Jeremi Junka

Literature Review of the ETL and ELT Data Integration Pipelines

School of Technology and Innovations
Bachelor's thesis in Technology
Data architecture

Vaasa 2026

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Jeremi Junka
Title of the Thesis: Literature Review of the ETL and ELT Data Integration Pipelines
Degree: Bachelor's thesis in Technology
Programme: Data architecture
Supervisor: Maarit Välisuo
Year: 2026 **Sivumäärä:** 37

ABSTRACT:
Tämä tutkielma on kirjallisuuskatsaus ETL (Extract-Transform-Load)- ja ELT (Extract-Load-Transform) -dataintegraatioprosesseista. Kirjallisuuskatsaus keskittyy prosesseihin ja niiden toimintaan ja siihen, miten niitä voidaan optimoida suorituskyvyn näkökulmasta. Sekä ETL- että ELT-prosesseja käytetään samaan tarkoitukseen, mutta prosessien toteutus eroaa merkittävästi toisistaan. ETL-prosessoinnissa muunnosprosessi suoritetaan ennen datan lataamista kohdejärjestelmään, kun taas ELT-prosessoinnissa muunnosprosessi suoritetaan datan kohdejärjestelmään lataamisen jälkeen. Tässä tutkielmassa keskitytään kolmeen päävaiheeseen, jotka esiintyvät molemmissa dataintegraatioputkimalleissa. Tässä kirjallisuuskatsauksessa tutkitaan näiden kolmen päävaiheiden optimointia. Tutkielmassa vertaillaan myös näiden kahden prosessointiprosessimallin soveltuvuutta eri arkkitehtuureihin.

KEYWORDS: Data, data processing

Contents

Figures

Abbreviations

ETL - Extract-Transform-Load

ELT - Extract-Load-Transform

1 Introduction

Data pipelines play a key role in modern data management. They can be utilized for varying purposes for example data collection or information delivery. ETL and ELT pipelines have been widely adopted into use as data integration needs have increased over time. These pipeline models have many similarities as both apply a phased structure to process data, but they also have differences such as the data transformation phase location. ETL pipelines are the more traditional model. In the ETL pipeline model data is transformed before loading into the target system. ETL pipelines offer strong data governance possibilities. ELT pipeline models handle data transformation after loading the data into the target system using its operational capabilities. ELT pipelines offer scalable cloud storage options for the storage and transformation of data. Both pipeline models are used in the collection, remodelling and storage of data. The pipeline models architecture and process differ but both are used to achieve the same goal.

1.1 Background and motivation

The motivation of this thesis is the limited amount of research in comparing the ETL and ELT data integration pipeline processes. The studies on ETL and ELT pipelines where the scope of study is only either one of the pipeline models are numerous, but comparative studies on both are rare to be especially from the performance standpoint. The increased use of data integration tools also presents a need for further study in this field.

1.2 Problem definition

This literature review attempts to study the similarities and differences of ETL and ELT pipelines from the standpoint of process, performance and optimization. With the prior research into this topic this literature review examines how ETL and ELT pipelines operate and evaluate how these pipelines can be optimized.

1.3 Research objectives and scope

The scope of this study is a review of ETL and ELT data integration pipelines, their architecture and processes based on academic studies and industry articles. The study mainly focusses on the processes within the pipelines and the optimization of these processes from the standpoint of enhancing performance. While no benchmarks are used in this study the analysis is based on findings within relevant documentation.

1.4 Structure of the literature review

The literature review starts by examining the main phases of both ETL and ELT data integration pipelines. After which each pipeline model is examined in a chapter individually. The individual examination chapters include a review of the structures of the pipeline, optimization of each phase of the structure and key performance indicators. This is followed by a comparison of the two pipeline models which include an assessment of similarities and differences of the pipeline models and an analysis of which pipeline models are suitable for certain cases. The final chapter is a summary of the findings of previous chapters.

2 Stages of data integration process

Both the ETL and ELT processes occur in phases. These phases are categorised as data extraction, data transformation and data load. Each of these phases are an integral part of the process and are necessary for the system to operate.

2.1 Data extraction phase

The data extraction phase is the process in which both ETL and ELT processes extract data from multiple or heterogeneous sources. In this phase of both the ETL process and The ELT process the system gathers data for further processing in the later stages of both processes. Nithis et al. (2024) states that in data integration pipelines the first stage is the extraction phase. This is applicable for both ETL and ELT data integration pipelines. This data collection is commonly done on a large scale as to have the necessary amount of data to process into valuable information. The article goes into detail on how different source systems such as databases and APIs along other data sources can be used in the extraction process a both structured and unstructured data can be extracted Shaker, Abdeltawab & El Bastawissy al. (2011) states that the extraction phase is the first faze conducted in an ETL pipeline while also emphasising its responsibility to extract the data.

The optimization of the extraction phase is similar in both ETL and ELT cases as the pipelines are collecting data for the same source. The key difference between the procedures becomes more prevalent in the later stages of the data integration process.

2.2 Data transformation phase

The data transformation phase in both ETL and ELT pipeline solutions handles the same responsibilities. These responsibilities are case based and can be tuned to fit specific business rules and interests. Seenivasan (2022) expresses that the transformation phase

is used for the conversion of data into a desired form usable in the target system. Examples of procedures that the transformation process is responsible for can include but are not limited to data cleansing, data standardisation, data enrichment, data validation, data restructuring, type conversion and data restructuring. Gill (2020) provides examples frequently used transformation techniques such as data cleansing, standardization and validation. These techniques are commonly used but other data transformation techniques can be used if necessary with both data integration pipeline models.

The transformation phase takes time in both ETL and ELT pipeline solutions as each additional procedure adds complexity to the system increasing processing time. In both ETL and ELT cases the transformation phase integral as it handles the tasks of manipulating the existing data into a necessary format for a specific use case. Nithish et al. (2024) states that data transformation is the process of reconstructing raw data into an optimized form that can later be used in varying ways.

2.3 Data load phase

The data load phase is the process in which data is loaded into the target system after other phases of the pipeline are completed. Gill (2020) expresses that the load phase is the transferring of data into the target system. In the article he states that these target systems may be data warehouses, data lakes or analytical platforms. Notable in this is how the target system can affect the pipeline architecture as the target systems computational capabilities and scalability possibilities affect what data integration pipelines should be used for the optimal results.

The load phase is important for both ETL and ELT solutions as it handles the data storage aspects of the data integration process. In essence, the load phase of the data integration process can be summarised as the storing of data into a target system for later use. Walha et al. (2024) explains the load phase in ETL pipeline as the storage of transformed

data in the target system. The article states that this stored data can be later used querying and reporting.

3 ETL

Behrend & Jörg (2010) explain the ETL pipeline as a process that first extracts data after which it transforms the data and finally the pipeline loads the data into the target system. ETL processes are used in the handling of data in many circumstances. While other methods of data cleansing exist ETL is a widely used as it has been adopted into many systems and in many cases, it is compatible with legacy code and data management. Dinesh & Devi (2024) emphasises the important role of ETL pipelines automating the data integration processes and improving efficiency for these tasks. In their article they state that clean standardized, clean structured data in important for analytics, reporting and decision making.

3.1 ETL data processing method

ETL processes work by gathering a large amount of data from varying sources. After the extraction of data, the ETL method transforms the available data into a desired form defined by differentiating needs and business rules. After the data is transformed into a suitable schema it is loaded into a data warehouse from which the data can be accessed and used for various purposes. Dinesh & Devi (2024) state that the automation of the ETL process is crucial in making the data integration process more efficient and reliable. ETL tools and platforms are used in the process of transforming data into a suitable format for later use.

3.1.1 Data extraction process for ETL

The data extraction process in ETL methods is comprised of gathering a large amount of relevant data from multiple sources. The data gathered is not usually in the final format. In this stage the data is gathered into a staging area where it is stored to prepare for the next process of transforming the data. Mandala (2019) states that the extraction process

is the collection of data from various sources. The data extracted can consist of both structured and unstructured data. Mandala in the article explains that the extraction process is caried out while the system load is minimal. This is done to mitigate the performance decrease caused by the ETL pipeline processing data. Mandala in the article also states that raw data is extracted in bulk from the source systems to retain the efficiency of the pipeline.

3.1.2 Data transformation process for ETL

The transformation process in ETL methods is used to change raw data into usable structured data that can be used for other means later. Mandala (2019) explains that the raw data in the staging area is cleansed and standardised to the specifications of a certain schema. Mandala states that the transformation process can be conducted of data duplication, format standardization and applying business rules to the data. Notable to this is that these and other transformations can be included in the transformation phase to get the desired structure. The transformation process takes a large amount of time and resources as all the gathered data needs to be transformed into the final wanted structure. In the event that the wanted data form is complex the performance of the ETL process may decrease as each individual transformation increases the complexity and resource usage of the transformation process. InfluxData (n.d.) explains that the more complex the transformations conducted by the pipeline the more impact they have on the performance of the pipeline. This is due to the fact that each transformation of the raw data requires a corresponding need for processing power. InfluxData describes the duties of the transformation phase as reformatting extracted data, improving data quality by changing or removing inaccurate and inconsistent data, cleaning datasets from corrupted, duplicated or otherwise incorrect data. These processes are used to change the extracted unstructured data into structured data. “The goal of transformation is to make all data fit within a uniform schema before it moves on to the last step.” (Informatica, n.d.) This is because only the transformed data is used by the load phase.

3.1.3 Data loading process for ETL

Mandala (2019) states that after the transformation process is completed the transformed data is then transferred into the target system. In the article it is stated that running the load process is scheduled for hour in which the system traffic is lower to prevent performance issues. Once the transformation process is completed for the entirety of the data available or just for a part of the data in total the transformed data can be downloaded into the target system or in many cases a data warehouse. The loading process is the final step in the ETL process and after it is complete the data within the data warehouse can be used for other operations. These operations can include for example analytics, reporting, machine learning and forecasting. For example, the processed data can be used in the creation of operational reports for a business.

3.1.4 Data transfer process within ETL processes

In ETL data integration pipelines the data flows from the source database to the staging area by the extraction process. The transformation process occurs in the staging area. The cleaned data is then loaded into the target system from which it can be used for varying purposes.

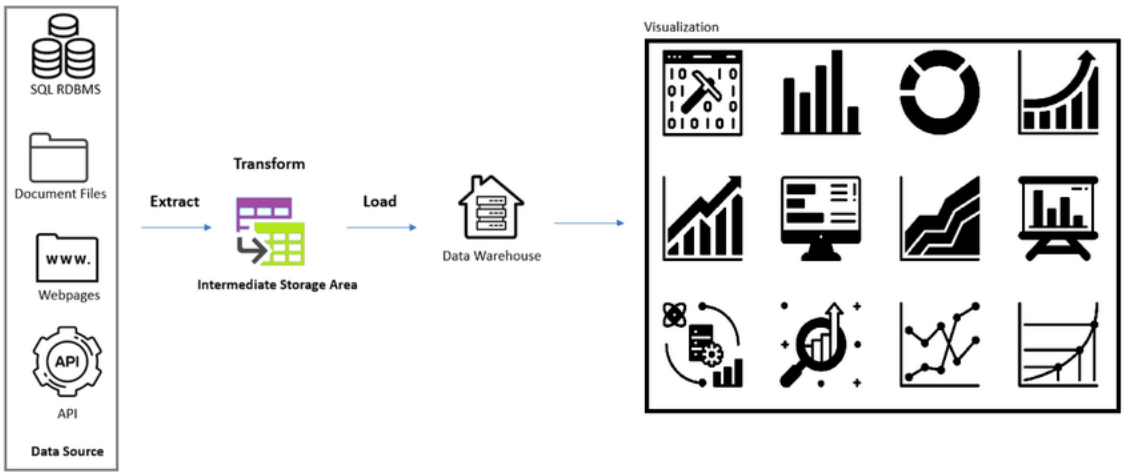

Figure 1 “A diagram depicting an ETL pipeline for the integration of dashboard data.” (Myakala, Bura & Juma, 2024).

3.2 Key performance indicators

Throughput and latency are complementary metrics as they can be used in the measuring of how much data an ETL process handles and in what amount of time. Theodorou & Ayomide (2025) lists of ETL key performance indicators and one of them is latency. Latency is the delay in between user input and the response of the system. In their article they also explain that another key performance indicator is throughput. They explain that throughput is the amount of data handled by ETL operations. Arsan & Amagowni (2022) states that the performance of a pipeline can be quantified by measuring the throughput of the pipeline.

3.2.1 Throughput

Throughput is a key metric in understanding the amount of data handled by ETL processes. Rongala (2025) states that ETL pipeline used for financial applications use throughput as a way to measure the amount of data handled by the pipeline in a certain amount of time. Throughput can be measured directly as the amount of data that has been processed with ETL systems. Akkaoui, Vaisman & Zim´anyi (2019) states that throughput is a key performance indicator in the efficiency aspect of ETL pipelines.

3.2.2 Latency

Latency is a key metric for how well a ETL process operates and how efficiently it can handle data. There are different types of latency with ETL processes. Adetnji (2025) defines latency into different types end to end latency, processing latency and ingestion latency. In the article end to end latency is defined as a time taken from when the data

first arrives in the pipeline to the time it is acted upon. The article describes processing latency as the time taken for the extraction, transformation and loading of an output. The article describes ingestion latency as the delay occurring before the pipeline receives an event. These three types of latency within ETL processes can be used as metric to define what parts of the data pipeline operate at a sufficient level and what still needs to be improved. The understanding of latency within a system is critical in the assessment of the performance of a system. The number of errors also affect the performance of a system and latency needs to be balanced with accuracy to achieve usable results from an ETL process. Rongala (2025) describes latency as the time taken for the whole pipeline to process and store the data. This difference shows that latency can be measured in different ways in data integration pipelines.

3.3 Optimization of ETL models

ETL tool optimization can be considered to be done in three stages which are extracting data, the processing of data into a suitable form and loading the data into a storage system such as a data warehouse. Each of these steps should be optimized for the best possible system for a use when considering performance. The optimization of ETL tools is a large contributor of how well the ETL data pipeline operates. In many cases the most basic design can affect how much time is spent for the operations to run through. This may take a considerably long time when large data quantities are being processed. To decrease this wait time and getting the transformed data faster batch ETL is a good approach to consider. Batch ETL processes large data quantities in batches and thus increases transformed data availability. Walha, Ghozzi and Gargouri (2024) imply that data becomes available only after the ETL process is completed and the result of the process is a unified dataset. Other ETL optimization techniques are real time ETL and micro batch ETL. These techniques are used to balance throughput and latency for systems. The choice between what technique is dependent on the needs of connected systems and the user.

3.3.1 Data extraction optimization

A way of optimizing the extraction process of data for ETL is applying incremental loading techniques. Incremental extraction techniques work on the premise that loading only the data that has changed is faster and more resource efficient than extracting a total duplicate of the data already stored in the system. Oracle (n.d.) describes incremental extraction as only extracting data that has change in comparison to a known point in time. An example of incremental extraction is the leveraging of the timestamp of data. During the first run of the ETL pipeline all timestamps are the same and all data is extracted. On later runs of the pipeline incremental extraction compares the timestamps of the data to be extracted in comparison to the timestamps previously extracted data. Incremental extraction only extracts newer data from the source system. This process decreases the amount of data extracted and thus allowing for faster processing of the extraction phase.

Partitioning data is also a valuable optimization technique. It divides data into partitions based for example on time, business keys or source. This technique achieves efficiency optimization by discarding unnecessary partitions and thus reducing the systems overhead. “The framework first partitions an ETL dataflow into multiple execution trees according to the characteristics of ETL constructs, then within an execution tree pipelined parallelism and shared cache are used to optimize the partitioned dataflow.” (Liu & Iftikhar, 2015).

3.3.2 Data transformation optimization

Data transformation optimization is a key part of how well an ETL model operates. This is due to the fact that each data transformation adds complexity to an ELT algorithm. The added complexity affects the amount of time taken and resources used to complete the processing of data with the ETL software. Oracle (n.d.) states that the transformations for the data are in many cases the most time consuming and complex part of ETL pipeline.

ETL models handle the data transformation processes before the data is loaded into the target system. This increases the latency of the system as a whole and increases processing times.

ETL data integration pipeline models use parallelization of transformation tasks to optimize the transformation process. In parallelization the transformation tasks are divided among multiple threads to accomplish optimization of the model in the performance aspect. Incremental and partitioning techniques can also be used in the transformation phase of the ETL process. Other techniques can be used to optimize the transformation phase as well. Kumari (2017) gives examples of ETL performance optimization techniques such as pushdown optimization, partitioning and parallelization among other techniques.

3.3.3 Data loading optimization

There are various ways to optimize the load process in ETL pipelines. One of these optimization possibilities is to use multithreaded or parallel loading with optimally selected batch size load processes. Oracle (2009) states that parallel execution processes are a viable optimization technique for optimizing an ETL pipeline.

Another way of optimizing the load procedure is to use shared cache memory along with the ability to use multiple cores for the procedure. The main idea behind this optimization route is to reduce the amount of memory overhead along with reducing the amount of redundant copies. Through this method the load process as a whole can be optimized is the performance aspect. Masouleh, Afshar, Alborzi & Toloie (2016) states that an ETL pipeline performance can be optimized considerably by using shared cache memory and parallel processing. These and other methods can be used in conjunction or separately for the optimization of the pipeline.

3.4 Relevance of ETL tools in data architecture

ETL processes are a key piece of data architecture. This is especially clear when considering the aspects of big data and data warehousing. These large projects regularly consist of cleansing and optimizing data that needs to be used later on. Dhaouadi et al. (2022) states that 80% of the time spent on a data warehouse project is used for extracting cleaning and loading data when the time taken on inherent issued of Big Data is not considered.

4 ELT

ELT is a data pipeline process where data is first extracted from a number of sources. After the extraction process the raw data is loaded into a target system. Examples of target systems can be data warehouses or data lakes. Ballard et al. (2011) states that the ELT process first extracts the data after which it is loaded into a data warehousing environment. After the data is loaded into a target system the raw data is transformed into a suitable form. Google Cloud (n.d.) states that the target system of the ELT pipeline can be a data lake or a cloud data warehouse. The transformation process of the data is completed using the target systems operational power. In the article Google Cloud stated emphasises that ELT pipelines leverage the computational resources of the target system to conduct the transformations on the raw data.

4.1 ELT data processing method

The ELT data processing method consists of three primary processes. These processes are the data extraction process, the data loading process and the data transformation process.

The loading process is the second phase of the ELT process. With this phase new data is loaded into a target system. “In the second step, the extracted raw data is loaded, often in its original format or with minimal processing, directly into a high-capacity storage system.” (Google Cloud, n.d.) The final phase of the ELT pipeline is the transformation processes performed onto the preloaded data within the target system.

4.1.1 Data extraction process for ELT

The data extraction process in the ELT pipeline is the first phase of the process. In this step the main objective is to retrieve useful data from source systems and files which in many cases are not structured. Spooner (2011) describes the extraction process as the

accessing of a source system from which only the data used in the ETL pipeline is extracted. This is useful as it filters out unnecessary data. This phase handles data gathering for future phases where the raw data is stored and processed.

4.1.2 Data load process for ELT

The data load process is the phase of the ELT pipeline in which the previously extracted data is loaded into a target system. This phase occurs directly after the raw data is obtained in the extraction phase. Seenivasan (2022) states that the load process transfers the raw data into a target system where the raw data can be stored. These target systems may vary but a common feature of the target storage systems is that they have innately a large amount of processing power or have access directly to a large amount of processing power. Various cloud storage systems, for example, can be used for these purposes. Seenivasan (2022) also states that the raw data can also be copied into a staging area or directly into the target system. This can allow for more flexibility when designing ELT pipelines. Overall, the load process is responsible for the transfer of data into a storage device and the storing of the data itself.

4.1.3 Data transformation process for ELT

In this stage of the ELT data integration pipeline the data that has been obtained and stored in previous phases is then transformed into a suitable form. In a Google cloud (n.d.) article the transformation process is described as the final step of the ELT pipeline. The article also states that in ELT data integration pipelines the transformation process occurs within the target system. The transformation process is mainly responsible for the cleansing, standardization, normalization, enrichment, modeling, joining and validation. In a Google Cloud (n.d.) article the transformation process is described to be used for the purposes of cleaning, structuring, enriching raw data while converting it into a desired format. The article states that this process can be used for analytics, reporting or

for machine learning needs. The article also describes possible transformations conducted during the transformation process these include for example filtering data, joining data, data aggregation, data format standardization and the derivation of new data points. These methods can be applied to the raw data obtained in the extraction phase of the data integration pipeline. The transformations mentioned can be used to manipulate varying types of raw data into structured data usable in the future. The proper use of each method of transforming the raw data is key in obtaining usable material for future use.

The data that is to be transformed is already within a suitable target system that can handle the computation for the data transformation process. Seenivasan (2022) describes the transformation process as the level where the transformations are done. While he in his article he explains the data transformation happens within the target system as the target system of and ELT pipeline owns the necessary computation capability to handle intense transformations of the data. This is exemplified with data stored within cloud-based systems as they commonly have considerable computational resources allowing ELT pipelines to operate smoothly within them.

The main goal of the transformation process is to change raw data into clean and usable data for future processing. This is an important step as the raw data does not fit all system requirements and business rules.

4.1.4 Data transfer process within ELT processes

The ELT data integration pipeline processes data in the following order. First data is extracted from varying sources. The extracted data is then loaded directly into the target system. This target system uses its processing power to transform the data into a cleansed form which it can be used for varying analytics.

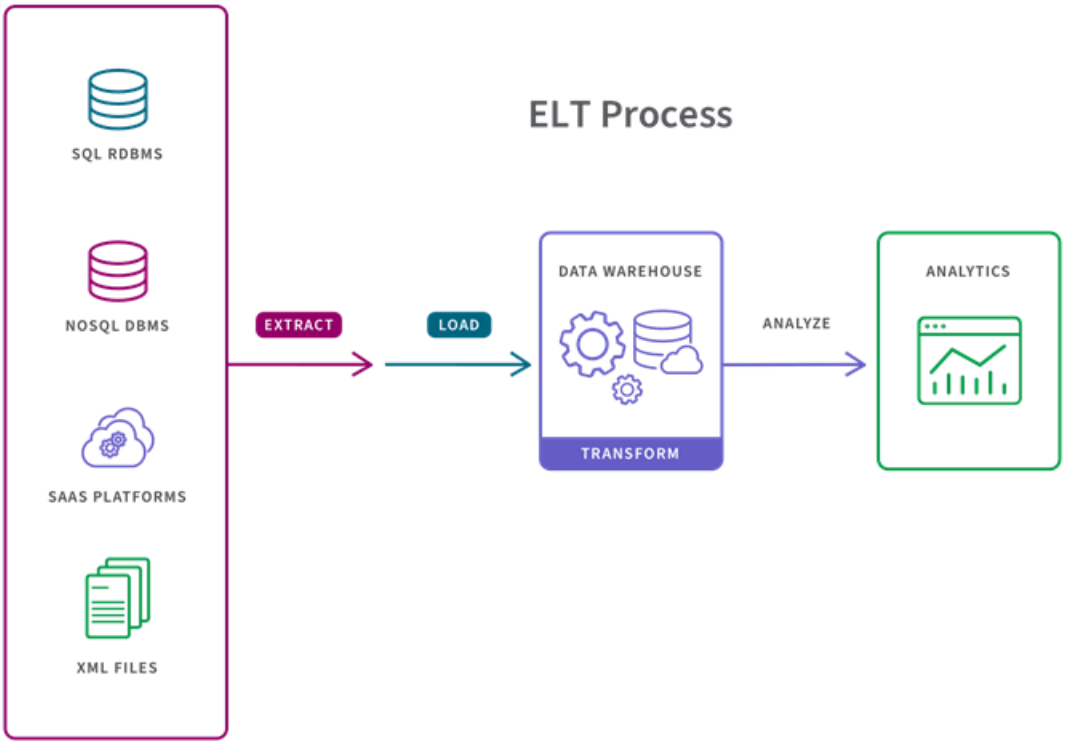

Figure 2 "ELT Process (Qlik, n.d.)".

4.2 Key performance indicators

The key performance indicators that can be used to determine the effective usage of ELT systems are varied but the main indicators considered in this review are throughput and latency. These indicators have been chosen to focus on because they can be used to determine the efficiency of ELT data integration pipelines. Allwell et al. (2025) describes that latency is the delay of time from the start of the process to the throughput of the data. In this article the throughput of data is described as the volume of data processed by a system in a certain amount of time. Both throughput and latency can be inferred as key performance indicators for ELT data integration pipelines as the they are designed to handle the largest amount data in the shortest amount of time.

4.2.1 Throughput

Throughput measures the amount of data processed in a certain amount of time. This metric is important in understanding the efficiency of ELT data pipeline solutions as it gives clear information on how quickly the pipeline can handle a certain amount of data. Allwell et al. (2025) in their article consider data throughput as the data volume a pipeline can process in a given time frame. Throughput is a key performance indicator in all types of ETL pipelines, but real-time pipeline models emphasize its role in optimization of the pipeline. Throughput is measured for the whole processing of the pipeline including the extraction, load and transformation phases. A large factor in how large the final throughput for an ELT pipeline solution is the complexity of the transformation phase. This is due to each additional transformation increasing the pipelines complexity. With the increase of complexity, the processing time increases affecting the throughput if proper optimization techniques such as parallel processing are not used. Owen 2024 states that if a pipeline does not use optimization techniques the pipeline's overall performance decreases and that the performance in such cases is affected noticeably by the complexity of the transformation phase. A higher throughput for a functional pipeline means the efficiency is better in comparison than a pipeline with a lower throughput while both pipelines function identically in all other aspects.

4.2.2 Latency

Latency is the end to end measurement of how long it takes for the pipeline to operate fully and the processed data to be available for use. In the case of ETL pipelines this means that latency is the entire amount of time taken by the extraction phase, load phase and the transformation phase overall.

Latency can be measured for each individual phase of the pipeline or as a whole for example the time latency for the set transformations to operate may be named as transformation latency. Another form of latency is ingestion latency. Dupe et al. (2025) states

that ingestion latency is the time delay between the beginning of the whole process to the first event of a pipeline. Latency is a key metric in determining the efficiency of the data integration pipeline as the time taken for processing data may be long. With knowing the latency of a data integration pipeline, the appropriate solution may be taken on a case-by-case basis. Latency is largely affected by the way a data integration pipeline is optimized. By optimizing a data pipeline, the latency of the system can be changed to fit different needs.

4.3 Optimization of ELT models

The optimization of an ELT data integration pipeline model can be divided into three different categories based on the phase of the pipeline. A pipeline should be optimized as a whole as well as the parts of the pipeline individually. Each optimization needs to be planned beforehand to maximize the benefits the optimization can provide. The optimization of a pipeline can affect the performance of a model considerably. Zvonarev et al. (2023) in an article state that while that the amount of optimization methods proposed for ETL pipelines is lower than optimization methods proposed for ETL pipelines many of the optimization models proposed can be used in both data integration pipeline models.

4.3.1 Data extraction process optimization

The data extraction phase of the data integration pipeline should be optimized as the extracted data is key in defining the possible uses of the data and scalability based of the systems using the final data. Optimization of the extraction process can also impact the performance of the pipeline.

The extraction process can be optimized in many ways. Examples of the optimization related techniques are change data capture, parallel extraction, batch sizing,

compressing data for the transfer, incremental extraction and extracting only relevant data. Change data capture operates on the idea to not extract data that the system already has. Batch sizing on the other hand can be used to tune the model for optimal through-put by changing the size of each batch being processed.

4.3.2 Data load process optimization

The data load phase of the ELT data integration pipeline should be optimized as it is responsible for the writing of data into the target system of the pipeline. By optimizing this process the system can handle larger amounts of data in less time.

The optimization of the load process can be done in many ways and the chosen optimization technique needs to be chosen on a case-by-case basis. The load process for ELT pipeline solutions can be done by partitioning the load process, partitioning the extracted data, compacting small files into larger files and using clustering methods. Richman (2024) states that by applying incremental load strategies in data storage systems the time taken by the system to load the data can be decreased and use of computational resources can be lessened.

4.3.3 Data transformation process optimization

Zvonarev et al. (2023) state that optimization of the transformation process is a key part in the optimization of the data integration pipeline. I their article they state that executing the transformation processes directly within the target system the transformation phase as a whole can be significantly optimized. In their article Zvonarev et al. also provide other optimization techniques such as batch processing, incremental transformations and parallel processing among other techniques optimize the transformation process. An example of transformation optimization techniques is data layout optimization. Data layout optimization is a technique where the process of data storage is

optimized meaning data partitioning, clustering and key distribution. With these methods data layout optimization can make the transformation process more efficient. Oracle (2024) states that throughput can be increased by per- forming the SQL operations in batches. With the use of these techniques the transformation process can be optimized for various circumstances including the optimization of the efficiency of the data pipeline.

4.4 Relevance of ELT tools in data architecture

ELT data integration pipelines are highly relevant in data architecture and the overall design of a functional system as they handle a large amount of data gathering, storing and normalization operations a system handles. ELT data integration pipeline solutions are becoming more prevalent with the adaptation of new technology. Garcia (2023) notes in his article that cloud-based data storage systems such as Snowflake or BigQuery can be easily scaled to meet the requirements of a system and that they can be optimized to process data within the storage system itself. This can improve the throughput and latency of a pipeline if designed properly. ELT solutions use the computation power of the storage systems. This increases the performance of the pipeline models while also simplifying the architecture of the data integration pipeline. When using an ELT data integration pipeline the gathered data is both centralized and democratized as raw and trans-formed data are included in the same storage system.

5 Comparison between ETL and ELT tools

This chapter aims to provide a clearer understanding of the ETL and ELT data integration pipeline models through comparison. This chapter aims to analyse what similarities and differences the ETL and ELT pipeline models include.

5.1 Overall process comparison

The goal itself of both ETL and ELT tools is to transfer and transform data from source systems to target systems and to transform the data into a desired form. With automation the ETL and ELT processes data integration can be accelerated to meet needs of data analytics. The main difference of these tools is how the process is handled. While the extraction of data with both models is similar the transforming and loading components vary considerably.

5.2 Process comparison between ETL and ELT tools

The largest difference between ETL and ELT data integration pipelines is the operation pattern the each of them uses. In these two pipeline models the transformation phase occurs at different points. In ETL pipeline models the transformation of the data occurs immediately after the extraction phase in a separate staging area, after which the transformed data is loaded into the target system. On the other hand, in the ELT data pipeline model the transformation phase occurs after the raw data is loaded into the target system. The transformation of the raw data occurs directly within the target system leveraging the systems inherent processing power. The pipeline models differ in their structure due to the inherent differences in their target systems, business rules, approaches on transforming data and data volume. The inherent difference in target system forces an ETL pipeline to complete the transformation phase before the data is loaded into the target system as the target system does not have the processing power to complete the

necessary transformations. In comparison an ELT pipeline with a target system that has considerable processing power can handle the needed transformations. Another example is the difference in data quality control ETL pipelines transform.

5.2.1 Extraction process review

From an operational aspect the data extraction process is used for the data collection in both pipeline models. Data is collected from varying sources to be used at a later stage in the pipeline. The extraction process is completed in both pipeline models as the first stage of the data integration pipelines. Google Cloud (n.d.) article states that the extraction process only gathers the data without using it for other purposes. The extraction process philosophy is slightly different within the data pipelines. While both pipeline models can handle differing quality of data the ETL model more easily operates if the extracted data is cleaner beforehand in comparison to ELT data integration pipeline models. This also affects the amount of data each pipeline model can accommodate with ease as cleaner and more optimized data takes less memory to store. Overall, the data extraction process in both data integration pipeline models is used for the same purpose of data gathering. The extraction process is completed in the same way in both pipeline models and in both models is the first stage used in preparation so the following stages can be completed.

The optimization of ETL and ELT data integration pipelines for the aspect of efficiency is similar as many of the techniques work for both models. One of the key optimization techniques that can be used in the optimization process in both techniques is CDC or incremental loading. CDC or change data capture means that only the changes in the sourced data are extracted instead of all source data each time. Agarwal (2025) explain the incremental data load as a method in which only data that has changed after the latest process is complete is selected and used. This can be leveraged to gain efficiency for during the pipeline as the same data does not need to be reworked allowing to better use a systems resources.

5.2.2 Transformation process review

The transformation process is the core of ETL and ELT data integration pipeline processes as it is directly responsible for transforming raw data into a clean, consistent and usable form. For both ETL and ELT data integration pipeline models the task the transformation phase is used for is the same. The key difference between the transformation phase in the two pipeline models is the location where the transformation phase occurs and at what point in the pipelines the transformation phase is performed. In the ETL pipeline model the transformation phase is performed in a separate staging area before the transformed data is loaded into the target system. On the other hand, for the ELT pipeline model the transformation phase is performed in the target system itself after the load process. The memory usage efficiency is notable as the staging area in ETL data pipeline model can bottleneck the efficiency of the transformation process. This is affected by the usage rate and overall amount of staging area memory. This is not so much a concern for the ELT data pipeline model as the transformation phase occurs directly in the target system. This can allow the ELT model to dismiss concerns of the staging area, but other bottlenecks such as the amount of processing power still affects the transformation phase. Qlik (n.d.) states that an ELT system works on an as-needed basis. ELT pipelines integrate data only when queried. This can cause delays for the future needs of the system.

The optimization of the transformation process is different between the two data integration pipeline models. ETL data integration pipelines achieve transformation process optimization through process level optimizations. Such as parallelization of transformation processes, incremental transformation processes and partitioning are all used to optimize the ETL pipelines transformation process. The ELT data integration pipeline transformation process is optimized with pushdown optimization, using inbuilt tools in the storage system and batch processing among other techniques. The ELT pipeline uses in-database optimizations such as query optimization, index optimization and in-database processing to improve the efficiency of the transformation process. These

optimizations include Garcia (2023) states that a data integration pipeline can use the computational capabilities in the pursuit of lower cost and increased performance. Notable in this is the need for data governance. The data security of systems that operate on the principle of in target system data transformation the security aspects may be lacking for certain purposes. As data arrives in the target system without prior transformation sensitive data such as credentials may be transferred in an unencrypted or unmasked state. This can create data security concerns as security controls are used only after the data is in the target system.

5.2.3 Load process review

The load process is the phase in which both ETL and ELT data integration pipeline models load data to be stored into the target system. In ETL pipeline models the loaded data is cleansed while in the ELT pipeline the data is raw directly extracted from the source systems. Thus, in many cases the load process complexity is higher with ELT pipeline models. Snowflake (n.d.) compares the load process in ETL pipeline models to be more agile than in ETL pipeline model. In the article Snowflake emphasizes the advantages of also loading raw data directly into the target system. These advantages include the ability to ingest data in a raw form, decreasing the need to transfer data between multiple pipelines and enabling the pipeline to handle large data volumes. This difference can allow for more flexibility in later usage. With ELT pipeline models the memory usage is heavier as both raw and transformed data can be stored in comparison to ETL pipeline in which only transformed data is commonly stored. The key similarity of ETL and ELT pipeline models in the load process is the purpose for which the load process is used. The key differences are the order in which the phase is executed, load complexity and load target system.

The load process can be optimized in varying ways for both ETL and ELT data integration pipeline models. These optimizations strategies include but are not limited to using incremental loads, partitioning, clustering, parallel processing and other optimization techniques. Agarwal (2025) notes the possible optimization of data integration pipelines

with the incremental load strategy to be capable of improving the load process performance. The layout of the data can also be used in optimizing the load process. Overall, the optimization of the load process can aid in the optimization of the entire pipeline.

5.3 Suitability for different environments and use cases

ETL and ELT differ in the aspect of what environments each model is optimal for. The use cases of these data integration pipeline models also are different. It must be noted that there is considerable overlap in what these systems can be adapted to handle. The differences between the ETL and ELT data integration pipeline models are most notable when an on-premises system needs to be implemented. The more optimal approach when an on-premises data integration pipeline needs to be designed is an ETL pipeline. This is be-cause ETL pipeline models can handle the data in a specific system without using cloud storage or other offsite data storage systems. Qlik (n.d.) states that sensitive data can be left out before the transformation process in ETL pipeline models. This can be very useful when considering the systems compliance with local laws. In the article Qlik provides examples of standards where the ability to leave out sensitive data is necessary. These standards can include GDPR or HIPAA for example. The ETL pipeline model can also operate complex transformation logic without having to use SQL or other transformation logic engines.

ELT data integration pipeline models can handle a large quantity of data with the use of computational power of the data storage system in use. This is because ELT systems can leverage the cloud storage systems where the data is stored to operate the transformation logic. Google Cloud (n.d.) explains how data lakes can be used as a viable target system as they are designed to handle large amounts of data while also having the computational capability to handle ELT pipelines. Use cases in which scalability need due to a large quantity of data is also optimal for ELT data integration pipeline models as the scalability needs can more easily be accommodated with the aid of cloud data storage.

6 Conclusions

This study gives a comprehensive review of ETL and ELT data integration pipelines. It is a literature review that uses as sources both industry articles and academic articles. The study begins with an introduction to what ETL and ELT data integration pipelines are followed by further study into how they process data and how the pipelines can be optimized in the efficiency aspects. After this the processes are compared on what similarities they have and what differences they include.

Both ETL and ELT pipelines include three main phases. These three phases are the extraction process, the transformation process and the load process. The extraction process is similar in both processes as raw data is extracted from varying source systems. The transformation process handles the same task in both pipeline models, but the tools used in this process are different between the pipeline models. The load process is used to store data into the target system. With ETL pipelines this is the final phase of the pipeline while with ELT pipelines the load process precedes the transformation process. Differences in the phase structures are the clearest distinctions between the two pipeline models.

The differences in ETL and ELT data integration pipelines are the result of deviating design in hardware, software and the optimization of the models. The hardware used by ETL pipelines as the target systems can for example be a data warehouse with limited processing power. The hardware used by ELT pipelines is a cloud storage system that includes notable processing power. This inherent difference in capability of transforming loaded data forces ETL pipelines to transform the raw data outside the target system in a staging area before loading the curated data into the target system. An ELT pipeline can use the processing resources of the target system to transform raw data. These differences caused by the hardware also affect the software as ETL pipelines are designed in a way that allows for the transformation of a large amount of data while having limited processing power in the target system. The software of ELT pipelines is designed to take advantage of the processing power of the target system allowing large amounts of raw

data to be loaded directly into the target system. The differences in the available optimization techniques come from the differences between pipeline architecture and target systems themselves. Some of these optimization techniques can be used in one or both pipeline models. Incremental extraction can be used in both pipelines to decrease memory usage as with incremental extraction only changed and new data that the pipeline has not already processed is extracted from the source system. SQL batch execution is used in ELT pipelines to optimize the transformation phase by grouping and executing multiple transformations at once. With this technique the pipelines throughput can be increased with better CPU utilization. The load phase in ETL pipelines can be optimized with optimized batch size loading. Optimized batch size loading combines data into batches of rows that are loaded into the target system as one operation. This can maximize throughput by decreasing loading overhead. The usage of optimization techniques can improve the performance of both ETL and ELT data integration pipelines considerably.

References

Agarwal Sanchit, (2025). ETL Incremental Loading 101: A Comprehensive Guide, Hevodata, https://hevodata.com/learn/etl-incremental/

Arsan Roy & Amagowni Sudesh, (2022). Benchmarking your Dataflow jobs for performance, cost and capacity planning, Google Cloud, https://cloud.google.com/blog/products/data-analytics/benchmarking-dataflow-jobs Retrieved 7.11.2025

Ayomide Joel, (2025). Optimizing data latency and throughput in etl processes through reinforcement learning, ResearchGate, https://www.researchgate.net/publication/398931364_OPTIMIZING_DATA_LATENCY_AND_THROUGHPUT_IN_ETL_PROCESSES_THROUGH_REINFORCEMENT_LEARNING

Ballard Chuck, Gomes Veronica, Hilz Gregory, Panthagani Manjula & Samuelsen Claus, (2011). Data Warehousing with the Informix Dynamic Server, www.redbooks.ibm.com/redbooks/pdfs/sg247788.pdf

Behrend Andreas & Jörg Thomas, (2010). Optimized incremental ETL jobs for maintaining data warehouses, Conference proceeding https://doi.org/10.1145/1866480.1866511

Dhaouadi, A., Bousselmi, K., Gammoudi, M. M., Monnet, S., & Hammoudi, S. (2022). Data Warehousing Process Modeling from Classical Approaches to New Trends: Main Features and Comparisons. *Data*, *7*(8), 113. https://doi.org/10.3390/data7080113

Dinesh, L., Devi, K.G. An efficient hybrid optimization of ETL process in data warehouse of cloud architecture. *J Cloud Comp* 13, 12 (2024). https://doi.org/10.1186/s13677-023-00571-y

Dupe Adetunji, Dorcas, Sheng Bou & Sulaimon Tunde, (2025). Latency vs. Accuracy: Balancing Performance with Data Integrity in Low-Latency Streaming ETL Workflows, https://www.researchgate.net/publication/392876455_Latency_vs_Accuracy_Balancing_Performance_with_Data_Integrity_in_Low-Latency_Streaming_ETL_Workflows

El Akkaoui Zineb, Vaisman Alejandro & Zim´anyi Esteban, (2019). A Quality-based ETL Design Evaluation Framework, In Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019), pages 249-257, https://doi.org/10.5220/0007786502490257

Shaker El-Sappagh H. Ali, Abdeltawab Ahmed M. Hendawi & El Bastawissy Ali Hamed, (2011). “A proposed model for data warehouse ETL processes” Journal of King Saud University -

Computer and Information Sciences, Volume 23, Issue 2, July 2011, Pages 91-104 https://doi.org/10.1016/j.jksuci.2011.05.005

Faridi Masouleh M., Afshar Kazemi M. A., Alborzi M. & Toloie Eshlaghy A., (2016). Optimization of ETL Process in Data Warehouse Through a Combination of Parallelization and Shared Cache Memory, Engineering, Technology & Applied Science Research, https://etasr.com/index.php/ETASR/article/view/849

Garcia Miguel, (2023). "The Evolution of Data Pipelines: ETL, ELT, and the Rise of Reverse ETL", https://dzone.com/articles/the-evolution-of-data-pipelines Retrieved 7.11.2025

Gill Sukhdeep, (2020). "The First Step to Business Intelligence: Ensuring Data Quality Through Rigorous ETL Processes" International Journal of Trend in Research and Development, Volume 7(4), 2020, http://www.ijtrd.com/papers/IJTRD28919.pdf

Google Cloud, (n.d.). What is ELT (extract, load, and transform)?, Google Cloud, https://cloud.google.com/discover/what-is-elt?

InfluxData, (n.d.). ELT (Extract, Transform, Load), InfluxData glossary, https://www.influxdata.com/glossary/etl/ Retrieved 7.11.2025

Informatica, (n.d.). What is ETL (extract transform load)?, Informatica, Retrieved 24.9.2025, from https://www.informatica.com/resources/articles/what-is-etl.html Retrieved 7.11.2025

Kumari Deepika, (2017). Performance Optimization of ETL Process, ResearchGate, https://doi.org/10.13140/RG.2.2.13994.44480

Liu, X., & Iftikhar, N. (2015). An ETL Optimization Framework Using Partitioning and Parallelization. In *Proceedings of the 30th ACM Symposium on Applied Computing (SAC 2015)* Association for Computing Machinery. https://doi.org/10.1145/2695664.2695846

Mandala Nishanth Reddy, (2019). The evolution of ETL architecture: From traditional data warehousing to real-time data integration, World Journal of Advanced Research and Reviews, 2019, 01(03), 073–084, https://doi.org/10.30574/wjarr.2019.1.3.0033

Myakala Praveen Kumar, Bura Chiranjeevi & Juma Russell, (2024). Interactive Data Dashboards: Design Principles, Best Practices, and Applications , ResearchGate, https://doi.org/10.13140/RG.2.2.14205.06882

Nithish, Ravi & David, (2024). "Data Transformation Techniques in ETL" *International Journal of Multidisciplinary on Science and Management*, Vol. 1, No. 2, pp. 01-16, 2024. Retrieved from https://www.ijmsm.org/ijmsm-v1i2p101.html (11.2025)

Oracle, (n.d.). Introduction to Extraction Methods in Data Warehouses, https://docs.oracle.com/cd/B28359_01/server.111/b28313/extract.htm

Oracle, (2009). Parallel Capabilities of Oracle Data Pump, An Oracle White Paper, https://www.oracle.com/technetwork/database/datapump11g2009-parallel-1-132209.pdf Retrieved 7.11.2025

Oracle, (2024). "Oracle® Fusion Middleware Using Oracle GoldenGate for Big Data", Oracle Fusion Middleware Using Oracle GoldenGate for Big Data, Release 19c (19.1.0.0) https://docs.oracle.com/en/middleware/goldengate/big-data/19.1/gadbd/using-oracle-goldengate-big-data.pdf retrieved 7.11.2025

Owen Benjamin, (2024). Optimization of ETL/ELT Pipelines in High-Volume Data Platforms, ResearchGate, https://www.researchgate.net/publication/398467239_Optimization_of_ETLELT_Pipelines_in_High-Volume_Data_Platforms

Qlik, (n.d.). "ETL vs ELT" retrieved 7.11.2025 https://www.qlik.com/us/etl/etl-vs-elt?

Qlik, (n.d.). What is ELT?, Qlik, https://www.qlik.com/us/elt retrieved 13.11.2025

Raj A., Bosch J., Olsson H., et al., (2020). Modelling Data Pipelines. Proceedings -46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020:13-20. http://doi.org/10.1109/SEAA51224.2020.00014

Richman Jeffrey, (2024). "How to Load Data into a Data Warehouse: Methods & Challenges", https://estuary.dev/blog/how-to-load-data-into-data-warehouse/ Retrieved 7.11.2025

Rongala Samyukta, (2025). Optimizing ETL Processes for High-Volume Data Warehousing in Financial Applications, Journal of Information Systems Engineering and Management, 2025, 10(8s), e-ISSN: 2468-4376, https://doi.org/10.52783/jisem.v10i8s.1130

Seenivasan Dhamotharan, (2022). "ETL vs ELT: Choosing the right approach for your data warehouse", International Journal for Research Trends and Innovation(IJRTI),Vol.7, Issue 2, page no.110 – 122

Snowflake, (n.d.). What Is ELT (Extract, Load, Transform)?, Snowflake site, http://www.snowflake.com/en/fundamentals/understanding-extract-load-transform-elt

Spooner John, (2011). Creating a SAS® Model Factory Using In-Database Analytics, SAS Global Forum 2011, Data Mining and Text Analytics https://support.sas.com/resources/papers/proceedings11/147-2011.pdf

Theodorou Vasileios, Abelló Alberto, Lehner Wolfgang, Thiele Maik, (2017). Frequent patterns in ETL workflows: An empirical approach, Data & Knowledge Engineering, Volume 112, 2017, Pages 1-16, https://doi.org/10.1016/j.datak.2017.08.004

Walha Afef, Ghozzi Faiza & Gargouri Faiez, (2024). "Data integration from traditional to big data: main features and comparisons of ETL approaches", The Journal of supercomputing 2024, Vol.80 (19), p.26687-26725, https://doi.org/10.1007/s11227-024-06413-1

Zvonarev Aleksei E., Gudilin Dmitriy S., Lychagin Dmitriy A. & Goryachkin Boris S., (2023). "Extract-Load-Transform (ELT) Process Runtime Analysis and Optimization", https://doi.org/10.1109/REEPE57272.2023.10086728