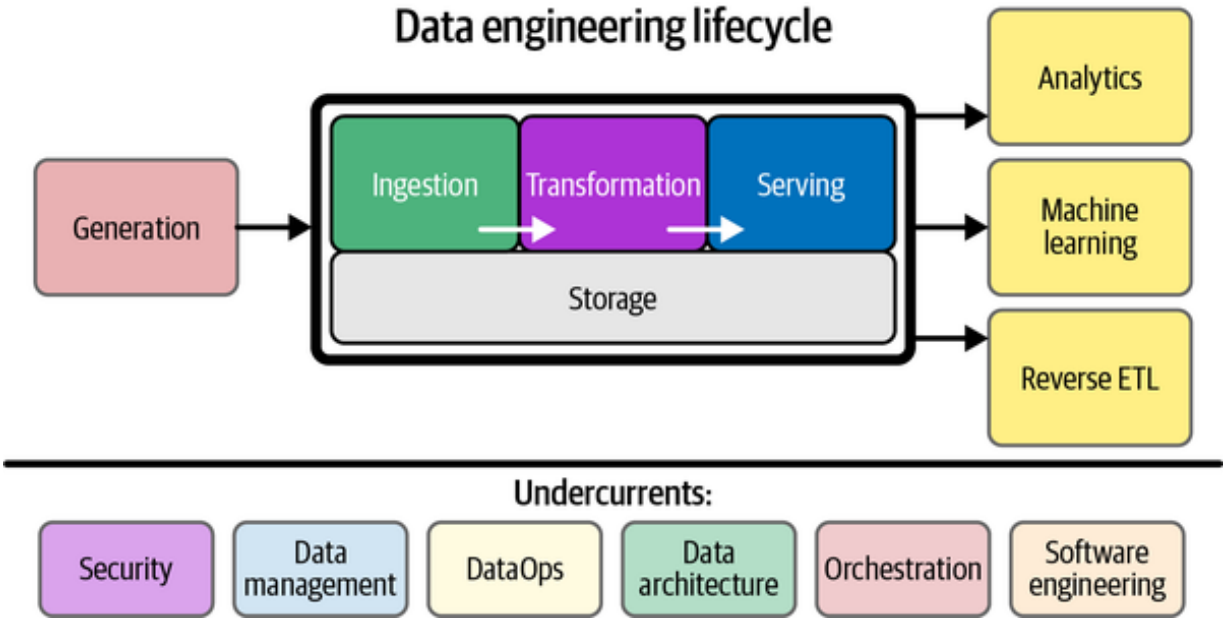
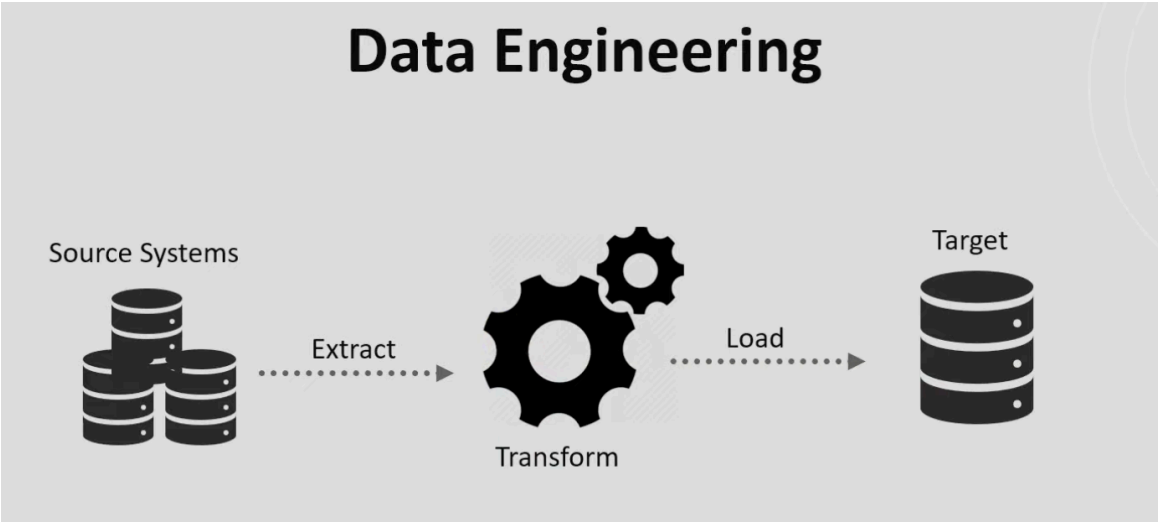
Data Engineering - Overview

- Data engineering is the process of designing and building large-scale data collection, storage, and analysis systems that let people collect and analyze raw data from multiple sources and formats.
- These systems empower people to find practical applications of the data, which businesses can use to thrive.



- In a nutshell, data engineers put up and maintain the organization's data infrastructure, ready it for analysis by data analysts and scientists.

Why Is Data Engineering Important?



- Companies of all sizes have huge amounts of disparate data to comb through to answer critical business questions.
- Data engineering is designed to support the process, making it possible for consumers of data, such as analysts, data scientists and executives, to reliably, quickly and securely inspect all of the data available.
- Data analysis is challenging because the data is managed by different technologies and stored in various structures.
- For example, consider all of the data a brand collects about its customers:
    - One system contains information about billing and shipping
    - Another system maintains order history
    - And other systems store customer support, behavioural information and third-party data.
- Together, this data provides a comprehensive view of the customer.
- Data engineering unifies these data sets and lets you find answers to your questions quickly and efficiently.

What Do Data Engineers Do?

- Data engineering is a skill that is in increasing demand. Data engineers are the people who design the system that unifies data and can help you navigate it.
- Data engineers perform many different tasks including:
    - **Acquisition:** Finding all the different data sets around the business
    - **Cleansing:** Finding and cleaning any errors in the data
    - **Conversion:** Giving all the data a common format
    - **Disambiguation:** Interpreting data that could be interpreted in multiple ways
    - **Deduplication:** Removing duplicate copies of data

- Once this is done, data may be stored in a central repository such as a data lake or data lakehouse or data warehouse.

Target Audience

- The target audience for data engineering is **business stack holders** which apply analytics for business processes.
- The **AI Application developers**, who need adequate data for building efficient cognitive solutions.
- The **Data analysts** professional who are generally involved in Exploratory data analysis using the raw data.
- **Data scientists** who use the data to develop and deploy machine learning models for business.

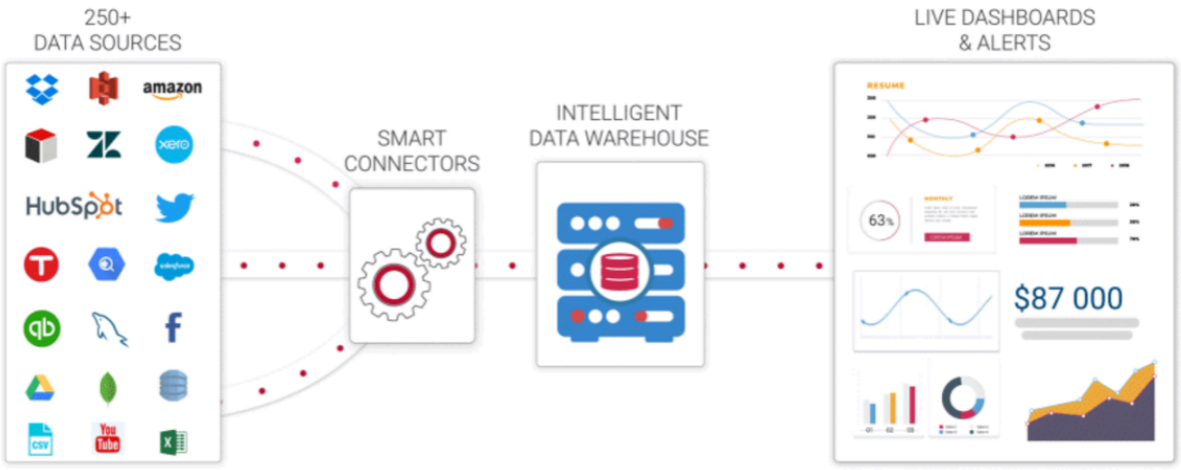
Basic Terminologies in Data Engineering

- Data engineers use many different tools to work with data. They use a specialized skill set to create end-to-end data pipelines that move data from source systems to target destinations.
- Data engineers work with a variety of tools and technologies, including:
    - **SQL:** Structured Query Language (SQL) is the standard language for querying relational databases.
    - **Cloud Data Storage:** Including Amazon S3, Azure Data Lake Storage (ADLS), Google Cloud Storage, etc.
    - **Query Engines:** Engines run queries against data to return answers. Data engineers may work with engines like Dremio Sonar, Spark, Flink, Kafka and others.
    - **Programming languages:** Python, Scala, Java, Spark, Go etc.,
- Data engineering relies upon several big data technologies, Following is a list of tools or technologies which are included as a part of industry best practices.

- Hadoop cluster, Apache Spark, Splunk, Apache Flink, Azure HDinsight.
- NoSQL data stores like Apache Cassandra database, MongoDB.
- In-memory cache databases like Redis, SAP HANA.
- Data processing tools such as Apache Kafka, Apache NiFi, Informatica Cloud services.
- Cloud-based tools like ASES data pipelines, Google Big Query, and Azure Data Factory.
- Standard RDBMS and file systems.
- Various OS-specific scripting like Linux Shell scripting, windows batch, and Power shell scripting.
- Cloud storage like S3.
- API based tools like AWS API gateway to prevision the data APIs for Souring data and deploying analytics
- Time series data stores.
- IoT specific tools like Node-Red.

Data Warehouse

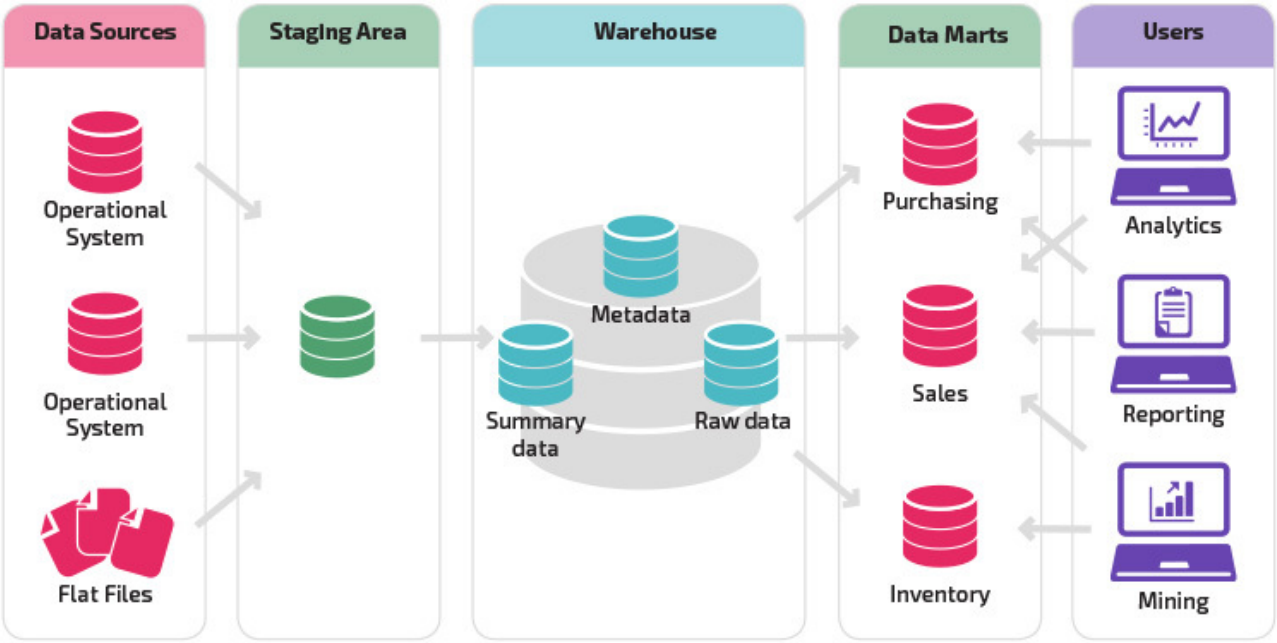
- A data warehouse is a database that stores all of your company's historical data and allows you to conduct analytical queries against it.



- A data warehouse is a relational database that is optimized for reading, aggregating, and querying massive amounts of data from a technical point of view.
- The data warehouse, which serves as an organization's single source of truth, streamlines reporting and analysis, decision-making, and metrics forecasting.
- Four essential components are combined to create a data warehouse:
    - Data warehouse storage.
    - Metadata.
    - Access tools.
    - Management tools.

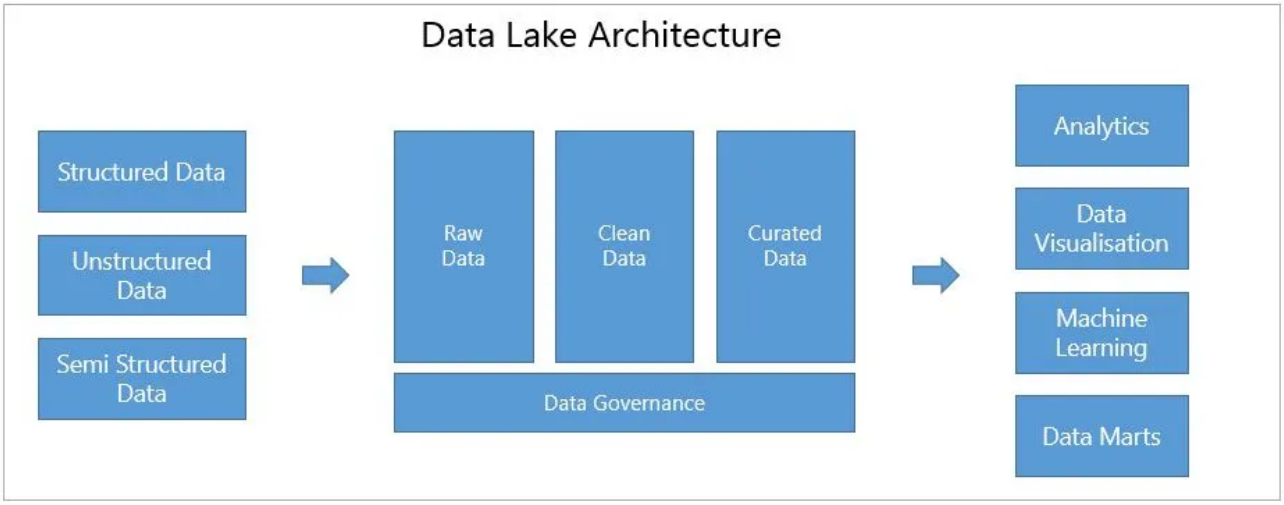
Data marts

- A Data mart is a smaller data warehouse (their size is usually less than 100Gb.). They become necessary when the company (and the amount of its data) grows and it becomes too long and ineffective to search for information in an enterprise DW.



- Instead, data marts are built to allow different departments (e.g., sales, marketing, C-suite) to access relevant information quickly and easily.
- There are three main types of data marts.
    - **Dependent data marts** are created from an enterprise DW and use it as a main source of information (it's also known as a top-down approach).
    - **Independent data marts** are standalone systems that function without DWs extracting information from various external and internal sources.
    - **Hybrid data marts** combine information from both DW and other operational systems.

Data lake



- A Data lake is a vast pool for saving data in its native, unprocessed form.
- A data lake stands out for its high agility as it isn't limited to a warehouse's fixed configuration.
- A data lake uses the ELT approach swapping transform and load operations. Supporting large storage and scalable computing, a data lake starts data loading immediately after extracting it, handling raw — often unstructured — data.

- A data lake is worth building in those projects that are going to scale and would need a more advanced architecture.

OLTP(online transactional processing) vs OLAP(online analytical processing)

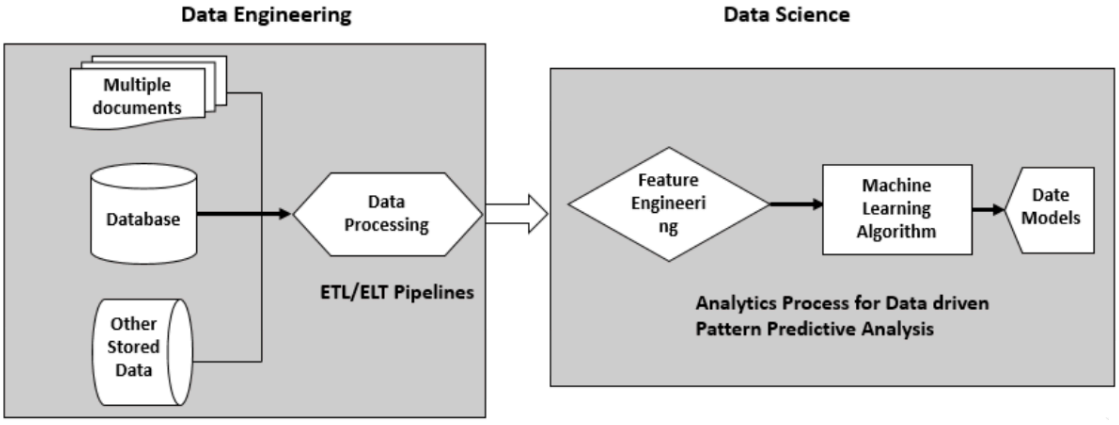
- OLTP and OLAP are both online database systems.
- An OLTP system usually processes a high number of short transactions like select or insert statements in basic SQL terms. Queries should be fast and the system should maintain a high level of data integrity.
    - Example purchase transaction in SQL database is an OLTP system. It will modify the database.
- An OLAP system by contrast will typically run a lower volume of transactions, but they could be longer running queries. An OLAP system might also run these queries over aggregated historical data, which could have been sourced from other OLTP systems, even multiple ones.
    - So following our previous example, this system could be used to produce a report on purchases and purchasing trends over time.It will query the database.
- ETL frameworks will move data from OLTP system to OLAP system.

Scope of Data Engineering

The Scope of data engineering mostly involves the pre-processing of data which reduces the overheads for data scientists and analysts for data preparation stages.

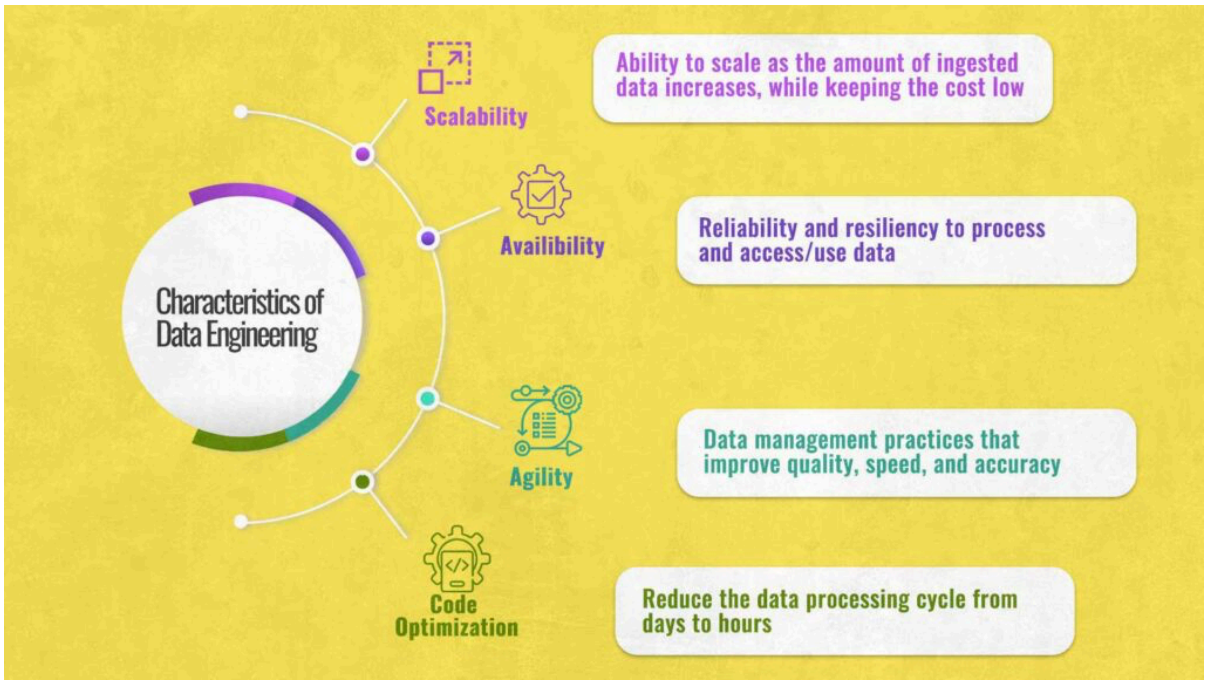
To understand it better following is a high-level framework overview of the data engineering setup for data science.

- In the diagram shown Data engineering is the first phase that links to Data science as the second phase.



- It collects raw data from various source applications, file systems, IoT sensors, and other file storage through ETL(Extract Transform, Load) or ELT(Extract, Load, Transform) pipeline.
- ETL is mainly for the implementation of the data warehouse, whereas ELT is for Big data frameworks.
- Data engineering includes data quality processes and transformation techniques.
- Store the pre-processed data in the data warehouse or data lakes for subsequent use.
- The set up provides input data to the Data Science framework.
- Data Analyst and Data Scientists do initial exploratory analysis for the feature engineering process.
- The data helps to generate Business Intelligence reports and charts apart from machine learning applications.
- Feature engineering is an iterative process to further optimize the data set to be processed by Machine learning.
- Data scientists apply several machine learning models iteratively to generate a best-fit Machine learning model for the use case.
- The input data is helpful to train and test the model while developing.

Data Engineering: Moving Beyond Just Software Engineering

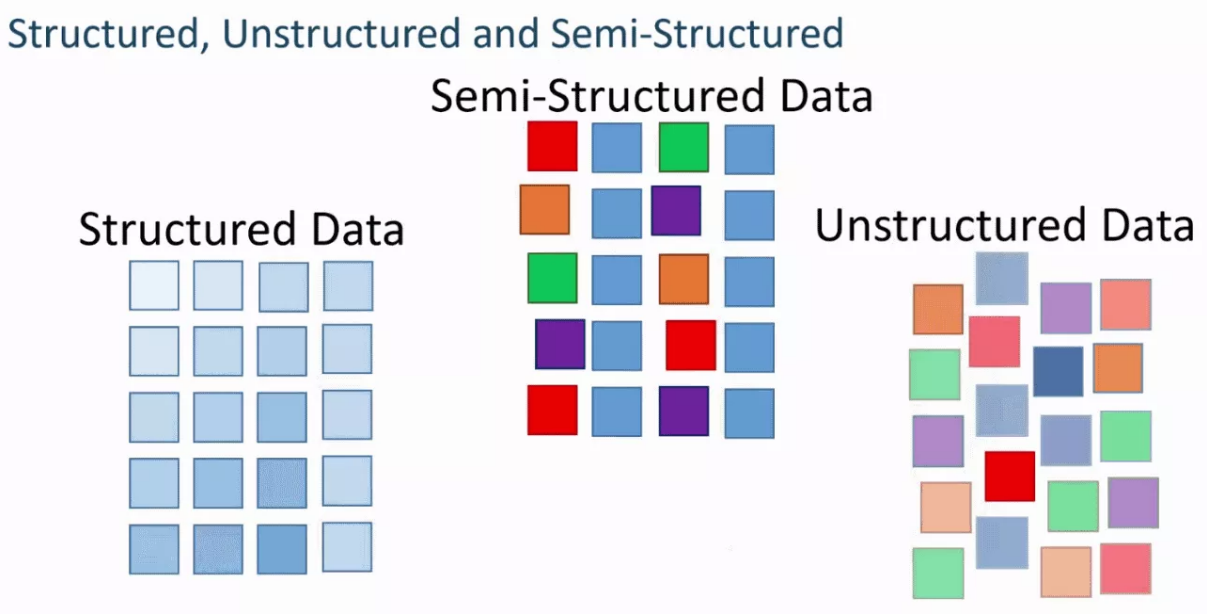


- Software engineering is well-known for its programming languages, object-oriented programming, and operating system development.
- Data engineering helps firms to collect, generate, store, analyze, and manage data in real-time or in batches.
- Traditional software engineering approaches entail mostly stateless software design, programming, and development.
- Data engineering, on the other hand, focuses on scaling stateful data systems and dealing with various levels of complexity.
- In terms of scalability, optimization, availability, and agility, there are also disparities in the complexity of the two fields.

Types of Data

A data type is a formal classification of the type of data being stored or manipulated within a program.

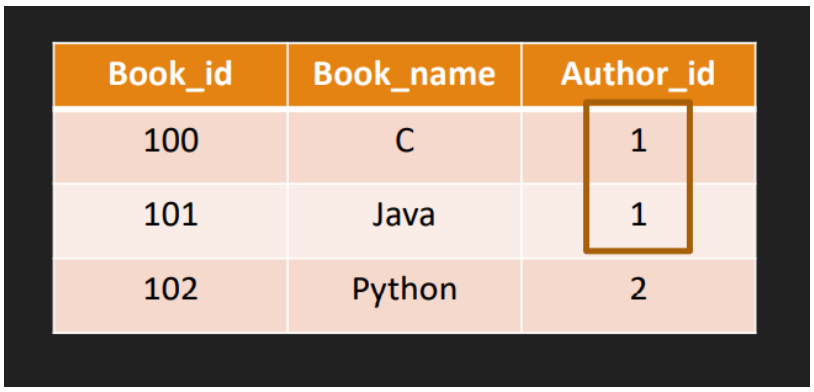
- Structured.
- Semi-Structured
- Unstructured



Structured Data

- Structured data is data whose elements are addressable for effective analysis.
- It has been organized into a formatted repository that is typically a database.
- It concerns all data which can be stored in database SQL in a table with rows and columns. They have relational keys and can easily be mapped into pre-designed fields.
- Today, those data are most processed in the development and simplest way to manage information. Example: Relational data.
    - Tabular Data

- Represented by Rows & Columns
- SQL can be used to interact with data
- Fixed Schema
- Each row has same number of columns
- Relational Database are structured
- MySQL, Oracle SQL, PostgreSQL, MSSQL



Book_id	Book_name	Author_id
100	C	1
101	Java	1
102	Python	2

Semi-Structured Data

- Semi-structured data is information that does not reside in a relational database but that has some organisational properties that make it easier to analyze.
- With some processes, you can store them in the relation database (it could be very hard for some kind of semi-structured data), but Semi-structured exist to ease space. Example: XML data.
    - Each Record has variable number of Properties
    - No Fixed Schema
    - Flexible structure
    - NoSQL kind of Data
    - Store data as key-value pair

- JSON – Java Script object Notation are base way to represent semi structure data - MongoDB, Cassandra, Redis, Neo4j.

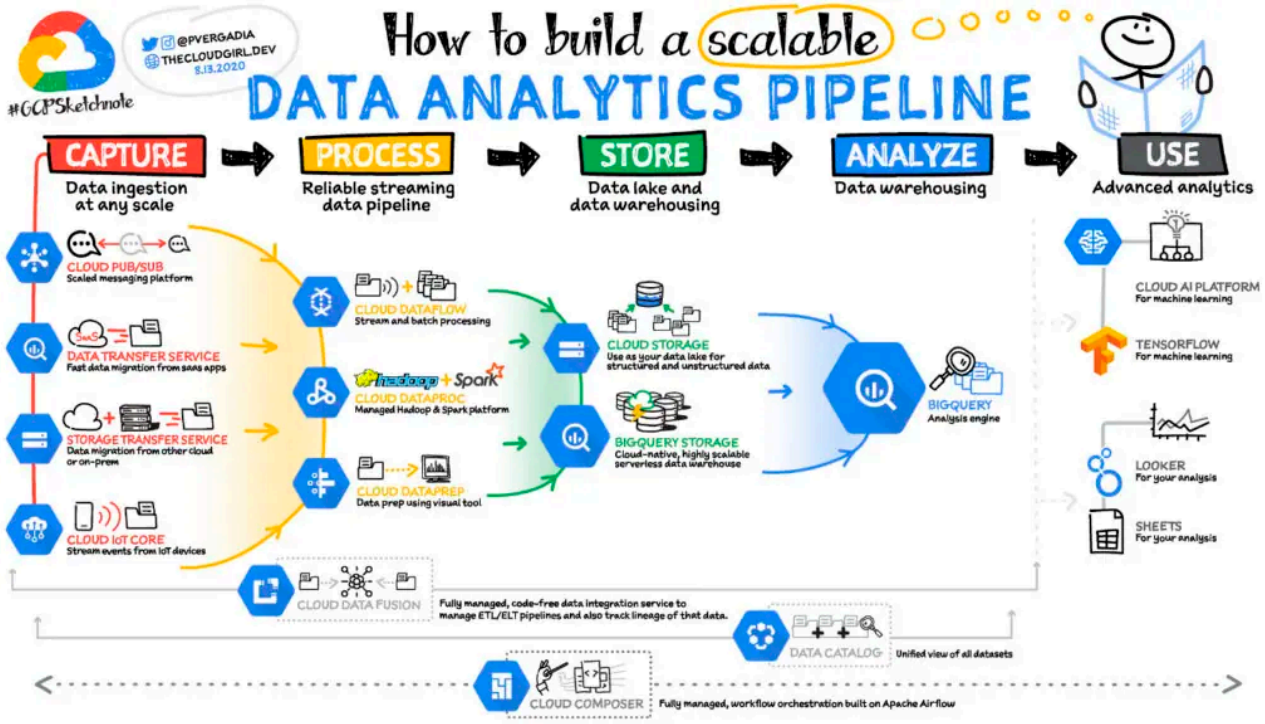
```
{
    'campaing_id': 13421234,
    'campaign_name': 'coolName',
    'ad1': {
        'id': 1245344,
        'type': 'banner',
        'metrics': {
            'clicks': [
                {
                    'time': '2016-03-14T08:00:00+0000',
                    'value': '5'
                },
                ...
```

Unstructured Data

- Unstructured data is a data which is not organized in a predefined manner or does not have a predefined data model, thus it is not a good fit for a mainstream relational database.
- So for Unstructured data, there are alternative platforms for storing and managing, it is increasingly prevalent in IT systems and is used by organizations in a variety of business intelligence and analytics applications. Example: Word, PDF, Text, Media logs.
- No Predefine Structure in Data
- Image, video data, natural Language are example of unstructured data

What is a data pipeline?

- A data pipeline is a series of data processes that extract, process and load data between different systems.
- There are two main types of data pipelines:
    - batch-driven
    - real-time.



What is Batch Processing

- Batch-driven: Batch data pipelines only process data at a certain frequency and are often scheduled by a data orchestration tool, such as Airflow, Oozie, or Cron.
- They usually process a large batch of historical data all at once, therefore taking a long time to finish and inducing more data delay at the end system.
- Batch data is simply data that is gathered together, usually within a defined time window and loaded into a system, all in one go.
- Defined Start & End of data
- data size is known
- Processing High volume of data after certain periodic interval
- Long time to process data
- Payment processing

- For example, a batch-based data pipeline downloads the previous day's data from an API at 12 AM every day, transforms the data, and then loads it into a data warehouse.

What is Streaming Processing

- Real-time: Real-time data pipelines process new data as soon as it is available and there is almost no delay between the source and end system.
- The architecture for real-time data processing is very different from that of batched pipelines because data is treated as a stream of events instead of chunks of records.
- Streaming on the other hand is just as it sounds, the **continuous collection of data** into a system. Data is collected as it happens if you like every transaction or metric being sent immediately to the system. It can be harder to process streaming data because of the sheer volume of it.
- Unbounded, No End defined
- Data is processed as it arrives
- Size is unknown
- No much heavy processing
- take millisecond - seconds to process data
- Stock data processing

Advantages

Some of the major advantages:

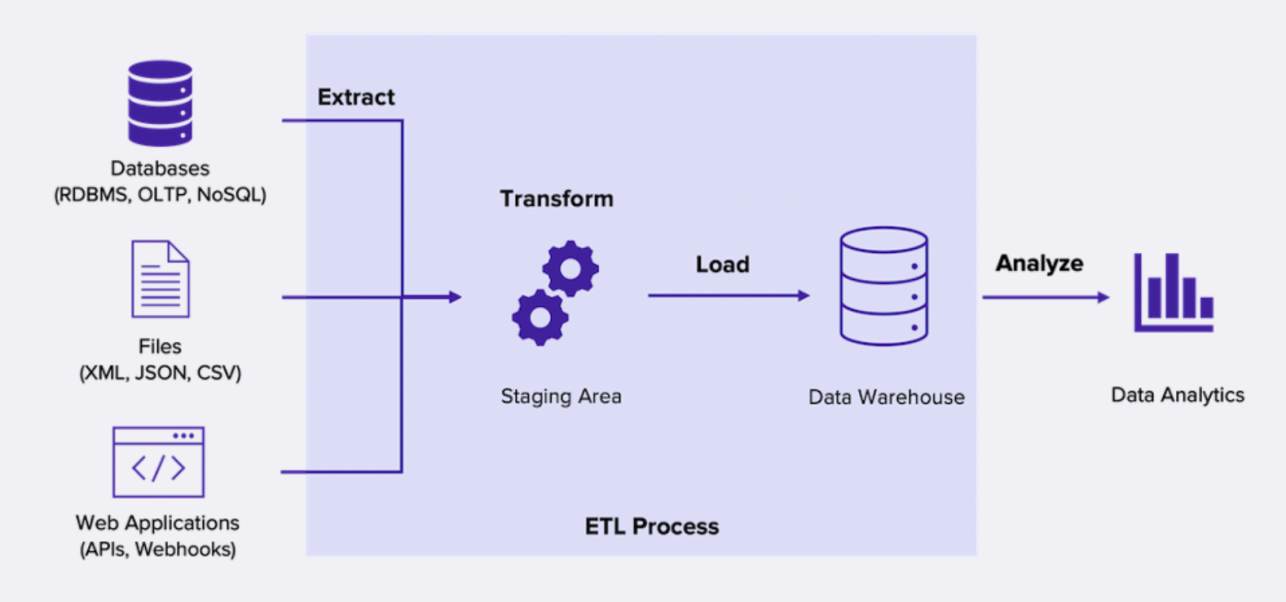
- It helps to pre-process data of various formats and various heterogeneous sources to a standard format and structure.
- Automate the pipeline for incremental data or the latest data to be used by the analytics solution by implementing automation tools for batch processing and scheduling.

- Real-time analytics support by data engineering by using the latest and best practices, technologies like Apache Kafka, Spark, and data-bricks.
- Applying the governance policies and security compliance of data by masking and encrypting the confidential information by applying various business rules.
- Creating production-ready data for faster completion of analytics project implementations.
- Customization of the data structure by joining and wrangling data to be best for the machine learning algorithm needs to be based upon the data scientist's recommendation.

What is ETL?

- ETL stands for extract, transform, and load and is a traditionally accepted way for organizations to combine data from multiple systems into a single database, data store, data warehouse.
- ETL can be used to store legacy data, or as is more typical today aggregate data to analyze and drive business decisions.
- Organizations have been using ETL for decades. But what's new is that both the sources of data, as well as the target databases, are now moving to the cloud.
- Today's modern ETL solutions must cope with the accelerating volume and speed of data.

- Additionally, the ability to ingest, enrich and manage transactions, and support both structured and unstructured data in real time from any source —whether on-premises or in the cloud.



- The three operations happening in ETL
  - Extract
  - Transform
  - Load
- **Extraction**
  - Extraction is the process of retrieving data from one or more sources—online, on-premises, legacy, SaaS, or others. After the retrieval, or extraction, is complete, the data is loaded into a staging area.
- **Transform**
  - Transformation involves taking that data, cleaning it, and putting it into a common format, so it can be stored in a targeted database, data store, data warehouse, or data lake.
  - Cleaning typically involves taking out duplicate, incomplete, or obviously erroneous records.

- **Load**
    - Loading is the process of inserting that formatted data into the target database, data store, data warehouse, or data lake.

Advantages of ETL

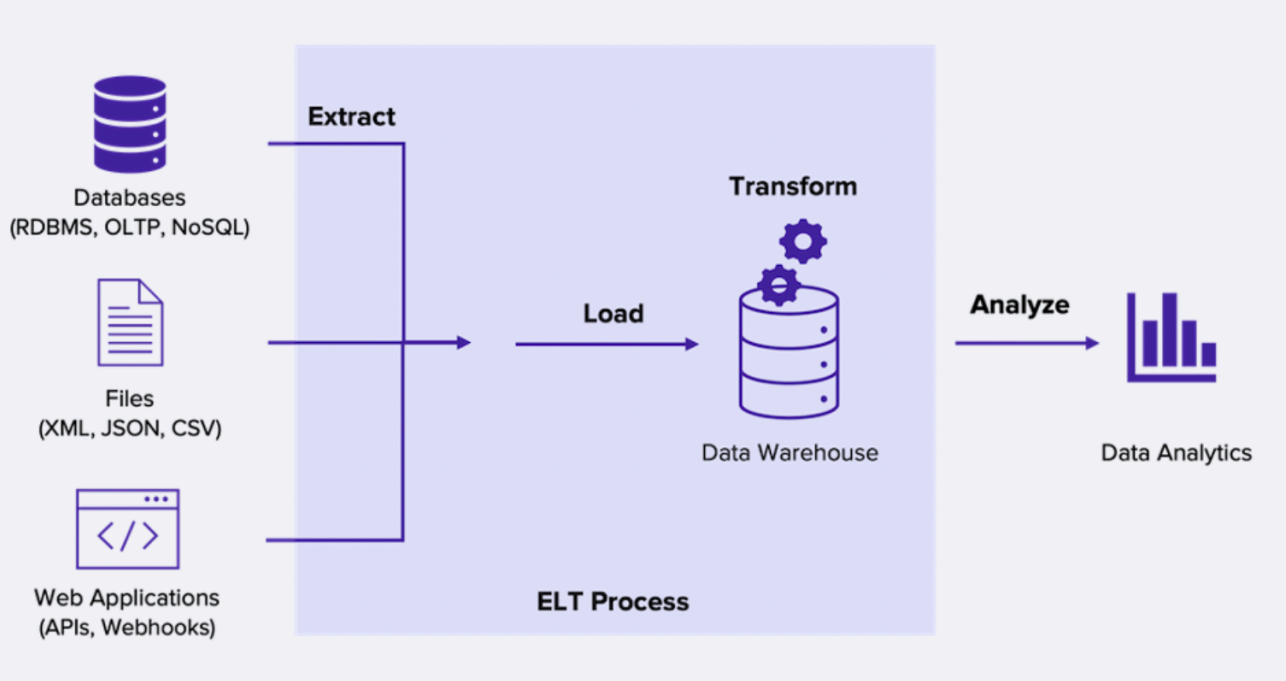
- Maturity:
    - ETL was developed first and has been in practice for more than two decades. This means that there are more engineers with experience in ETL implementations and more ETL tools in the marketplace to build data pipelines within organizations.
- Compliance:
    - ETL transforms data before it reaches its destination. When companies are subject to data privacy regulations such as GDPR, ETL allows them to remove, mask, or encrypt sensitive data before it' s loaded to the data warehouse to ensure compliance.

Disadvantages of ETL

- Higher upfront cost:
    - Defining business logic and transformations can increase the scope of a data integration project.
- Frequent maintenance:
    - ETL data pipelines handle both extraction and transformation. But they have to undergo refactors if analysts require different data types or if the source systems start to produce data with deviating formats and schemas.

What is ELT?

- ELT is an acronym for “Extract, Load, and Transform” and describes the three stages of the modern data pipeline. The ELT process is more cost effective then ETL, is appropriate for larger, structured and unstructured data sets and when timeliness is important.
- In this process, data gets leveraged via a data warehouse in order to do basic transformations. That means there's no need for data staging.
- ELT uses all different types of data - including structured, unstructured, semi-structured, and even raw data types.



- Data transformation is still necessary before analyzing the data with a business intelligence platform. However, data cleansing, enrichment, and transformation occur after loading the data into the data lake.
- The three operations happening in ELT
    - Extract
    - Load
    - Transform

- **Extraction**
    - Extracting data is the technique of identifying data from one or more sources. The sources may be databases, files, ERP, CRM or any other useful source of data.
- **Load**
    - Loading is the process of storing the extracted raw data in data warehouse or data lakes.
- **Transform**
    - Data transformation is the process in which the raw data source is transformed to the target format required for analysis.

Advantages of ELT

- High speed:
    - ELT allows for all of the data to go into the system immediately, and from there, users can determine the exact data they need to both transform and analyze.
- Flexibility:
    - Analysts no longer have to determine what insights and data types they need in advance but can perform transformations on the data as needed in the warehouse.
- Lower Cost:
    - Requires a less-powerful server for data transformation and takes advantage of resources already in the warehouse. This results in cost savings and resource efficiencies.

Disadvantages of ELT

- Security gaps:
    - Storing all the data and making it accessible to various users and applications come with security risks. Companies must take steps to

ensure their target systems are secure by properly masking and encrypting data.

- Increased latency:
    - The need to continually transform data slows down the overall time it takes to perform queries/analysis.

Difference between ETL and ELT

Difference Between ETL and ELT

Parameters	ETL	ELT
Process	Data is transformed at staging server and then transferred to Datawarehouse DB.	Data remains in the DB of the Datawarehouse.
Source Data	Support storing structured data from input sources	Can be used for structured, unstructured, and semi-structured data types
Time-Load	Data first loaded into staging and later loaded into target system. Time intensive.	Data loaded into target system only once. Faster.
Flexibility	Low, as data sources and transformations need to be defined at the beginning of the process	High, as transformation need not be defined when integrating new sources
Scalability	Can be low, as the ETL tool should support scaling of operations	High, as ELT tools can be easily configured for changing data sources
Support for Data warehouse	ETL model used for on-premises, relational and structured data.	Used in scalable cloud infrastructure which supports structured, unstructured data sources.
Data Lake Support	Does not support.	Allows use of Data lake with unstructured data.
Cost	High costs for small and medium businesses.	Low entry costs using online Software as a Service Platforms.
Compliance with Security Protocols	Easy to Implement	May need to be supported by data warehouse/ELT tool
Maturity	The process is used for over two decades. It is well documented and best practices easily available.	Relatively new concept and complex to implement.
Storage Type	Can be used for on-premises or cloud storage	Optimized for cloud data warehouses

Types Of ETL Tools

- Enterprise Software ETL Tools
- Open Source ETL Tools
- Cloud- Based ETL Tools
- Custom ETL Tools

Cloud Based ETL Tools

- AWS - AWS Glue
- Azure - Azure Data Factory
- GCP - Dataflow and Dataproc

Enterprise Software ETL Tools

- Informatica Cloud Data Integration for Cloud ETL and ELT
- Talend Data Integration
- Mulesoft anypoint platform

OpenSource ETL Tools

- Apache Camel
- Apache Kafka
- Apache Nifi

ETL Use cases

- Data warehousing and IoT data integration
- Machine learning and artificial intelligence
- Cloud migration