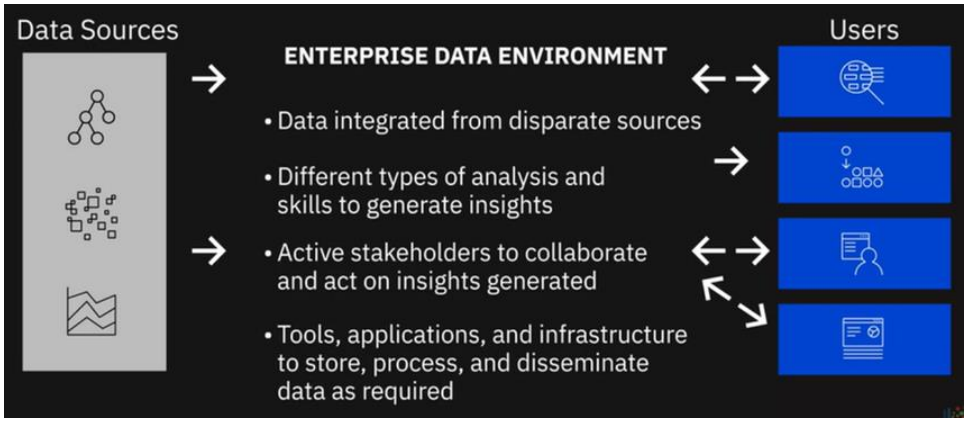
1. Introduction to Data Engineering

1.1. Modern Data Ecosystem

Modern data ecosystem includes a network of interconnected and continually evolving entities that include:

- **Data**, that is available in a host of different formats, structures, and sources.
- **Enterprise Data Environment**, in which raw data is staged so it can be organized, cleaned, and optimized for use by end-users.
- **End-users**, such as business stakeholders, analysts, and programmers who consume data for various purposes.



Emerging technologies such as Cloud Computing, Machine Learning, and Big Data, are continually reshaping the data ecosystem and the possibilities it offers.

Data Engineers, Data Analysts, Data Scientists, Business Analysts, and Business Intelligence Analysts, all play a vital role in the ecosystem for deriving insights and business results from data.

1.2. Responsibilities and Skillsets of a Data Engineer

1.2.1. Responsibility

The goal of Data Engineering is to make quality data available for analytics and decision-making. And it does this by collecting raw source data, processing data so it becomes usable, storing data, and making quality data available to users securely.

The role of a Data Engineer includes:

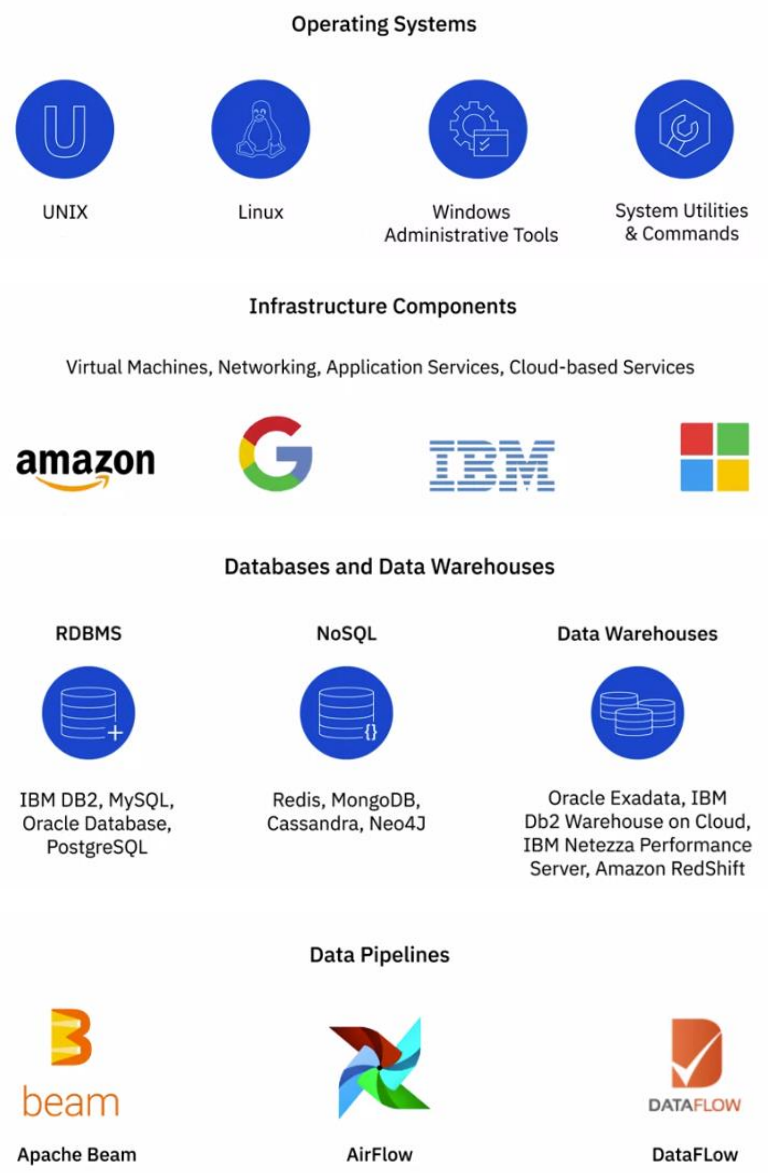
- Gathering data from disparate sources.

- Integrating data into a unified view for data consumers.
- Preparing data for analytics and reporting.
- Managing data pipelines for a continuous flow of data from source to destination systems.
- Managing the complete infrastructure for the collection, processing, and storage of data.

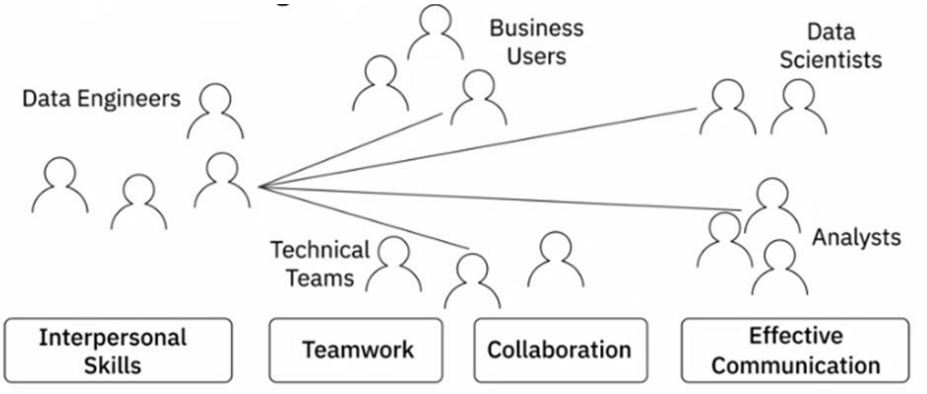
1.2.2. Skillsets of a Data Engineer

To be successful in their role, Data Engineers need a mix of technical, functional, and soft skills.

- **Technical Skills** include working with different operating systems and infrastructure components such as virtual machines, networks, and application services. It also includes working with databases and data warehouses, data pipelines, ETL tools, big data processing tools, and languages for querying, manipulating, and processing data.



- **Functional skills** include the ability to convert business requirements into technical specifications, an understanding of the software development lifecycle, and the areas of data quality, privacy, security, and governance.
- **Soft Skills** include interpersonal skills, the ability to work collaboratively, teamwork, and effective communication.



2. The Data Engineering Ecosystem

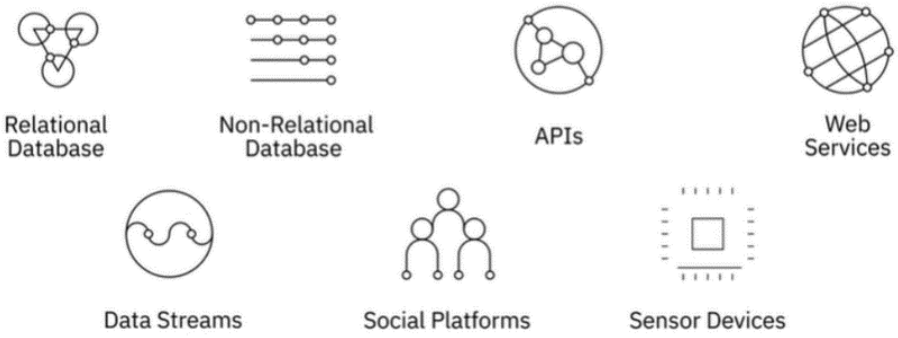
2.1. Data Ecosystem and Languages for Data Professional

2.1.1. Data Ecosystem

A Data Engineer's ecosystem includes the infrastructure, tools, frameworks, and processes for:

- **Extracting data from disparate sources**

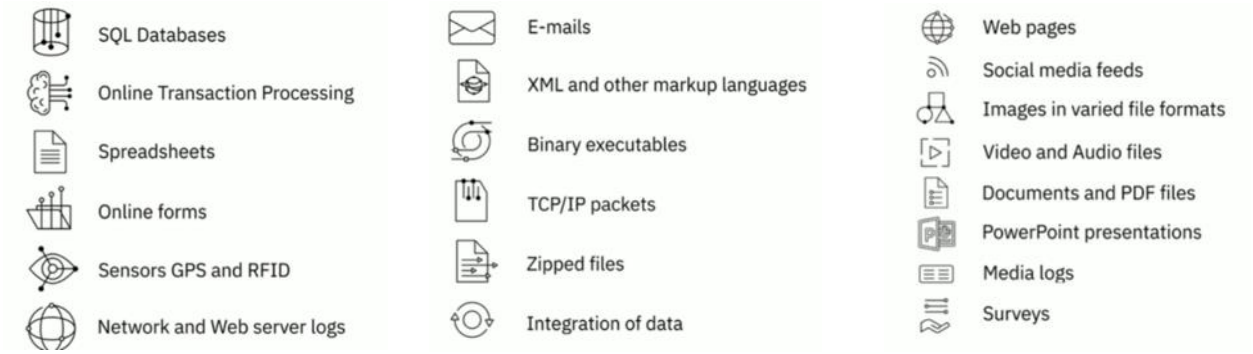
Data is extracted from multiple data sources, ranging from relational and non-relational databases, to APIs, web services, data streams, social platforms, and sensor devices.



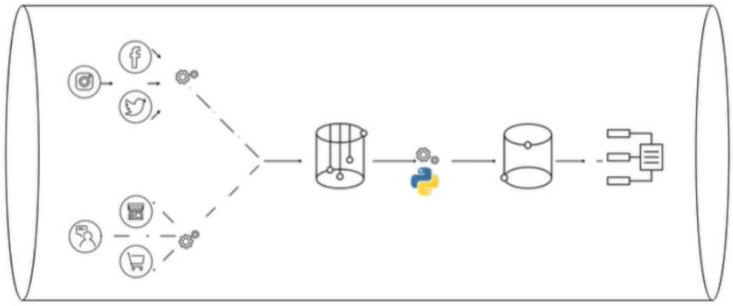
Data comes in a wide-ranging variety of file formats, such as, delimited text files, spreadsheets, XML, PDF, and JSON, each with its own list of benefits and limitations of use.

Based on how well-defined the structure of the data is, data can be categorized as

Structured data	**Semi-structured data**	**Unstructured data**
Data which is well organized in formats that can be stored in databases.	Data which is partially organized and partially free form.	Data which cannot be organized conventionally into rows and columns.



- **Architecting and managing data pipelines for transformation, integration, and storage of data**



A set of tools and processes that cover the entire journey of data from source to destination systems.

- **Architecting and managing data repositories**

Once the data is identified and gathered from different sources, it needs to be staged in a data repository so that it can be prepared for analysis. There are two main types of data repositories:

  - **Transactional systems**, also known as **Online Transaction Processing (or OLTP)** systems, are designed to store high-volume day-to-day operational data. Such as online banking transactions, ATM transactions, and airline bookings. While OLTP databases are typically relational, they can also be non-relational.
  - **Analytical systems**, also known as **Online Analytical Processing (OLAP)** systems, are optimized for conducting complex data analytics. These include relational and non-relational databases, data warehouses, data marts, data lakes, and big data stores.

The type, format, and sources of data influence the type of data repository that can be used.

- **Automating and optimizing workflows and flow of data between systems; and**
- **Developing applications needed through the data engineering workflow.**

2.1.2. Languages for Data Professional

Data professionals need a host of languages that can help them extract, prepare, and analyze data. These can be classified as:

- **Querying languages**, such as SQL, are used for accessing and manipulating data from databases.
- **Programming languages** such as Python, R, and Java, for developing applications and controlling application behavior.
- **Shell** and **Scripting languages**, such as Unix/Linux Shell, and PowerShell, for automating repetitive operational tasks.

2.2. Data Repositories, Data Pipelines, and Data Integration Platforms

2.2.1. Data Repositories

A Data Repository is a general term that refers to data that has been collected, organized, and isolated so that it can be used for reporting, analytics, and for archival purposes.

The different types of Data Repositories include:

- **Databases**, which can be relational or non-relational, each follow a set of organizational principles, the types of data they can store, and the tools that can be used to query, organize, and retrieve data. Factors governing choice of database include:
    - Data type
    - Data structure
    - Querying mechanisms
    - Latency requirements
    - Transaction speeds
    - Intended use of data

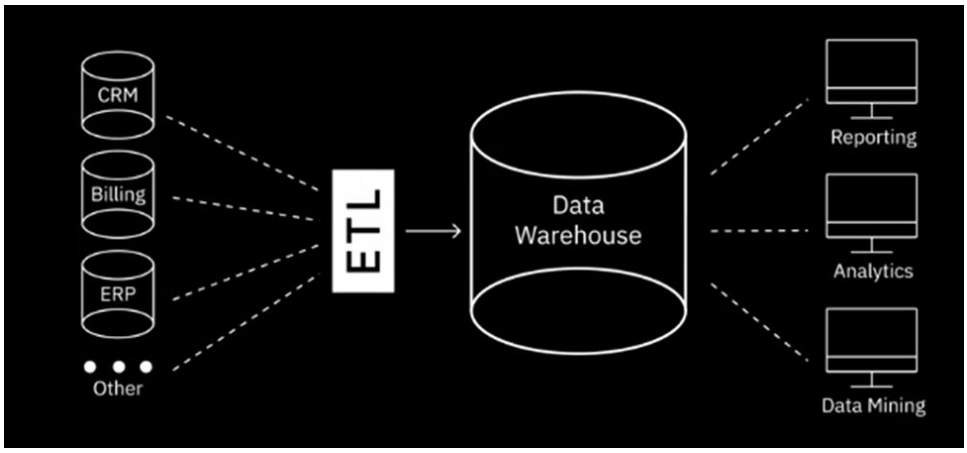
    i. **Relational database** also referred to as **RDBMS**, built on the organizational principles of flat files, with data organized into a tabular format with rows and columns following a well-defined structure and schema.

    ii. **Non-relational databases**, also known as **NoSQL**, or "Not Only SQL". Non-relational databases emerged in response to the volume, diversity, and speed at which data is being generated today, mainly influenced by advances in cloud computing, the Internet of Things, and social media proliferation. Built for speed, flexibility, and scale, non-relational databases made it possible to store data in a schema-less or free-form fashion.

    NoSQL is widely used for processing big data. Based on the model being used for storing data, there are four common types of NoSQL databases:

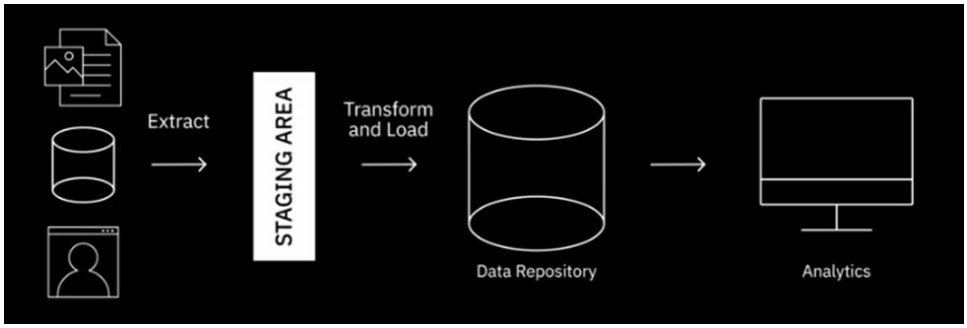
    - Key-value store
    - Document Based
    - Column Based
    - Graph Based

- **Data Warehouses** work as a central repository that merges information coming from disparate sources and consolidates it through the extract, transform, and load process, also known as the ETL process, into one comprehensive database for analytics and business intelligence.

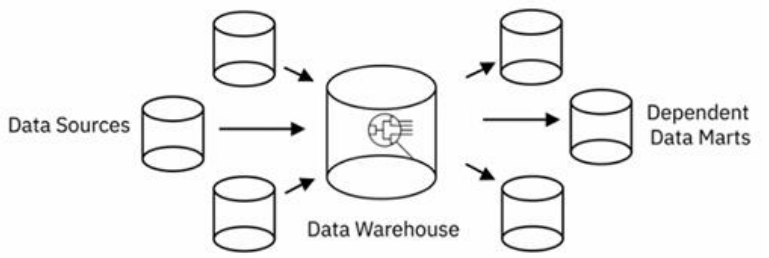


The ETL, or Extract Transform and Load, Process is an automated process that converts raw data into analysis-ready data by:

- Extracting data from source locations.
- Transforming raw data by cleaning, enriching, standardizing, and validating it.
- Loading the processed data into a destination system or data repository.

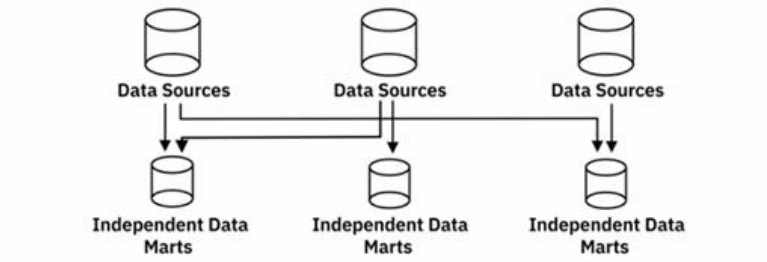


- **Data Marts**, that are essentially sub-sections of a data warehouse, built to isolate data for a particular business function or use case.
  There are three basic types of data marts:
  - Dependent marts



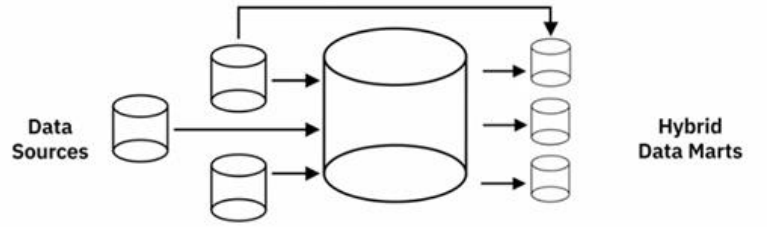
Dependent Data Marts offer analytical capabilities for a restricted area of a Data Warehouse.

  - Independent marts



Independent Data Marts are created from sources other than an Enterprise Data Warehouse, such as Internal Operational Systems or External Data.

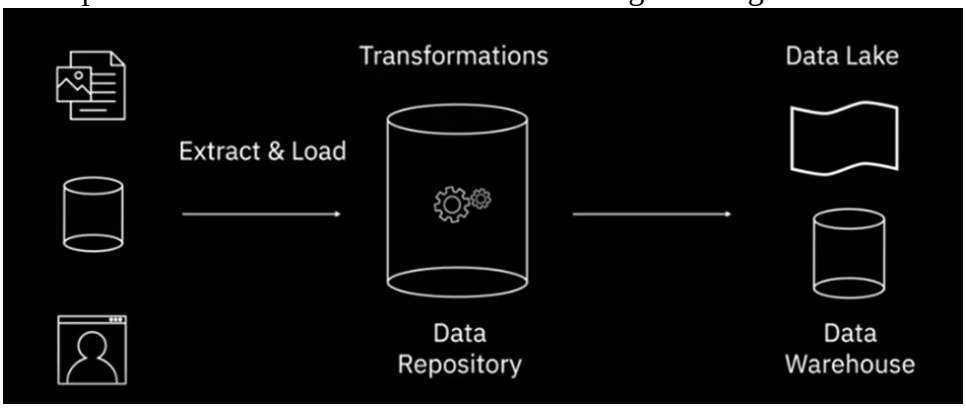
- Hybrid data marts



Hybrid Data Marts combine inputs from Data Warehouses, Operational Systems, and External Systems.

- **Data Lakes**, that serve as storage repositories for large amounts of structured, semi-structured, and unstructured data in their native format.

  The **ELT**, or Extract Load and Transfer, Process is a variation of the ETL Process. In this process, extracted data is loaded into the target system before the transformations are applied. This process is ideal for Data Lakes and working with Big Data.



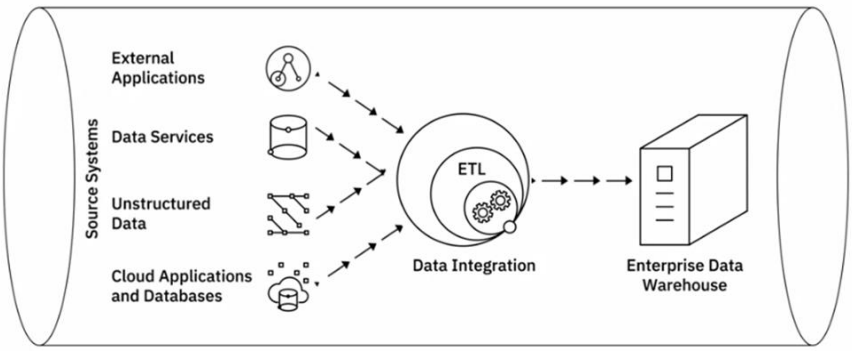
- **Big Data Stores**, that provide distributed computational and storage infrastructure to store, scale, and process very large data sets.

2.2.2. Data Pipelines

Data Pipeline, sometimes used interchangeably with ETL and ELT, encompasses the entire journey of moving data from its source to a destination data lake or application, using the ETL or ELT process.

2.2.3. Data Integration

Data Integration Platforms combine disparate sources of data, physically or logically, to provide a unified view of the data for analytics purposes.



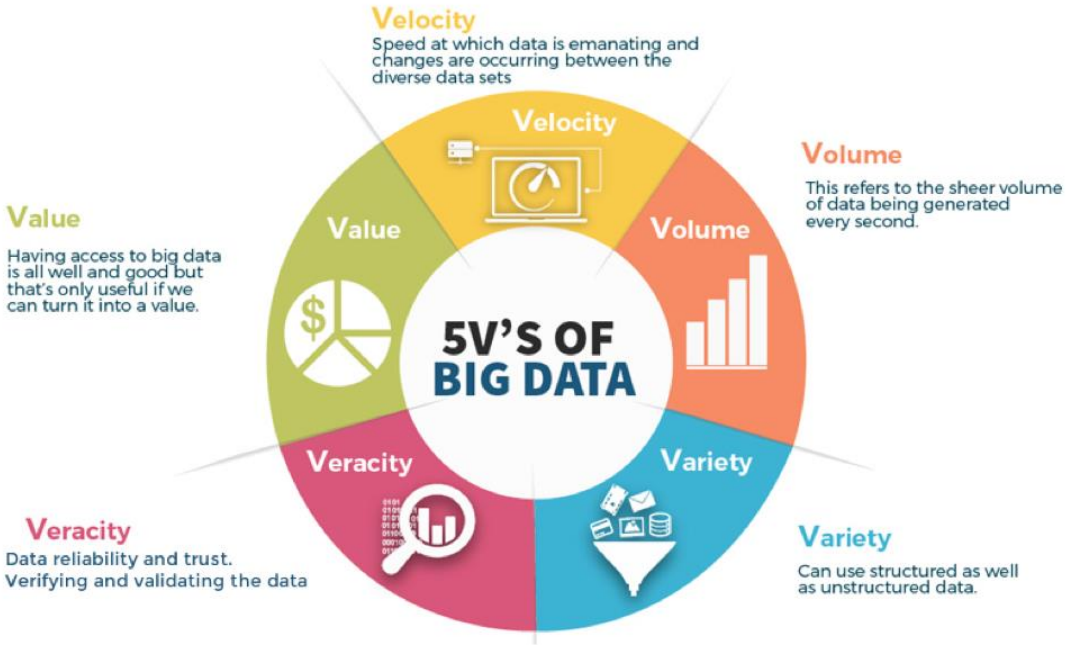
Data integration has several usage scenarios, such as data consistency across applications, master data management, data sharing between enterprises, and data migration and consolidation.

In the field of analytics and data science, data integration includes:

- Accessing, queueing, or extracting data from operational systems,
- Transforming and merging extracted data either logically or physically,
- Data quality and governance,
- Delivering data through an integrated approach for analytics purposes.

2.3. Big Data Platforms

Big Data refers to the vast amounts of data that is being produced each moment of every day, by people, tools, and machines. The sheer velocity, volume, and variety of data challenged the tools and systems used for conventional data, leading to the emergence of processing tools and platforms designed specifically for Big Data.



Big Data processing technologies help derive value from big data. These include NoSQL databases and Data Lakes and open-source technologies such as Apache Hadoop, Apache Hive, and Apache Spark.

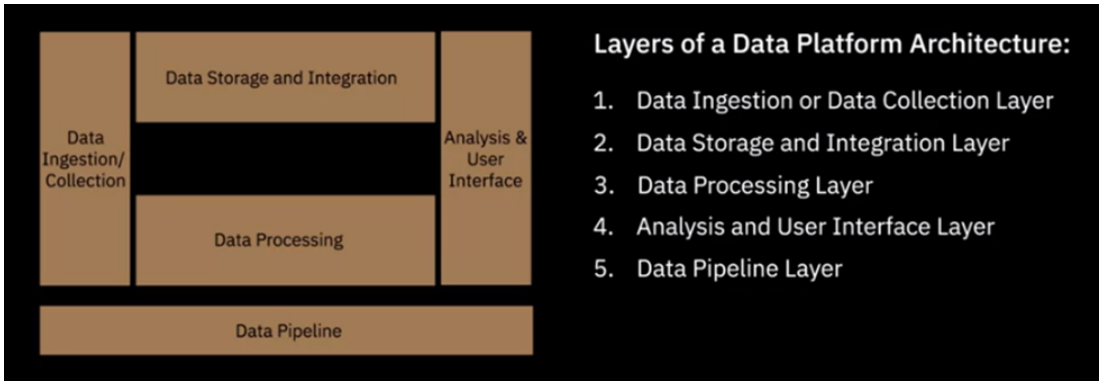
- Hadoop provides distributed storage and processing of large datasets across clusters of computers. One of its main components, the Hadoop File Distribution System, or HDFS, is a storage system for big data.
- Hive is a data warehouse software for reading, writing, and managing large datasets.
- Spark is a general-purpose data processing engine designed to extract and process large volumes of data.

3. Data Engineering Lifecycle

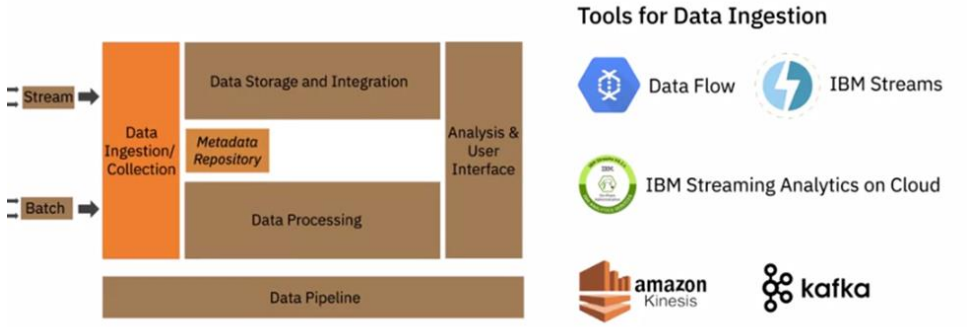
3.1. Data Platforms, Data Stores, and Security

3.1.1. Architecting the Data Platform

The architecture of a data platform can be seen as a set of layers, or functional components, each one performing a set of specific tasks. These layers include:



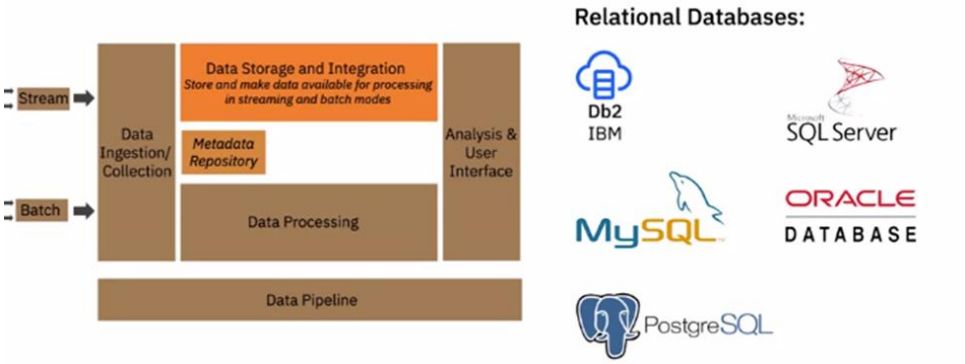
- **Data Ingestion or Data Collection Layer**, responsible for bringing data from source systems into the data platform. This layer performs the following key tasks:
  - Connect to data sources.
  - Transfer data from these data sources to the data platform in streaming, batch, or both modes.
  - Maintain information about the data collected in the metadata repository.



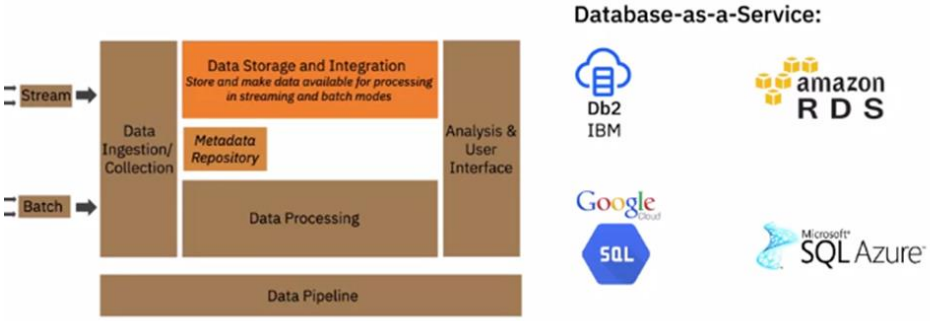
- **Data Storage and Integration Layer**, responsible for storing and merging extracted data. The Storage and Integration layer in a data platform needs to:
  - Store data for processing and long-term use.
  - Transform and merge extracted data, either logically or physically.
  - Make data available for processing in both streaming and batch modes.

  The storage layer needs to be **reliable**, **scalable**, **high-performing**, and **cost-efficient**.

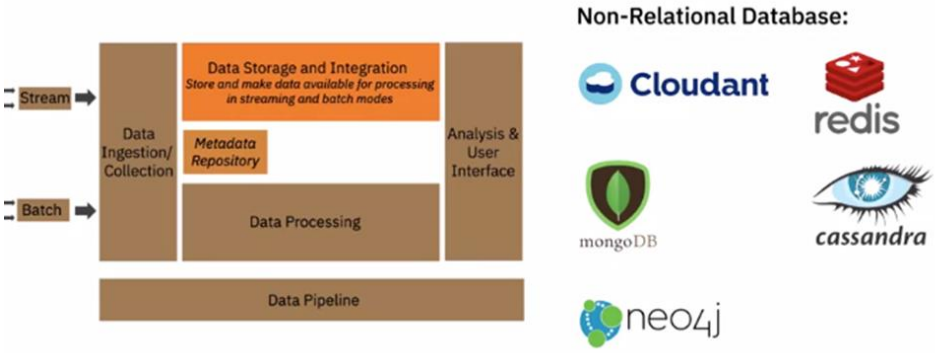
  Some of the popular relational databases.



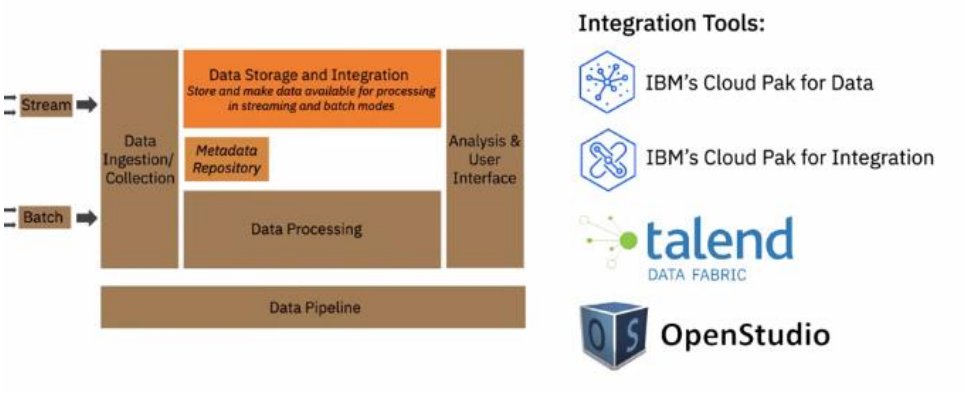
Cloud-based relational databases, also referred to as Database-as-a-Service, have gained great popularity over recent years.



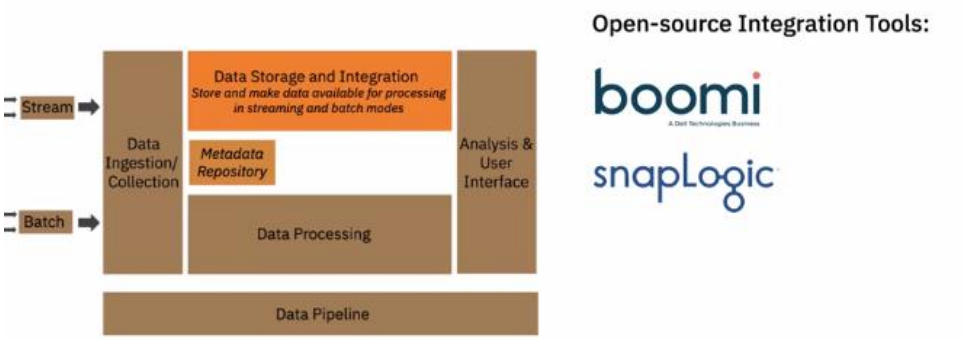
In the NoSQL, or non-relational database systems on the cloud, we have IBM Cloudant, Redis, MongoDB, Cassandra, and Neo4J.



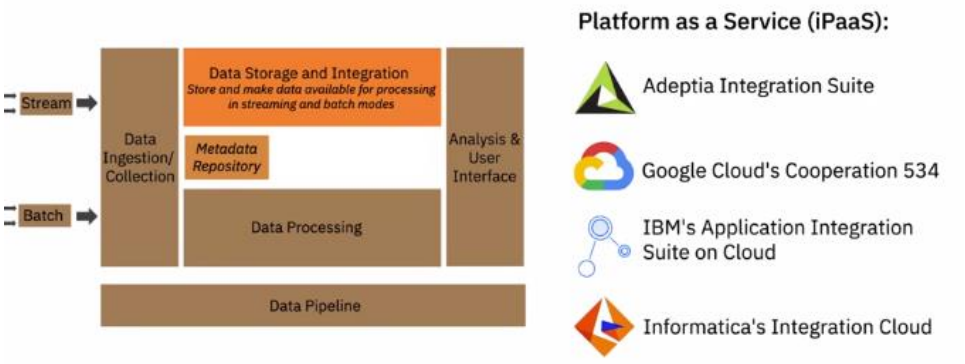
Tools for integration include IBM's Cloud Pak for Data and Cloud Pak for Integration; Talend's Data Fabric and Open Studio.



Open-source tools such as Dell Boomi and SnapLogic are also very popular integration tools.



There are several vendors offering cloud-based Integration Platform as a Service (or iPaaS). For example, Adeptia Integration Suite, Google Cloud's Cooperation 534, IBM's Application Integration Suite on Cloud, and Informatica's Integration Cloud.

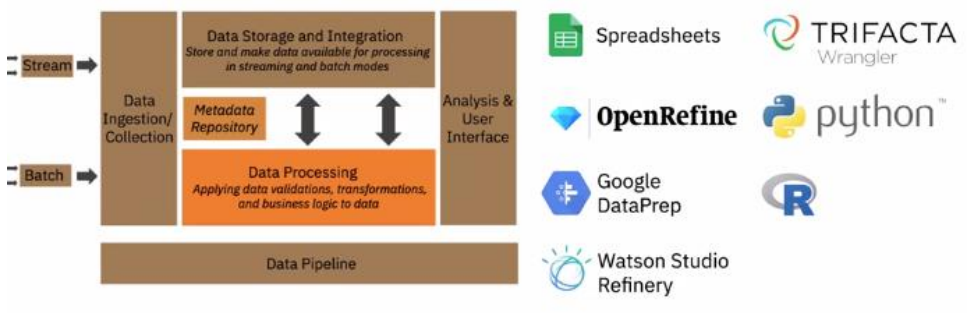


- **Data Processing Layer**, responsible for validating, transforming, and applying business rules to data. The processing layer should be able to:
  - Read data in batch or streaming modes from storage and apply transformations.
  - Support popular querying tools and programming languages.
  - Scale to meet the processing demands of a growing dataset.
  - Provide a way for analysts and data scientists to work with data in the data platform.

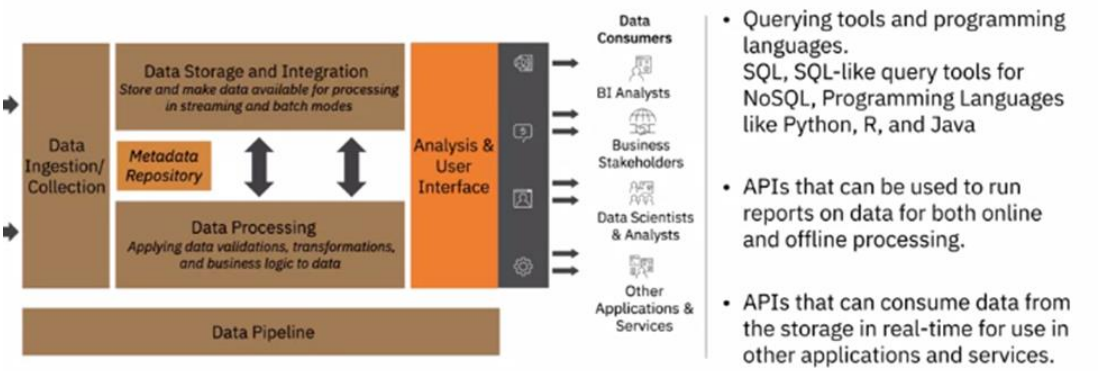
Some of the transformation tasks that occur in this layer include:

- **Structuring**, essentially - actions that change the form and schema of the data. This change may be as simple as changing the order of fields within a record or dataset or as complex as combining fields into complex structures using joins and unions.
- **Normalization** - which focuses on cleaning the database of unused data and reducing redundancy and inconsistency.
- **Denormalization** - which combines data from multiple tables into a single table so that it can be queried more efficiently for reporting and analysis.
- **Data Cleaning** - which fixes irregularities in data to provide credible data for downstream applications and uses.

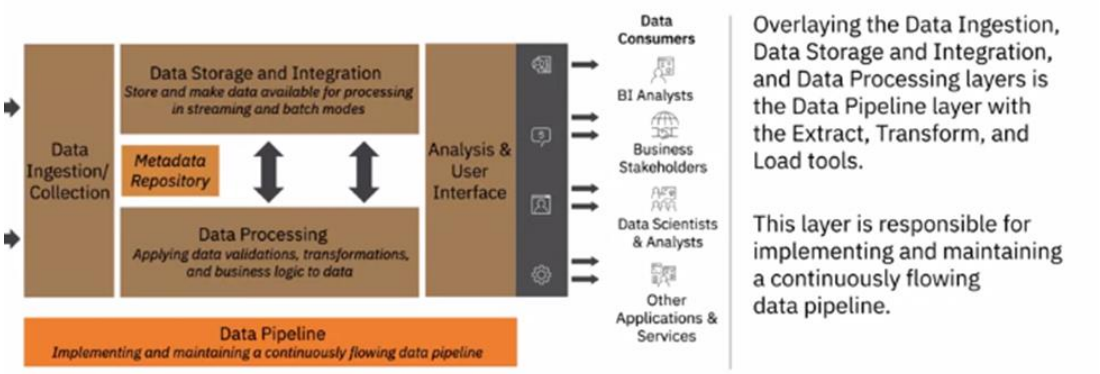
There are a host of tools available for performing these transformations on data, selected based on the data size, structure, and specific capabilities of the tool.



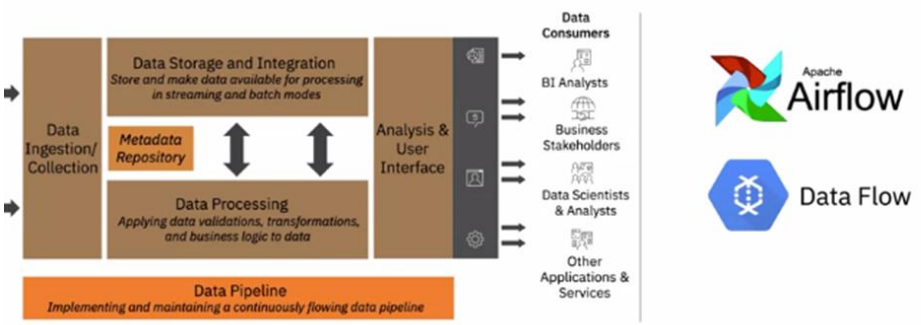
- **Analysis and User Interface Layer**, responsible for delivering processed data to data consumers.



- **Data Pipeline Layer**, responsible for implementing and maintaining a continuously flowing data pipeline.

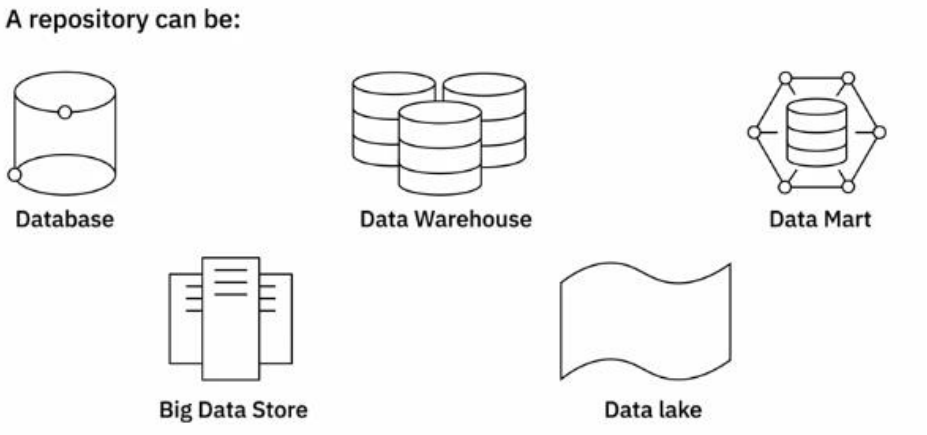


There are several data pipeline solutions available, the most popular among them being Apache Airflow and DataFlow.



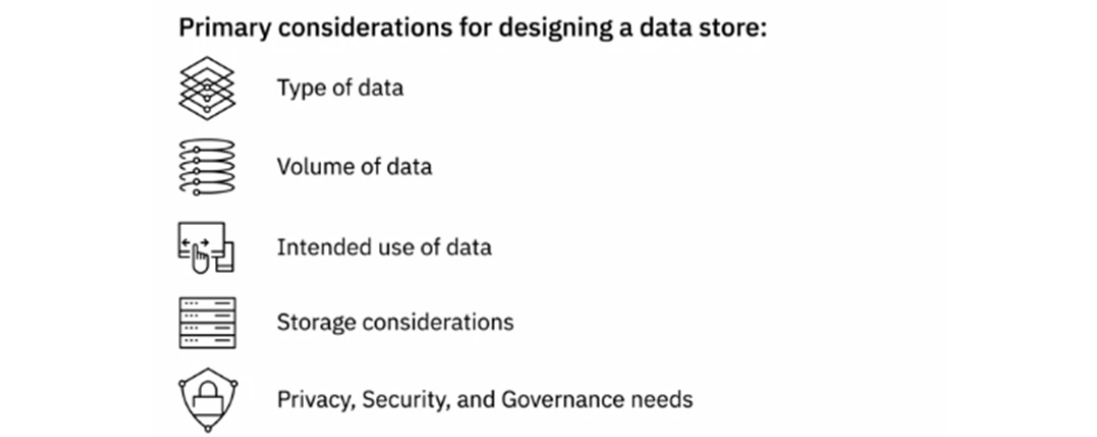
3.1.2. Factors for selecting and designing Data Stores

A repository can be a database, data warehouse, data mart, big data store, or a data lake.



A well-designed data repository is essential for building a system that is scalable and capable of performing during high workloads.

The choice or design of a data store is influenced by the type and volume of data that needs to be stored, the intended use of data, and storage considerations. The privacy, security, and governance need of your organization also influence this choice.



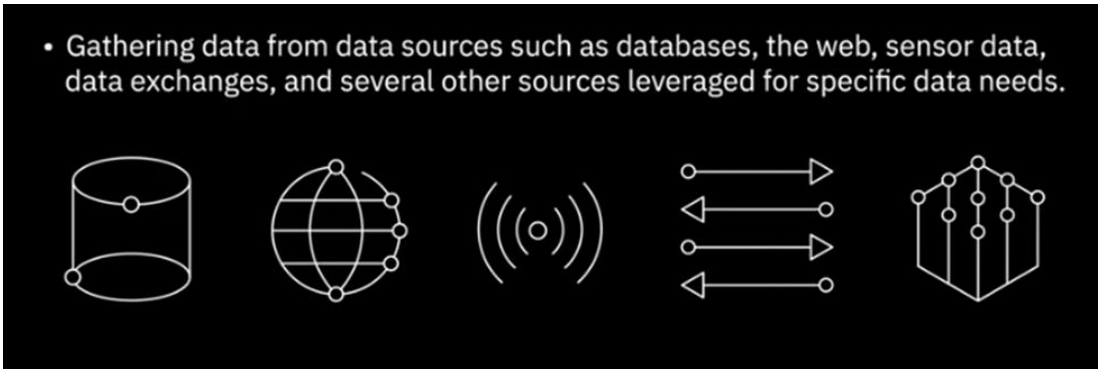
3.1.3. Security

The CIA, or Confidentiality, Integrity, and Availability triad are three key components of an effective strategy for information security. The CIA triad is applicable to all facets of security, be it infrastructure, network, application, or data security.

3.2. Data Collection and Data Wrangling

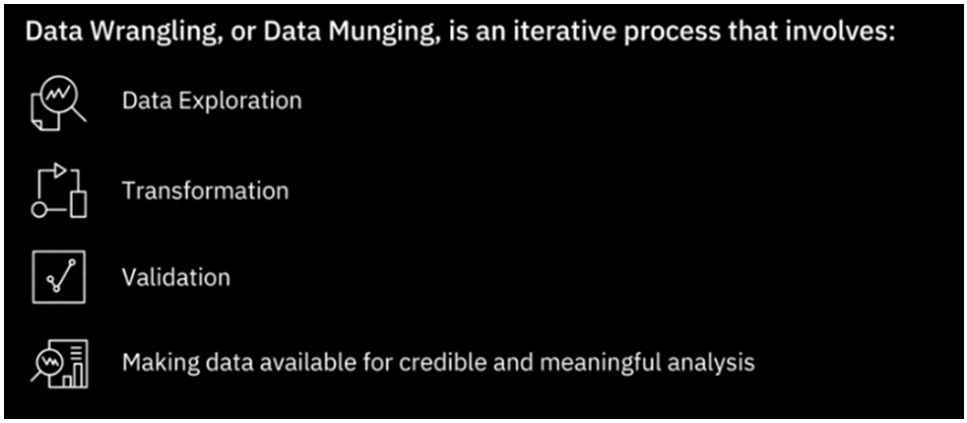
3.2.1. Data Collection

Depending on where the data must be sourced from, there are several methods and tools available for gathering data. These include query languages for extracting data from databases, APIs, Web Scraping, Data Streams, RSS Feeds, and Data Exchanges.



3.2.2. Data Wrangling

Once the data you need has been gathered and imported, your next step is to make it analytics ready. This is where the process of Data Wrangling, or Data Munging, comes in.



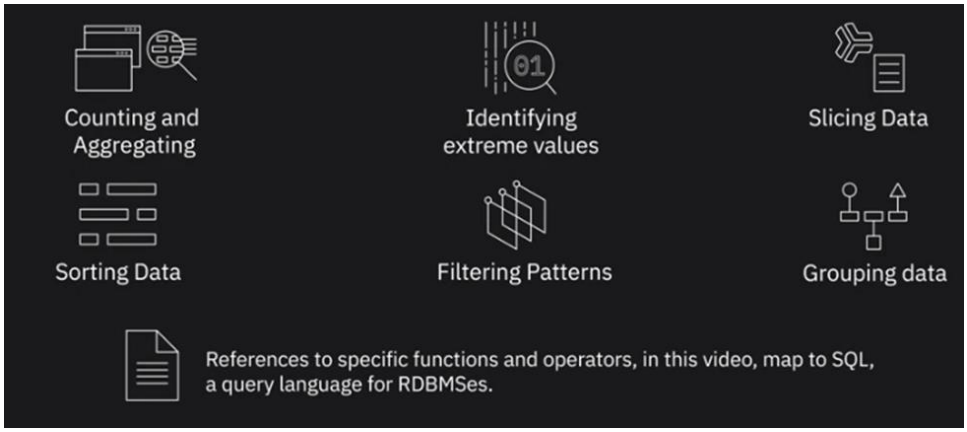
Data Wrangling involves a whole range of transformations and cleansing activities performed on the data.

- **Transformation** of raw data includes the tasks you undertake to:
  - Structurally manipulate and combine data using Joins and Unions.
  - Normalize data, that is, clean the database of unused and redundant data.
  - Denormalize data, that is, combine data from multiple tables into a single table so that it can be queried faster.
- **Cleansing** activities include:
  - Profiling data to uncover anomalies and quality issues.
  - Visualizing data using statistical methods in order to spot outliers.
  - Fixing issues such as missing values, duplicate data, irrelevant data, inconsistent formats, syntax errors, and outliers.

A variety of software and tools are available for the data wrangling process. Some of the popularly used ones include Excel Power Query, Spreadsheets, OpenRefine, Google DataPrep, Watson Studio Refinery, Trifacta Wrangler, Python, and R, each with their own set of features, strengths, limitations, and applications.

3.3. Querying Data, Performance Tuning, and Troubleshooting

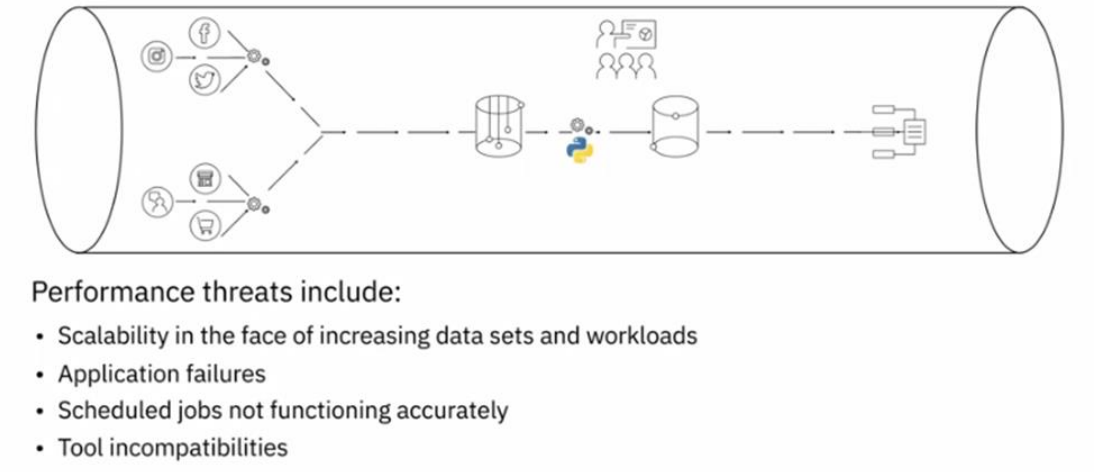
For raw data to become analytics-ready, several transformation and cleansing tasks need to be performed on raw data. And that requires you to understand your dataset from multiple perspectives. One of the ways in which you can explore your dataset is to query it. Basic querying techniques can help you explore your data, such as counting and aggregating a dataset, identifying extreme values, slicing data, sorting data, filtering patterns, and grouping data.



In a data engineering lifecycle, the performance of data pipelines, platforms, databases, applications, tools, queries, and scheduled jobs, need to be constantly monitored for **performance** and **availability**.

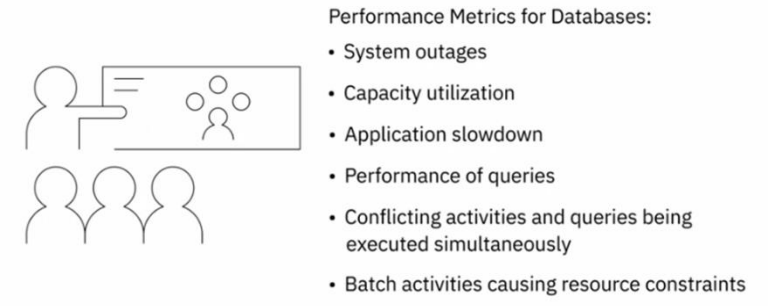
- **Data Pipelines – Performance Threats**

The performance of a data pipeline can be impacted if the workload increases significantly, or there are application failures, or a scheduled job does not work as expected, or some of the tools in the pipeline run into compatibility issues.



- **Performance Metrics for Databases**

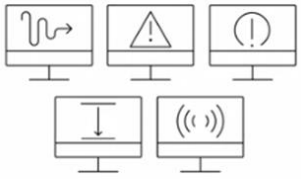
Databases are susceptible to outages, capacity overutilization, application slowdown, and conflicting activities and queries being executed simultaneously.



- **Monitoring**

Monitoring and alerting systems collect quantitative data in real time to give visibility into the performance of data pipelines, platforms, databases, applications, tools, queries, scheduled jobs, and more.

- **Maintenance routines**



Preventive maintenance routines generate data that we can use to identify systems and procedures responsible for faults and low availability.

These routines can be:

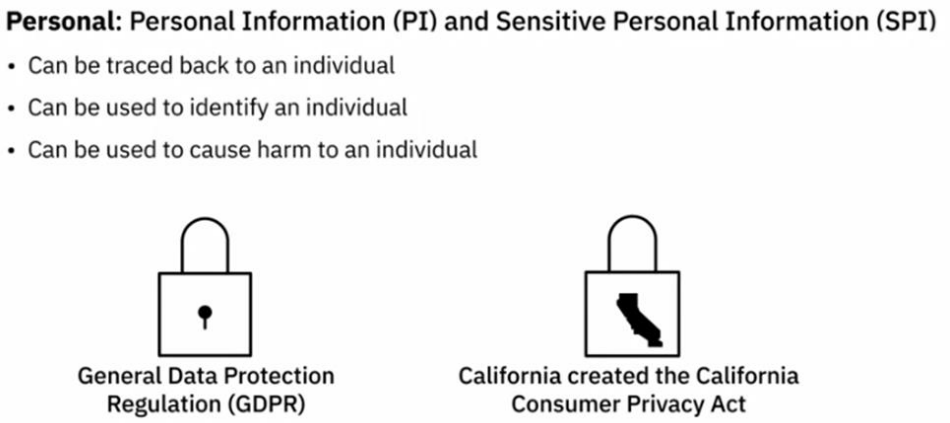
- Time-based – Planned as scheduled activities at pre-fixed time intervals.
- Condition-based – Performed when there is a specific issue or a decrease in performance.

3.4. Governance and Compliance

3.4.1. Data Governance

Data governance is a collection of principles, practices, and processes that help maintain the security, privacy, and integrity of data through its lifecycle. Personal Information and Sensitive Personal Information, that is, data that can be traced back to an individual or can be used to identify or cause harm to an individual, needs to be protected through governance regulations.

General Data Protection Regulation, or GDPR, is one such regulation that protects the personal data and privacy of EU citizens for transactions that occur within EU member states. Regulations, such as HIPAA (Health Insurance Portability and Accountability Act) for Healthcare, PCI DSS (Payment Card Industry Data Security Standard) for retail, and SOX (Sarbanes Oxley) for financial data are some of the industry-specific regulations.



3.4.2. Compliance

Compliance covers the processes and procedures through which an organization adheres to regulations and conducts its operations in a legal and ethical manner. Compliance requires organizations to maintain an auditable trail of personal data through its lifecycle, which includes acquisition, processing, storage, sharing, retention, and disposal of data.

Tools and technologies play a critical role in the implementation of a governance framework, offering features such as:

- Authentication and Access Control.
- Encryption and Data Masking.
- Hosting options that comply with requirements and restrictions for international data transfers.
- Monitoring and Alerting functionalities.
- Data erasure tools that ensure deleted data cannot be retrieved.