2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing and Communication Engineering (ICATIECE) | 978-1-6654-9396-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICATIECE56365.2022.10046997

ETL, ELT and Reverse ETL: A business case Study

Bharat Singhal
UPES
Dehradun, India
singhalbharat00@gmail.com

Alok Aggarwal
UPES
Dehradun, India
alok.aggarwal@ddn.upes.ac.in

Abstract - **Most organizations today rely heavily on their data warehouse to make enterprise level decisions. Data Warehouse pulls data from various heterogeneous sources and thus, when setting up a data warehouse, there are three ways to process data: ELT (Extract, Load and Transform), ETL (Extract, Transform and Load) and reverse ETL. It can be challenging to select the best approach when deciding how to implement a data warehouse because it has to do with costs, procedures, performance, and ongoing company improvement. In this paper, we'll be discussing the three approaches and their use cases.**

Keywords: ETL, ELT, Reverse-ETL

I. INTRODUCTION

A data warehouse is a repository for data that is extracted from multiple sources and organized for reporting and analysis. The data in a data warehouse is typically in a denormalized form, which means that it is not normalized to the same level as the data in the source systems [1]. This makes the data in the data warehouse easier to query and analyze.

There are a few different methods of processing data in a data warehouse. The first is called ETL, or extract, transform, and load [2]. This is the process of extracting data from the source systems, transforming it into a format that is suitable for the data warehouse, and loading it into the data warehouse. The second method is called ELT, or extract, load, and transform [3]. This is the process of extracting data from the source systems, loading it into the data warehouse, and then transforming it. Many other approaches have been developed using MANET, WSN and various ML approaches [6]-[27].

1.1 ETL

When working with databases, it is essential to properly plan and design information so that it can be stacked into frameworks for information stockpiling. ETL is a single programming tool that combines three distinct but crucial capabilities to facilitate the preparation of data and database administration. The functions of each of the three procedures will be visible below.

Extract: A source database's information is looked through, and the best part of it is pulled out during this process. The goal of this progression is to extract as much information as possible from the source framework while using as few resources as possible. The concentrating process ought to be planned in such a way that it does not adversely affect the source framework in terms of execution or reaction time [4].

Transform: The information is sifted and purified during this procedure, which also prepares the removed data by combining it with other data or using query tables or administration tools to return it to its original state. The approval of records, the rejection of information (if it is deemed unworthy), and the mixing of information are all part of the change step. One of the most common methods for change transformation is to arrange, separate, clear the duplicates, institutionalize, interpret, and look upward or check the consistency of information sources.

Load: One of the steps in the process is stacking the information into the information distribution center. The subsequent data, such as the extracted and modified data, are compiled by the heap capacity in a manner similar to that of an objective data vault. Several devices interface the extraction, change, and stacking forms for each record from the source, while others physically embed each record as another column into the table of the objective database using SQL embed explanation.

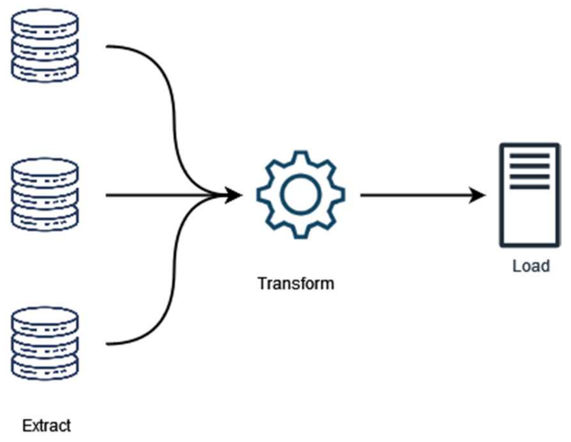

Fig. 1. ETL diagram

1.2 ELT

The Extract, Load, and Transform (ELT) method combines manual coding and the ETL method in a similar arrangement. Similar to the ETL approach, the data is separated. After you have removed your data, you quickly begin the stacking stage, which involves combining all of the data sources into a single, integrated archive [4]. When the data is separated from the source and put into the arranging tables, it is a crude duplicate. This means that the segment names stay the same as in the source database and that you don't add new data or information fields. However, you might channel unnecessary lines and sections as you remove information to avoid wasting resources on unnecessary information. Frameworks would now be able to support large capacity and flexible figures thanks to the current cloud-based framework advancements.

As a result, keeping track of all the extracted crude information requires a large and expanding data pool as well as quick preparation. The objective information distribution center framework currently houses the information that has been extracted from various sources. Using local SQL drivers, the changes and business justifications are connected which uses information streams to move data from the source to the organizing tables.

This saves money and reduces the amount of extra work required by the ETL center level.

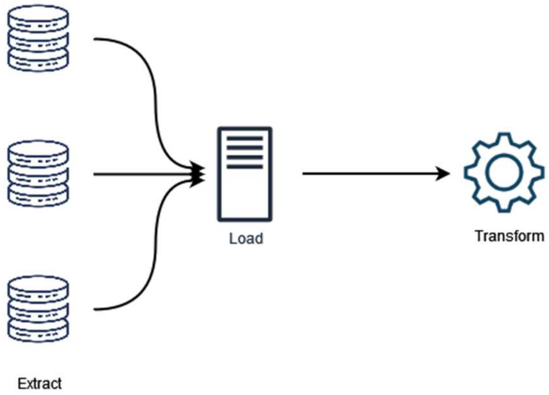

Fig. 2. ELT diagram

1.3 Reverse ETL

The opposite of ETL/ELT is reverse ETL. Reverse ETL makes the data warehouse the source as opposed to the destination. To enable action, data is extracted from the warehouse, processed to meet the data formatting needs at the destination, and fed into an application so it can be used by marketing, sales, support, and other teams in the tools they use. By reentering data into business systems, a Reverse ETL "operationalizes" data across a company [5].

In that regard, a Reverse ETL is utilized in conjunction with other data pipelines rather than replacing ETL or ELT pipelines.

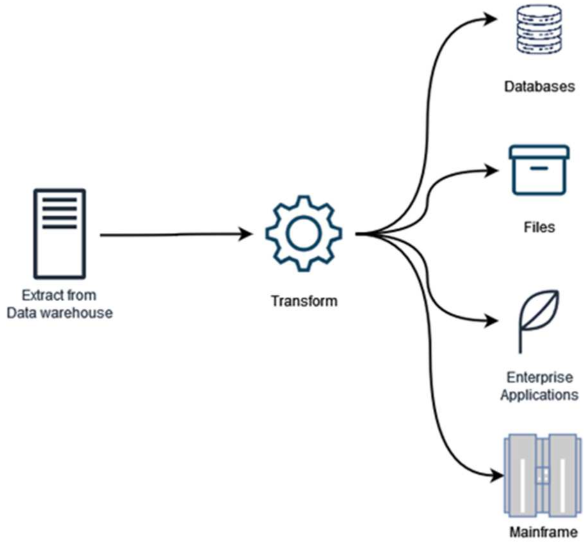

Fig. 3. Reverse ETL diagram

II. BUSINESS SCENARIOS FOR ETL & ELT

The ETL and ELT processes' most difficult step is data transformation. ETL processes are best suited for batch operations with transactional support for monolithic, often legacy, data sources. Despite their OLTP (Online Transaction Processing) nature, these databases do not insert or update data quickly. Most data modification is done manually or in batches. Typically, Non-relational data stores like NoSQL databases, or blob storage or Hadoop file systems are used to ingest from these schema-free, highly dynamic data sources. For the analytics workload, they are then transformed and processed into tabular form.

Additionally, Data based business decisions have to be made in very limited time. The frequency varies from weekly or daily for the generation of these traditional business intelligence reports. Decisions based on live data are almost expected entirely when dealing with large volumes of streaming data. One such illustration is a continuously updated live stream of stock market data.

In such circumstances, ETL applications rapidly slow down. The transformation processes grind to a halt and the queues for incoming data become overflowing, all of which cause the user to wait for an unreasonable amount of time. An alternative approach to data processing known as ELT (Extraction, Loading, and Transformation) is utilized to address this issue.

The location of the data transformation is the primary distinction between ETL and ELT. ELT, in contrast to ETL, does not change anything during transit. The back-end database handles the transformation. This indicates that data is directly pushed into the target data warehouse from the source systems in a staging area. Within the database, the transformation business logic then kicks in. After working with the staged, raw data, the transformation process copies the processed data to a separate area.

Rather than the ETL-native language, all the processing takes place in the native language of data warehouse of the target, So no heavy lifting is required from the pipeline. The rise and widespread adoption of cloud data warehouses are additional factors contributing to the rise of ELT. There are a lot of options for managed cloud-hosted data warehouse where everything is handled by the cloud provider, ranging from Scaling the system to managing the Storage or hardware and any software install requirement. Amazon Redshift, which is a common cloud hosted data warehouse can be provisioned in minutes. Snowflake, a cloud data warehouse, allows for unlimited data storage and complete separation of compute and storage.

Cloud data warehouses are run on powerful machines that have a lot of RAM, multiple core processors, and fast SSD disk storage. Even if one or more nodes fail over, multi-node clusters remain online. Based on load configuration, the number of nodes can dynamically increase or decrease. These are also known as systems with columnar stores for massively parallel processing (MPP). The columnar storage mechanism makes it possible to quickly retrieve data, and the MPP feature makes it possible for queries to be processed in parallel across all CPU cores of each node.

TABLE 1. DIFFERENCE BETWEEN ETL and ELT

Parameter	**ETL**	**ELT**
Optimal Use	Structured data, legacy systems, and relational DBs; transforming data before loading to Data Warehouse.	Quicker, timely data loads, structured and unstructured data, and large dataset; transforming data as per need
Privacy	Personal Identifiable Information can be eliminated in the Pre-load transformation step.	Major safeguards for privacy are required since data is directly loaded.
Transformations	Secondary server performs the transformations required. Precleaning and compute heavy transformation are optimal.	Higher speed and efficiency is achieved since the database performs transformations for load and transform simultaneously.
Maintenance	High maintenance due to the presence of multiple processing servers.	Reduced maintenance burden because of fewer system.
Expenses	Monetary issues due to separate servers	Less Monetary overhead because of simplified data stacks.
Compatibility with Data Lake	Data lake compatibility is not there with ETL	Data lake compatibility is there with ELT.
Output of Data	Output is Structured.	Output can be Structured, unstructured or semi-structured.
Amount of Data	Datasets of Small and moderate volume.	Datasets of Large volume.

III. BUSINESS SCENARIOS FOR REVERSE ETL

In general, marketing, sales, and support work more closely together when it comes to reverse ETL. Reverse ETL assists with email customization for marketing. Several businesses use newsletters to reach out to existing and prospective consumers, typically by developing data-driven email flows that can range in complexity. Reverse ETL enables businesses to import product usage data into Salesforce for Sales and combine it to provide a comprehensive product usage view. They can see, for instance, when someone registers, takes a specific activity, or spends a specific sum of money. For support, agents are better able to aid customers by having a complete view of them.

TABLE 2. DIFFERENCE BETWEEN ETL/ELT and REVERSE ETL

Parameter	**ETL/ELT**	**Reverse ETL**
Synchronization Mode	There can be full or incremental data extraction. We can use the UPSERT operation to merge data into a database or data warehouse	CDC is difficult to apply in Reverse ETL, as the warehouse typically doesn't provide a transaction log or "updated_at" columns. We have to keep track of what needs to be updated and what needs to be created.
Data transformations	We go from specific to general. We extract data from different specific sources to then integrate it into a common destination.	We go from general to specific, having to conform to each business application API.
Data quality	Less Data quality overhead as the destination is database/data warehouse	High data quality overhead as more validation and knowledge of the destination are required.
Failures and job re-execution	ETL/ELT jobs are idempotent, which means that no matter how often you run them, they should produce the same results.	Reverse ETL jobs are not idempotent since the re-execution might result in unwanted side effects as they depend on the business logic of the destination.

VI. CONCLUSION

ETL is a classic paradigm. It works with conventional data center infrastructures, which are being replaced by cloud technologies already. Since the already existing infrastructure or particular deployments are much more inclined to ETL, major companies still prefer this method.

ELT makes use of current cloud technologies effectively, making it the future of data warehousing. It provides key insights that can assist businesses in making the right business decisions and makes it possible for businesses to analyze large data sets with less upkeep. As native data integration tools for Hadoop and NoSQL solutions continue to advance, the scope of ELT may eventually expand.

The emergence of a new generation data stack highlights an important trend: businesses must embed data capabilities inside teams across business divisions rather than keeping them in centralized silos (data warehouses). Reverse ETL solutions that handle data operationalization, or in other words, close the operational analytics loop, are thus part of the future of the modern data stack.

REFERENCES

[1] Kumar A. et al. "Simulation and Analysis of Authentication Protocols for Mobile Internet of Things (MIoT)," *PDGC,* 2014, pp. 423-428.

[2] P. Gupta et al., "Trust and reliability based scheduling algorithm for cloud IaaS," *Lect. Notes in EE.* vol. 150, 2013, pp. 603-607.

[3] Govil Kapil et al., "*Cluster Head Selection Technique for Optimization of Energy Conservation in MANET*," *PDGC,* 2014, pp. 39-42.

[4] *A. Kumar et al., "*Lightweight Cryptographic Primitives for Mobile Ad Hoc Networks," *RTCNDSSCCIS*, vol. 335, 2012, pp. 240-251.

[5] Kumar A. et al., "Performance analysis of MANET using elliptic curve cryptosystem," *ICACT*, 2012, pp. 201-206.

[6] Singh T., Srivastava D. K. and Aggarwal A., "A novel approach for CPU utilization on a multicore paradigm using parallel quicksort," *CICT*, 2017, pp. 1-6.

[7] Mittal S. et al., "Situation recognition in sensor based environments using concept lattices," *IIT,* 2012, pp. 579-584.

[8] SK Gupta et al., "Routing Algorithm for Energy Conservation in MANET," *CICN*, 2015, pp. 165-167.

[9] Singh V. et al., "A holistic, proactive and novel approach for pre, during and post migration validation from subversion to git," *CMC*, vol. 66, no.3, pp. 2359-2371, 2021.

[10] S. Aggarwal et al., "Optimized method of power control during soft handoff in downlink direction of WCDMA systems," *PDGC,* 2014, pp. 433-438.

[11] Rajput I.S. et al., "An efficient parallel searching algorithm on Hypercube Interconnection network," *PDGC,* 2012, pp. 101-106.

[12] Goyal M.K. et al., "Effect of change in rate of genetic algorithm operator on composition of signatures for misuse intrusion detection system," *PDGC,* 2012, pp. 669-672.

[13] S. Aggarwal et al., "Performance analysis of soft handoff algorithm using fuzzy logic in CDMA systems," *PDGC,* 2012, pp. 586-591.

[14] MK Goyal et al. "QoS based trust management model for Cloud IaaS," *PDGC*, 2012, pp. 843-847.

[15] Kumar A., Krishan G. et al., "Design and Analysis of Lightweight Trust Mechanism for Secret Data using Lightweight Cryptographic Primitives in MANETs," *Jour. of N/w Security,* vol. 18, no. 1, pp. 1-18, 2016.

[16] Bijalwan D. et al., "Automatic text recognition in natural scene and its translation into user defined language," *PDGC,* 2014, pp. 324-329.

[17] Aggarwal S. et al., "On challenges and opportunities in second wave of ICT revolution for south Asian countries," *PDGC,* 2012, pp. 597-602.

[18] Singh V. et al. "A digital Transformation Approach for Event Driven Micro-services Architecture residing within Advanced vcs," *CENTCON*, 2021, pp. 100-105.

[19] S. Aggarwal et al., "Trends in power control during soft handoff in downlink direction of 3G WCDMA cellular networks," *PDGC,* 2012, pp. 603-608.

[20] V. Singh et al., "DevOps based migration aspects from Legacy Version Control System to Advanced Distributed VCS for deploying Micro-services," *CSITSS*, 2021, pp. 1-5.

[21] Aggarwal S. et al., "Soft handoff analysis and its effects on downlink capacity of 3G CDMA cellular networks," *PDGC,* 2012, pp. 1-6.

[22] S. Aggarwal et al., "Trends in power control during soft handoff in downlink direction of 3G WCDMA cellular networks," *Proc. PDGC,* 2012, pp. 603-608.

[23] Singh V. et al. "A digital Transformation Approach for Event Driven Micro-services Architecture residing within Advanced vcs," *CENTCON*, 2021, pp. 100-105.

[24] Aggarwal S. et al., "On challenges and opportunities in second wave of ICT revolution for south Asian countries," *PDGC*, 2012, pp. 597-602.

[25] S. Aggarwal et al., "Performance analysis of soft handoff algorithm using fuzzy logic in CDMA systems," *PDGC,* 2012, pp. 586-591.

[26] Rajput Iswar Singh et al., "An efficient parallel searching algorithm on Hypercube Interconnection network," *PDGC,* 2012, pp. 101-106.

[27] Goyal MK et al., "Effect of change in rate of genetic algorithm operator on composition of signatures for misuse intrusion detection system," *PDGC*, 2012, pp. 669-672.