# WHAT'S ETL? EXTRACT, TRANSFORM & LOAD EXPLAINED

Data is the key driving force in most applications today. All data—from simple application logs and system metrics to user data—are quantifiable data that can be used for data analytics. The importance of data has skyrocketed with the growing popularity and implementation of big data, analytics, and data sciences.

Short for extract, transform & load, ETL is the process of aggregating data from multiple different sources, transforming it to suit the business needs, and finally loading it to a specified destination (storage location).

With the growing popularity of databases in 1970, ETL was introduced as a process for loading data for computation and analysis. However, ETL has now evolved to become the primary method for processing large amounts of data for data warehousing and data lake projects. Thus, ELT has become an important factor in an organizational data strategy.

The primary goals of adapting ETL in organizations are to:

1. Create a consolidated view of your data in various formats and multiple locations.
2. Streamline the reviewing process leading to better business decisions.

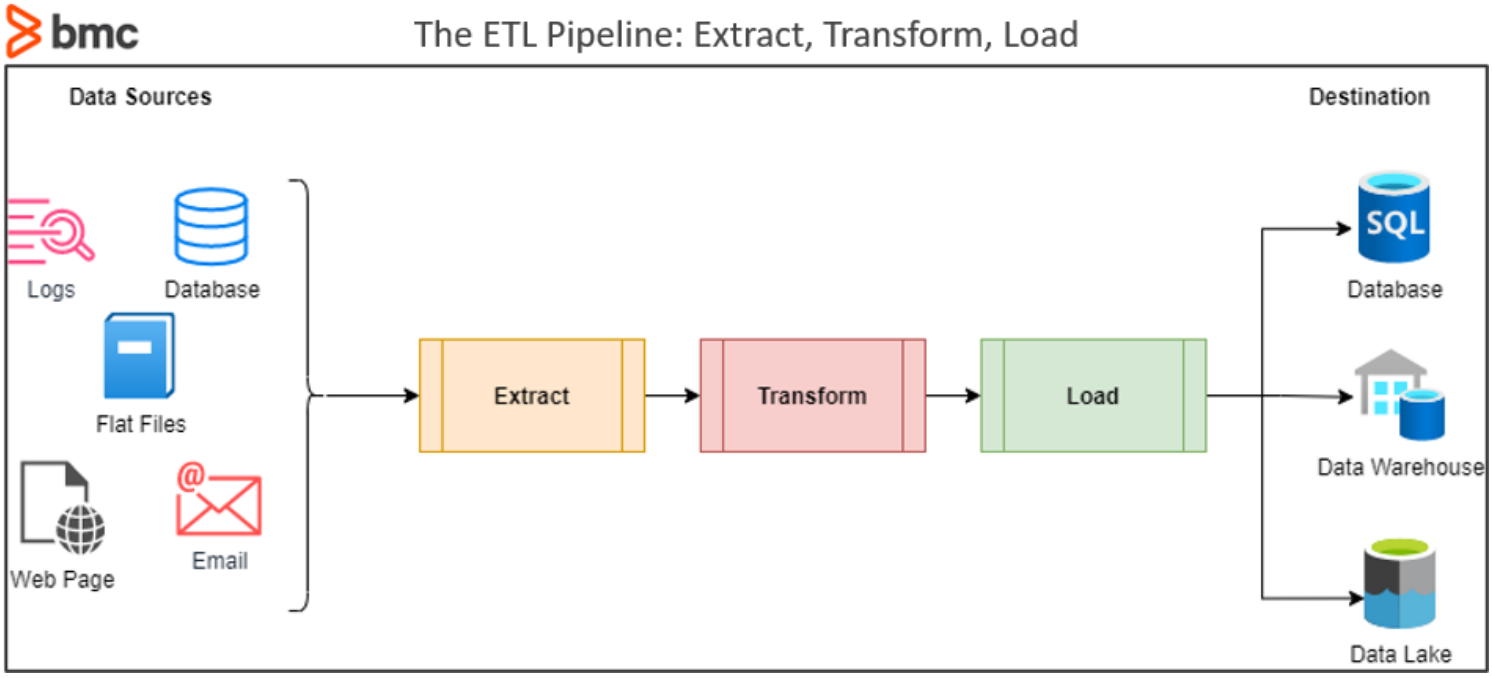
Let's take a deeper look into ETL in this article.

# Extract, transform & load basics

ETL consists of three components:

- Extract
- Transform
- Load

Each of these components or tasks represents a separate function of an ETL pipeline. In this section, we will dive into the exact functionality of these components.




# ETL: What is Extract?

This is the first step of the ETL process. The basic workflow of the extraction process is to copy or export raw data from different locations and store them in a staging location for further processing. There, the locations from where the raw data is extracted are known as sources or source locations. Sources can consist of any type of structured or unstructured data such as:

- SQL or NoSQL servers
- Flat files
- Emails
- Web pages
- Logs
- CRM and ERP system
- Metrics
- Spreadsheets

It might not be possible to pinpoint exact subsets of data depending on the source in a typical extraction phase. Thus, as a general rule, a broader range of data is extracted to ensure that all the required data is collected.

This is also a crucial factor when dealing with new data sets where users might know the contents as it will be ideal for exploratory analysis with wide-ranging data. The volume and the data sources

are dependent on the requirements and the organizational needs.

Extraction can be applied to process kilobytes of data to terabytes of data and can vary between a couple of minutes to days or can be a real-time process.

# Common data extraction methods

Let's look at common data extraction methods.

## Partial extraction (with update notification)

This method is based on a notification strategy where the system provides an update notification to carry out the extraction process when a record or data set is modified in a source location.

This is the simplest extraction method.

## Partial extraction (without update notification)

Some sources will not be able to provide an update notification, yet they will be able to indicate the updated records or data. In this scenario, the process will:

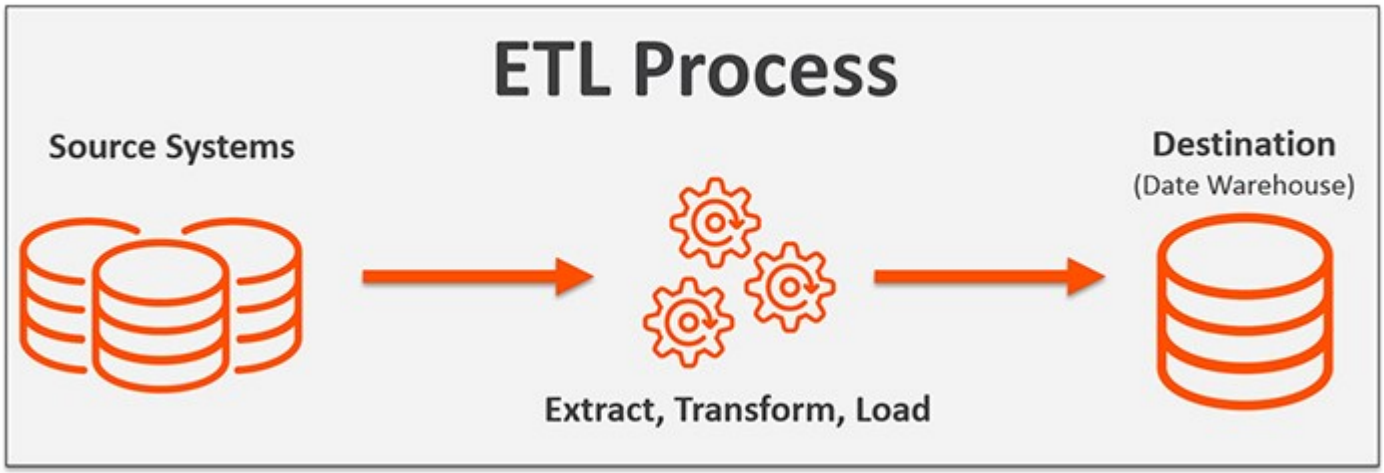
- Query the data source intermittently
- Identify the updated data for the extraction process

## Full extraction

Full extraction is only required if the system is incapable of identifying updated data. In these instances, the only option is to reload the entire data set. However, this method should only be used on small datasets as it can be a time-consuming and resource-intensive process.

## Note on extractions

In any extraction method, we have to ensure that it will not affect the performance of the underlying system. For example, assume that extraction from a production database causes adverse performance issues that hinder the overall application performance. In that case, extraction should be carried out by different means, such as a read replica of the production database.



# ETL: What is Transform?

This is the transformation stage of the ETL process. There, the data extracted from various sources and within the staging area (temporary storage) goes through a data processing phase to transform so that they can be used for analytics. Essentially this converts raw data to a more focused and meaningful data set.

This data transformation process can consist of several tasks that will be used to apply different kinds of data transformations.

- **Cleaning and standardization** resolves inconsistencies, missing values, remove unnecessary data, and format the data to a standardized format such as specific date-time formations, numeric formats, etc.
- **Verification and validation** verifies if the raw data consists of the required information while removing unusable data and identifying data anomalies.
- **Filtering and sorting** organizes the data according to specific requirements such as type, field, value, etc.
- **Deduplication** discards or excludes redundant data from getting processed.

In addition to these tasks, the transformation process can involve advanced functionality such as:

- **Data audits to ensure data quality and compliance.** This is highly useful when interacting with personally identifiable information (PII) as a method to ensure only required, and approved data are processed.
- **Data encryption and protection.** Some data sets will need to be encrypted and protected in these instances to ensure regulatory compliance. The transform process is responsible for carrying out these duties.
- **Performing calculations, translations on the raw data.** This includes tasks like calculating an entirely new field from an existing data set, translating data to a different language, changing row and column headers, etc.
- **Formatting, joining, or splitting data** to match internal or existing schemas of the target analytical or storage system.

# ETL: What is Load?

This is the final step of the ETL process, which involves loading the transformed data to its final destination. The destination can range from a simple database to a massive data warehouse, depending on:

- The size and complexity of the underlying data
- The overall organizational requirements

The load process can be divided into two types: full loading and incremental loading.

Typically, **the full loading process** happens only at the first data loading task to populate the destination with all the available data. Only after that, **the incremental loading** that loads the updated data happens. These increments can be either:

- Streaming increments to handle small volumes of data with regular updates
- Batch increments to handle a large volume of data

The only other instance a full loading will be required is disaster recovery or to migrate the destination data source.

## Benefits of the ETL process

- Provides a standardized process to aggregate and transform row data and store the transformed data for further analytics.
- Facilitates efficient data analytics by introducing an automated data processing pipeline to gather and format data without the need to offload the data transformation task to other BI or analytical tools.
- Handles Big Data and enables advanced data profiling and cleaning.
- Easily obtains deep historical context for the organization and facilitate impact analysis.
- Leverages AI and ML tools more easily with ETL pipelines to increase the accuracy and effectiveness of the analytical process.
- Quickly adapts to changing technological and integration needs.

## ETL supports data warehouses & data lakes

Organizations need to store and analyze large data sets consisting of historical data and have adequate expandability to support ever-growing data needs. Thus, data warehousing has become the common practice. Major cloud services provide warehousing services such as AWS Redshift and Google BigQuery.

However, the increased complexity of data and the growing need to support a multitude of data sources has given birth to data lakes. These data lakes far surpass the capabilities of data warehouses and allow users to store all structured or unstructured data at any scale.

*(Compare use cases for data lakes & data warehouses.)*

In both these instances, ETL provides the ideal framework to extract data from different sources, transform it, and then store it in the appropriate storage service.

Users are free to implement ETL to support their exact needs as ETL is not bound to a specific technology or system. With the need to keep the data up to date, ETL provides the ideal solution to create data ingestion pipelines that can be used to aggregate data from multiple sources. Organizations can create a unified information base for all their analytical needs with a data warehouse or a data lake with an ETL pipeline.

## ETL tools & services

As we are dealing with data, any data service such as databases, data warehouses, or data lakes can be considered a part of the ELT process either at the Extract or Load phases. However, it can be a complex process to build the pipeline from the ground up when creating ETL pipelines. So there are some specialized tools and services to simplify this process.

- **Azure Data Factory.** Fully managed serverless data integration service that can be used to create ETL pipelines.
- **AWS Glue.** AWS-based managed serverless data integration service with support for both visual and code-based interfaces.

- **Xplenty**. A cloud-based scalable ETL service to create ETL pipelines.
- **Hevo.** A no-code data pipeline platform to create real-time data integration pipelines.
- **Matillion.** Cloud data integration and transformation platform with a comprehensive toolset to create ETL pipelines for any enterprise need.

## ETL summary

In this post, we had a look into the basics of ETL or Extract, Transform, and Load process. ETL is the backbone for most modern data ingestion and integration pipelines that facilitate accurate and efficient analytics. The importance of ETL will only grow in the future with the unprecedented demand for data.

## Related reading

- BMC Machine Learning & Big Data Blog
- What Is a Data Pipeline?
- Logstash 101: Using Logstash in a Data Processing Pipeline
- Using Python for Big Data & Analytics (Python is Perfect for Big Data)
- DataOps Explained: Understand how DataOps leverages analytics to drive actionable business insights
- Data Ethics for Companies