ETL pipeline development. The selected frameworks and libraries enable efficient data extraction, transformation, and schema identification. The key tools include:

Python – The core programming language for scripting ETL automation and integrating machine learning models.

Pandas – Used for data manipulation, transformation, and validation of structured and unstructured data.

Scikit-learn – Employed for training and implementing machine learning models for data type identification.

NumPy – Supports numerical operations, especially in handling large datasets efficiently.

3.3 Data Collection and Preparation

The dataset used in this study consists of structured and semi-structured data sourced from various databases and files. The steps involved in data preparation are as follows:

Data Acquisition:

Collect raw data from different sources such as CSV files, relational databases, and external APIs.

Ensure diverse datasets to test the robustness of the ETL automation system.

Preprocessing:

Handle missing values, inconsistencies, and outliers in the dataset using Pandas.

Normalize and standardize data formats for consistency across different sources.

Apply machine learning techniques to automatically identify data types and detect schema anomalies.

**3.4 System Architecture**

The system architecture consists of the following components:

1. **Data Ingestion**

The process starts with a data ingestion where users can input the location of the file path that needs to be processed. The user also inputs the destination server, database and table details to ensure the file loading to the designated place where the users intends to load the file. The backend validates the file to ensure before proceeding to the next stage.

1. **Meta Data Identification**

To identify Meta data such as column delimiter, row separators, and column data types, a combination of rule-based logic and machine learning techniques is used. The rule-based system examines predefined patterns to identify Meta data. Simultaneously, machine learning models are trained on sample datasets to predict Meta data with higher accuracy, particularly for those with ambiguous cases. This dual approach enhances reliability and adaptability in detecting various Meta data available in the source files.

1. **Configuration File Generation**

Once the data types and other Meta data are identified, a configuration file is generated. This file provides detailed metadata about the dataset, including column names, data types, delimiters, and record separators. The configuration file acts as a blueprint for subsequent transformations and loading operations.

**Human Interaction for Validation**

A provision is made for human users to review and modify the generated configuration file. This optional step is controlled via a check button on the frontend interface. If enabled, users can view the configuration file and adjust, such as adding specific transformation logic or correcting any discrepancies in data type detection. This step ensures greater flexibility and allows customization for unique use cases.

**Incorporating Transformation Logic**

If additional transformation rules are required, users can specify them directly in the configuration file. For example, a user may need to standardize date formats, remove invalid rows, or aggregate data based on specific conditions. These transformations are dynamically parsed and applied to the dataset, ensuring that the data meets the desired quality and format standards before loading.

**Data Transformation and Cleaning**

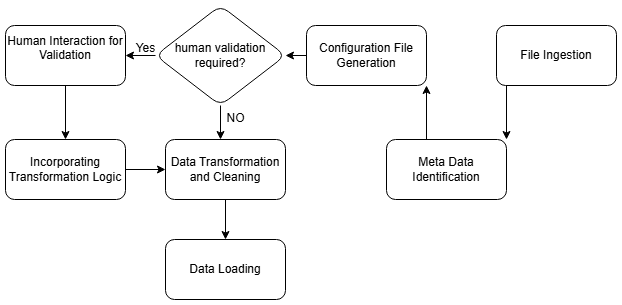
Using the instructions from the configuration file, the backend applies all specified transformations. These include renaming columns, handling missing values, normalizing data, and applying advanced transformations as defined by the user. This phase ensures that the dataset is prepared for seamless integration into the target database.

**Data Loading**

The final step involves loading the transformed data into a SQL Server database. The loading process uses optimized techniques to handle large datasets efficiently. Indexing and batch operations are employed to minimize load time and enhance performance. The system also logs the loading process, capturing details such as the number of rows loaded, errors encountered, and time taken.

**Automation and Scalability**

Although workflow orchestration tools like Apache Airflow are not implemented in the initial version, provisions are made to include automation for scheduling and monitoring tasks in future iterations. The current process can be run manually or triggered programmatically, ensuring flexibility in deployment.



*Fig. 3.1 Proposed Methodlogy for ETL*

CHAPTER 4: REFERENCE

[1] K. C. Mondal, N. Biswas, and S. Saha, "Role of Machine Learning in ETL Automation," in \*Proceedings of the 21st International Conference on Distributed Computing and Networking (ICDCN 2020)\*, Kolkata, India, Jan. 2020, pp. 1-6.

Available: [(PDF) Role of Machine Learning in ETL Automation](https://www.researchgate.net/publication/339368691_Role_of_Machine_Learning_in_ETL_Automation)

[2] W. Yaddow, "Considerations for Automating Data Warehousing and ETL Tests," presented at the Datagaps, Feb. 2020. Available:

[(PDF) Considerations for Automating Data Warehousing and ETL Tests](https://www.researchgate.net/publication/339384332_Considerations_for_Automating_Data_Warehousing_and_ETL_Tests)

[3] C. Van der Putten, "Transforming Data Flow: Generative AI in ETL Pipeline Automatization," M.S. thesis, Dept. Data Sci. and Eng., Politecnico di Torino, Turin, Italy, Apr. 2024. Available:

[tesi.pdf](https://webthesis.biblio.polito.it/30855/1/tesi.pdf)

[4] P. Pham, "A Case Study in Developing an Automated ETL Solution – Concept and Implementation," Bachelor's thesis, Dept. Inf. and Commun. Technol., Turku Univ. of Appl. Sci., Turku, Finland, 2020. Available: <https://www.theseus.fi/bitstream/handle/10024/340208/Pham_Phuong.pdf?sequence=2>

[5] M. T. Maulik, "Automated ML ETL Pipeline of Electric Motor Temperature Sensor Data for Commercial Vehicles," Master's thesis, Dept. Data Sci. and Eng., Politecnico di Torino, Turin, Italy, Apr. 2024. Available:

[https://github.com/maulikt04/Automated-ML-ETL-pipeline-of-electric-motor-temperature-sensor-data-for-commercial-vehicles-.](https://github.com/maulikt04/Automated-ML-ETL-pipeline-of-electric-motor-temperature-sensor-data-for-commercial-vehicles-.%20)

[6] Shopdev, "ETL Pipeline Using Snowflake," Shopdev, Dec. 16, 2022. [Online]. Available: [https://www.shopdev.co/blog/etl-pipeline-using-snowflake.](https://www.shopdev.co/blog/etl-pipeline-using-snowflake.%20)

[7] “Real-Time Data Warehousing.” Auckland University of Technology. [Online]. Available: <https://dsrc.aut.ac.nz/our-research/research-projects/real-time-data-warehousing.>

[8] L. Lucius, "What is ETL," presented at SlideServe, [Online].

Available[: https://www.slideserve.com/lucius/what-is-etl-powerpoint-ppt-presentation.](file:///C:\Users\pikesh.maharjan\AppData\Roaming\Microsoft\Word\:%20https:\www.slideserve.com\lucius\what-is-etl-powerpoint-ppt-presentation)

[9] "ETL," IBM. [Online].

Available: <https://www.ibm.com/think/topics/etl>.

[10] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, D. W. Lonsdale, Y. K. Ng, and R. D. Smith, "Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages," Data and Knowledge Engineering, vol. 31, no. 3, pp. 227-251, 1999.

Available: <https://pages.cs.wisc.edu/~smithr/pubs/dke99.pdf.>

[11] V. Radhakrishna, K. SravanKiran, and V. Ravikiran, "Automating ETL Process with Scripting Technology," in Nirma University International Conference on Engineering (NUiCONE), 2012

Available: <https://ieeexplore.ieee.org/document/6493198.>

[12] A. S. V. V. Akisetty, A. Kumar, M. M. K. Dandu, P. Goel, A. Jain, and A. Shrivastav, "Automating ETL Workflows with CI/CD Pipelines for Machine Learning Applications," in Nirma University International Conference on Engineering (NUiCONE), 2012, pp. 1-4. [Online]. Available: <https://www.irejournals.com/formatedpaper/1705069.pdf.>

[13] W. Qu, V. Basavaraj, S. Shankar, and S. Dessloch, “Real-Time Snapshot Maintenance with Incremental ETL Pipelines in Data Warehouses,” in Big Data Analytics and Knowledge Discovery, Springer, 2015

Available: [https://link.springer.com/chapter/10.1007/978-3-319-22729-0\_18.](https://link.springer.com/chapter/10.1007/978-3-319-22729-0_18.%20)

[14] V. Radhakrishna, K. SravanKiran, and V. Ravikiran, “Automating ETL process with scripting technology,” in Nirma University International Conference on Engineering (NUiCONE), IEEE, 2012, pp. 1–4.

Available: [https://ieeexplore.ieee.org/document/6493192. Last accessed: Nov. 26, 2024.](https://ieeexplore.ieee.org/document/6493192.%20Last%20accessed:%20Nov.%2026,%202024.)

[15] F. Sebastiani, “Machine learning in automated text categorization,” ACM Computing Surveys (CSUR), vol. 34, no. 1, pp. 1–47, 2002.

Available: <https://dl.acm.org/doi/10.1145/505282.505283.>

[16] D. Skoutas and A. Simitsis, “Designing ETL processes using semantic web technologies,” in Proceedings ACM 9th International Workshop on Data Warehousing and OLAP (DOLAP 2006), Arlington, Virginia, USA, 2006

Available: [https://dl.acm.org/doi/10.1145/1183346.1183358.](https://dl.acm.org/doi/10.1145/1183346.1183358.%20)

[17] S. Suresh, J. P. Gautam, G. Pancha, F. J. DeRose, and M. Sankaran, “Method and architecture for automated optimization of ETL throughput in data warehousing applications,” US Patent 6,208,990, 2001.

Available: [https://patents.google.com/patent/US6208990B1/en.](https://patents.google.com/patent/US6208990B1/en.%20)

[18] M. N. Tho and A. M. Tjoa, “Zero-latency data warehousing for heterogeneous data sources and continuous data streams,” in 5th International Conference on Information Integration and Web-based Applications Services, 2003, pp. 55–64.

Available: <https://ieeexplore.ieee.org/document/1241161>

[19] V. Tziovara, P. Vassiliadis, and A. Simitsis, “Deciding the physical implementation of ETL workflows,” in Proceedings of the ACM Tenth International Workshop on Data Warehousing and OLAP, ACM, 2007, pp. 49–56.

Available: <https://dl.acm.org/doi/10.1145/1317331.1317341.>

[20] P. Vassiliadis, “A Survey of Extract - Transform - Load Technology,” International Journal of Data Warehousing and Mining, vol. 5, no. 3, pp. 1–27, 2009.

Available:

<https://www.igi-global.com/article/survey-extract-transform-load-technology/37403>

[21] P. Vassiliadis and A. Simitsis, “Near Real Time ETL,” Springer Annals of Information Systems, vol. 3, no. 978-0-387-87430-2, 2008.

Available: [https://link.springer.com/chapter/10.1007/978-0-387-87431-9\_3.](https://link.springer.com/chapter/10.1007/978-0-387-87431-9_3.%20)

[22] P. Vassiliadis and A. Simitsis, “Extraction, transformation, and loading,” in Encyclopedia of Database Systems, Springer, 2009, pp. 1095–1101.

Available: [https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9\_110.](https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_110.%20)

[23] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, “On the Logical Modeling of ETL Processes,” in Proc. International Conference on Advanced Information Systems Engineering, 2002, pp. 782–786.

Available: [https://link.springer.com/chapter/10.1007/3-540-47919-1\_53.](https://link.springer.com/chapter/10.1007/3-540-47919-1_53.%20)

[24] H. Zhou, D. Yang, and Y. Xu, “An ETL strategy for real-time data warehouse,” in Practical Applications of Intelligent Systems, Springer, 2011, pp. 329–336.

Available: [https://link.springer.com/chapter/10.1007/978-3-642-25658-7\_40.](https://link.springer.com/chapter/10.1007/978-3-642-25658-7_40.%20)