Big Data Course

Udemy– Course –1

5 V's of Big Data

The 5 V's of Big Data provides a framework for understanding the challenges and characteristic of Big Data.

1. Volume : size of data getting generated

   myth : Volume is not the only factor & no specified limit

2. Velocity : speed at which data is getting generated and needs to be processed.

Real Time	Near real time	Batch
Credit card / Bank Transaction Live feed	data is coming continously but we are processing in say 2 min / 5 min	Credit card Bill

3. Variety : data can be in different formats & we have to deal with all of them.

Structured — rows and column

Structured data

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Semistructured — json, xml, csv

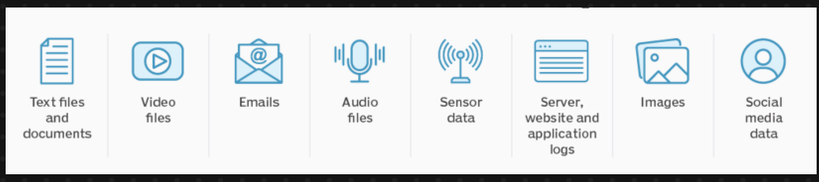
```
{
  "customer": "John Doe",
  "gender": "Male",
  "items": {
     "SSN": "123-45-6789",
     "exp": "10/01/2025",
     "moresubnesting": {
        "SSN": "123-45-6789",
        "newfield2": "123 Main Street"
     }
  }
}
```

json

```
<University>
  <Student ID="1">
    <Name>John</Name>
    <Age>18</Age>
    <Degree>B.Sc.</Degree>
  </Student>
  <Student ID="2">
    <Name>David</Name>
    <Age>31</Age>
    <Degree>Ph.D. </Degree>
  </Student>
  ....
</University>
```

xml

Unstructured



4. Veracity : Trustworthiness or Quality of data
   - → Age is -ve
   - → messy

5. Value : data should have / provide value or insights. helpful in making business decision.

Big Data and Distributed System

gta ⟶ gta5

we need more resources

1. Storage
2. Memory (RAM)
3. Processor

Single system

Monolythic systems

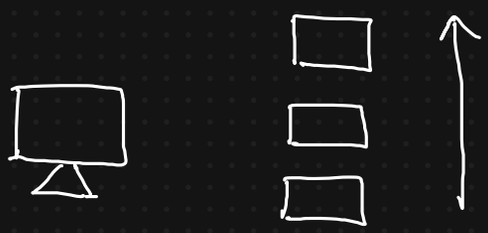
mono → 1

monolythic system has a hardware limitation

X ⟶ 2X ⟶ 10X

P ⟶ 2P ⟶ 6P

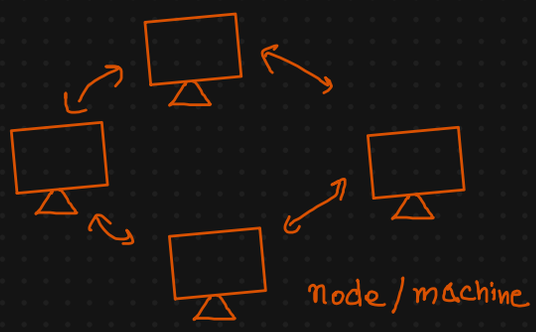
Scalability problem



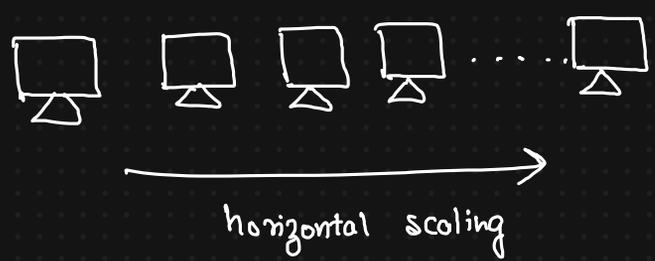
Vertical scaling

Multiple systems

Distributed systems



We don't have any limitation wrt scaling



Big Data and Distributed System

gta 3 gta 5

When the complexity or size of our data increases, we will be needing more resources.

Resources

1. Storage → hdd, ssd
2. Memory → RAM
3. Performance → cores

Single system
Monolithic System

increase the resource

mono → 1

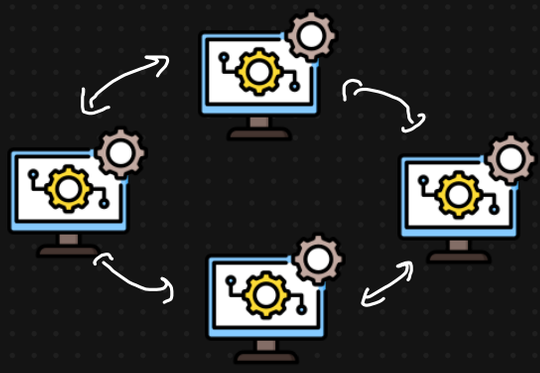
X → 2X → 10X

P → 2P → 6P

hardware limitation

they are not scalable

Multiple systems
Distributed systems



group of machines → Cluster

We can add more m/c here as & when required.

True Scaling

X → 2X → 10X

P → 2P → 10P

All good big data systems are based on distributed systems.

Designing a good Big Data Systems

a good big data system is the one designed to handle large scale data efficiently, ensuring scalability, reliability and fault tolerance.

1. Scalability: amazon sale

   a) Storage scalability: increased data can be stored

   b) Processing/computation scability: process the larger datasets faster by adding more compute machines.

2. Reliability and fault tolerance: it should continue to work/operate even if some component fails

3. Cost effectiveness: system should balance performance and scalability with cost.

4. Security and Data Privacy: data is saved from unauthorized access.

On Premise vs Cloud

When choosing to deploy a Big Data system, organizations/companies often face a decision between on-premise infra and cloud solutions.

On Premise Infrastructure

high initial investment

Setting up is our responsibility

Buying an office

20 node cluster

 * 20

Cloud Solutions

renting a house / Co-working space

Pay-as you go model

aws → amazon
azure → microsoft
gcp → google

Aspect	On-Premise	Cloud
Deployment	Hardware and software are hosted within the organization's facilities.	Resources and services are hosted on the provider's servers and accessed via the internet.
Cost Model	High upfront costs for hardware, maintenance, and IT staff. Capex↑↑	Pay-as-you-go pricing, with minimal upfront costs. Opex↑
Scalability	Limited by the organization's hardware capacity; scaling requires purchasing and installing new equipment.	Highly scalable—add resources on demand instantly.
Maintenance	The organization is responsible for managing and maintaining hardware, software, and updates.	Managed by the cloud provider (e.g., AWS, Azure, GCP).
Flexibility	Fixed capacity with little room for dynamic needs.	Highly flexible, allowing resources to scale up or down.
Security	Data remains within the organization's premises, offering greater control.	Security is managed by the provider; often meets global compliance standards but might raise concerns for sensitive data.
Disaster Recovery	Requires internal backups and disaster recovery systems.	Cloud providers offer built-in disaster recovery and redundancy.

Types of Cloud

1. Public cloud : aws, azure, gcp
2. Private cloud : Infra set up by single organization for its use
3. Hybrid Cloud : Public + private
4. Community cloud : Shared by organizations with common concern
   eg. University
   Hospitals

Database vs Data Warehouse vs Datalake

Database

system which holds data

organized collection of data.

stored in tables with defined schema

Transaction data → day to day operational data (eg. orders, balance update, inventory)

best for OLTP (online transaction processing)

majorly structure data ⇒ schema will be well defined

```
Employee {
    id
    name
    email
}
```

mobile ✗ error on schema mismatch

* We don't hold years of data but rather the recent one for better performance

overall cost of storing data is high

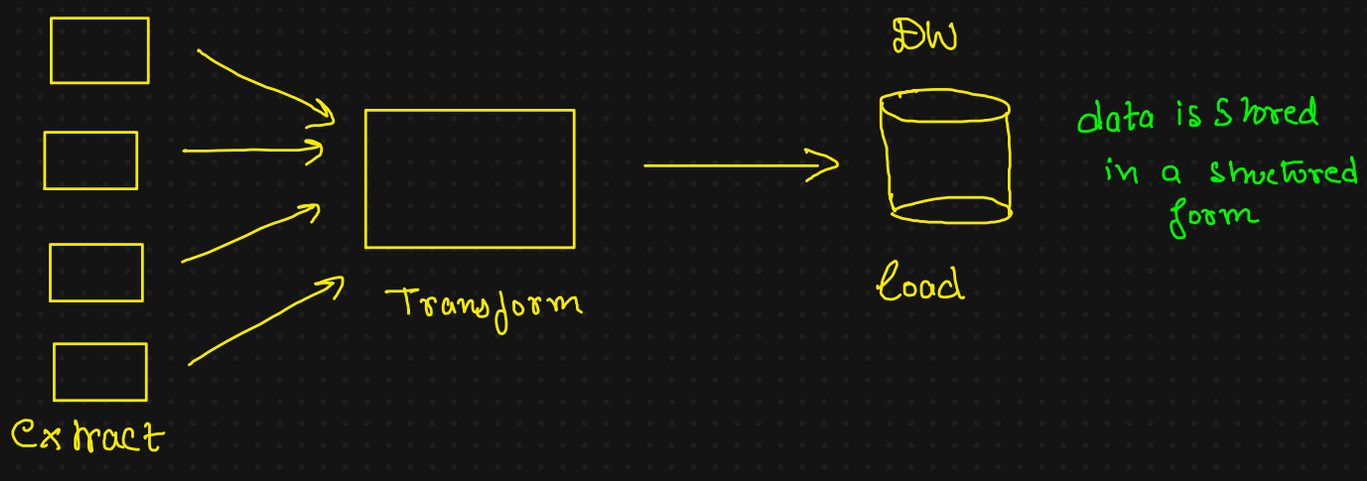
thus they are used for recent data

2. Data Warehouse

eg. Terraform

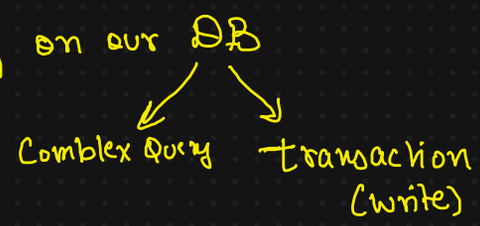
we store a lot more data than a database

ETL



Q: Why can't we use database for analysis?

1] there can be multiple data sources
2. Lots of data i.e. cost
3. we write complex Query on our DB



Performance ↓↓

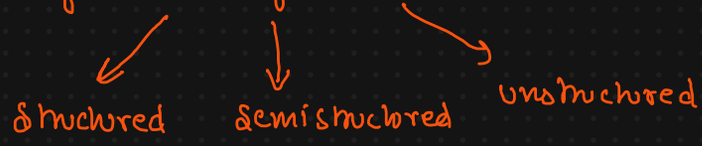
having a storage cost not as high as a database

Read heavy

OLAP → Online Analytical Processing

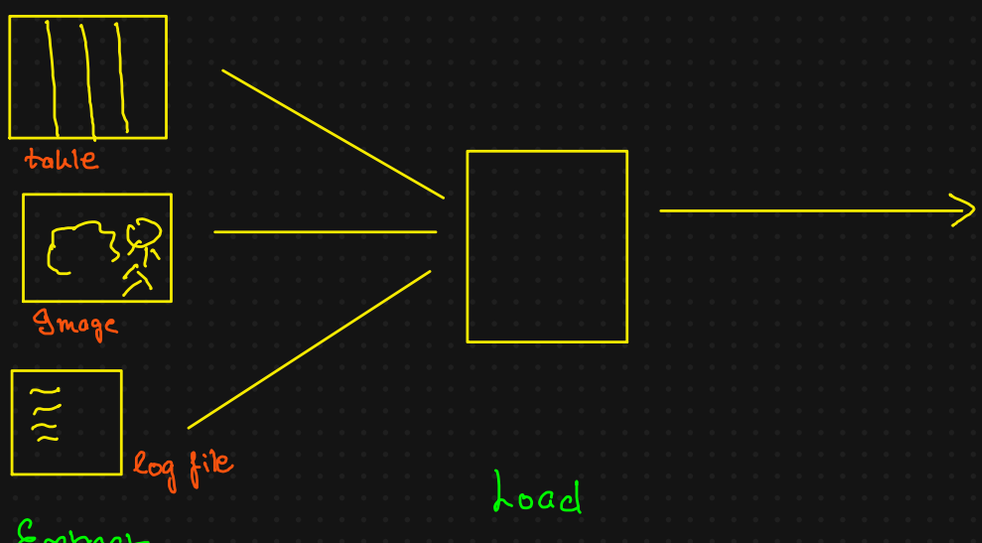
3. Data Lake eg. hdfs, s3

want to get insights from large amount of data.



a data lake is a centralized repository that stores all kind of raw at a scale, without any pre-structuring

E L T



⊕ cost effective
enough flexibility

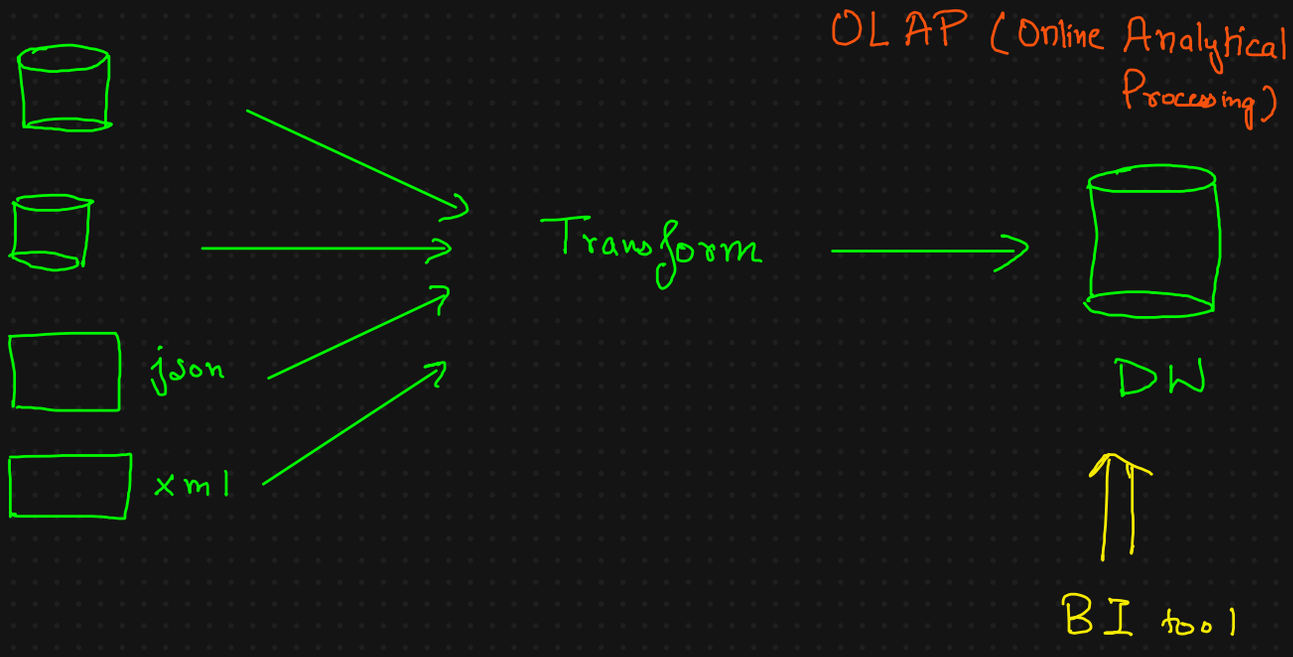
Many team can use the same.

Feature	Database	Data Warehouse	Data Lake
Purpose	Real-time transactions	Historical data analysis	Raw data storage for diverse use cases
Data Structure	Structured	Structured	Structured, Semi-structured, Unstructured
Speed	High-speed for small queries	Optimized for analytical queries	Variable (depends on data processing)
Use Case	Operational systems (e.g., POS)	Business intelligence	Advanced analytics, machine learning
Scalability	Limited	Moderate	Highly scalable

ETL vs ELT

The difference between ELT and ETL are very important to understand as a beginner.

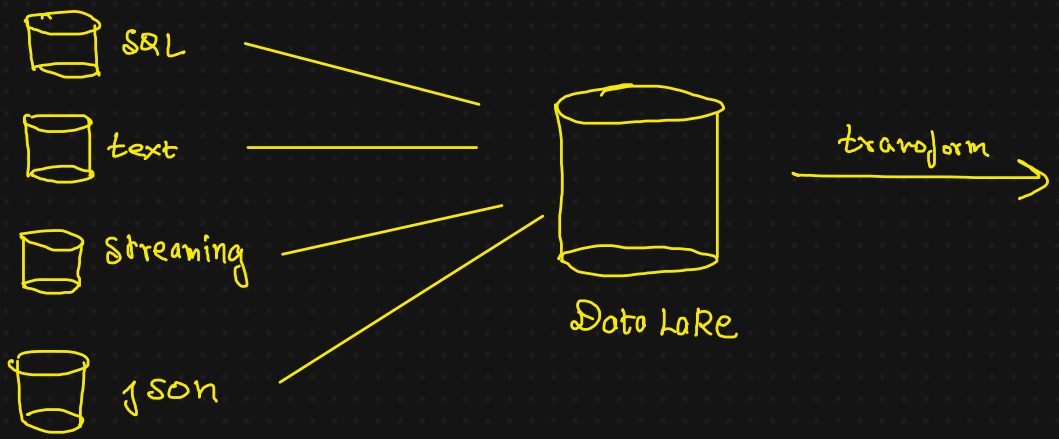
ETL → Extract Transform load



Informatica / Talend / Fivetran

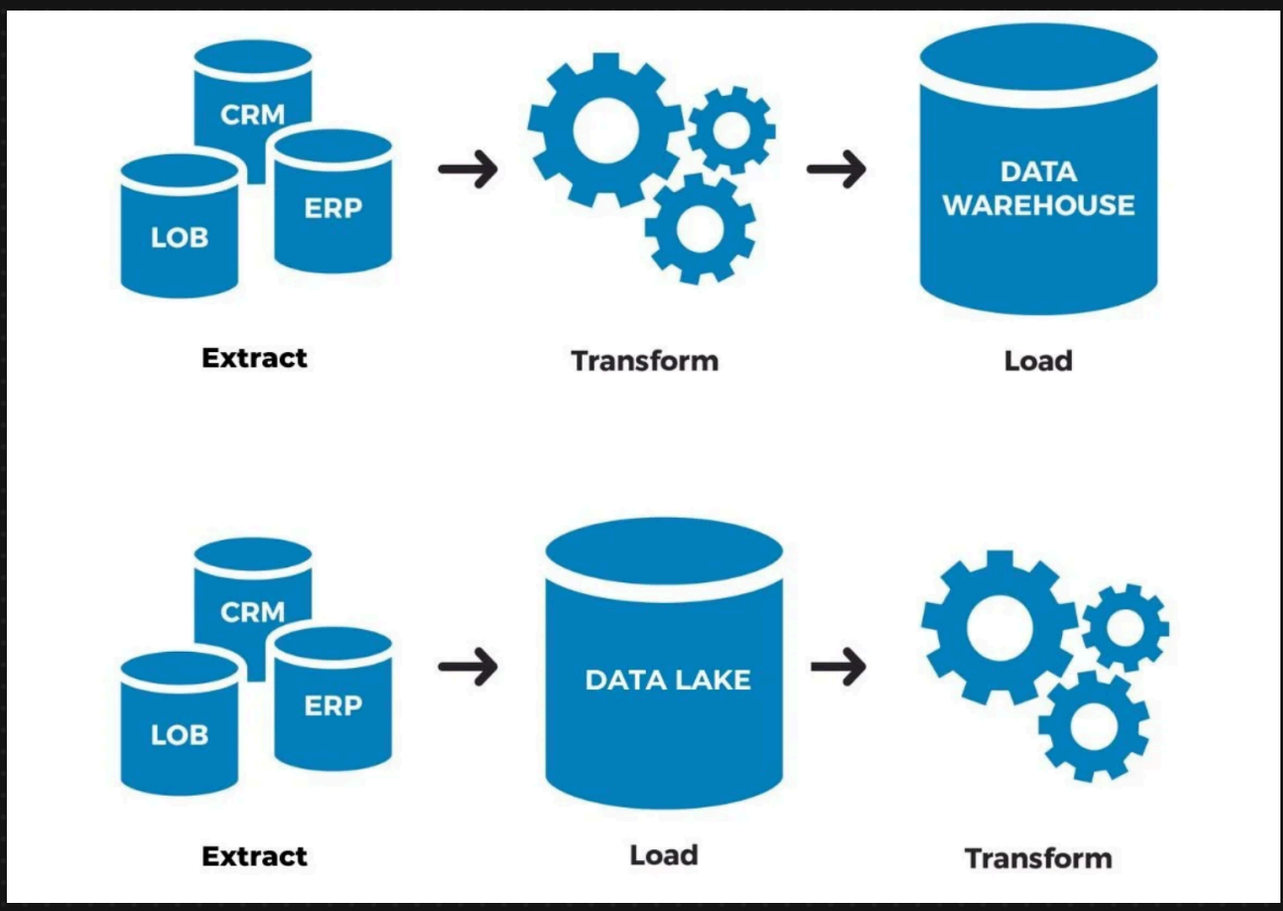
1. Collect data from multiple sources
2. Transform: Clean, filter and format the data for analytics
3. Load : Store cleaned data into DW

ELT process



Social media analysis

Aspect	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Processing Location	Data is transformed before loading.	Data is transformed after loading.
Target System	Traditional data warehouses with limited compute power.	Modern data lakes or cloud platforms with high compute power.
Data Type	Structured data.	Structured, semi-structured, unstructured data.
Speed	Slower, as transformations occur beforehand.	Faster, as transformations happen after loading.
Use Case	Bank transactions (cleaned before storage).	Social media analysis (raw data stored for future use).

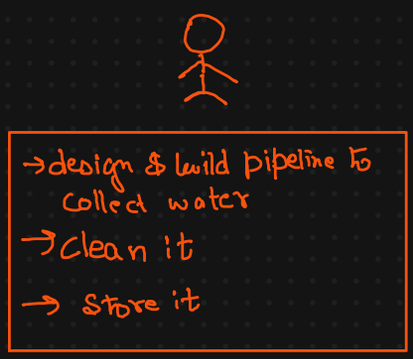

CRM
LOB
ERP
Extract
Transform
DATA WAREHOUSE
Load
CRM
LOB
ERP
Extract
DATA LAKE
Load
Transform

What Does a Data Engineer do & How Does Big Data Fit in ?

What is the difference between a data engineer and a Big Data engineer?

What is the role of data engineer ?

Water Sources



data analyst
data scientist
management

a data engineer is responsible for designing, building and maintaining the system that store, process & make data accessible for analysis.

→ Data pipeline creation
→ Data transformation
→ Ensuring data Quality
→ Automation

ETL + DW

Where does Big data fit?

massive flood of water

⇒ Specialized dams ⇒

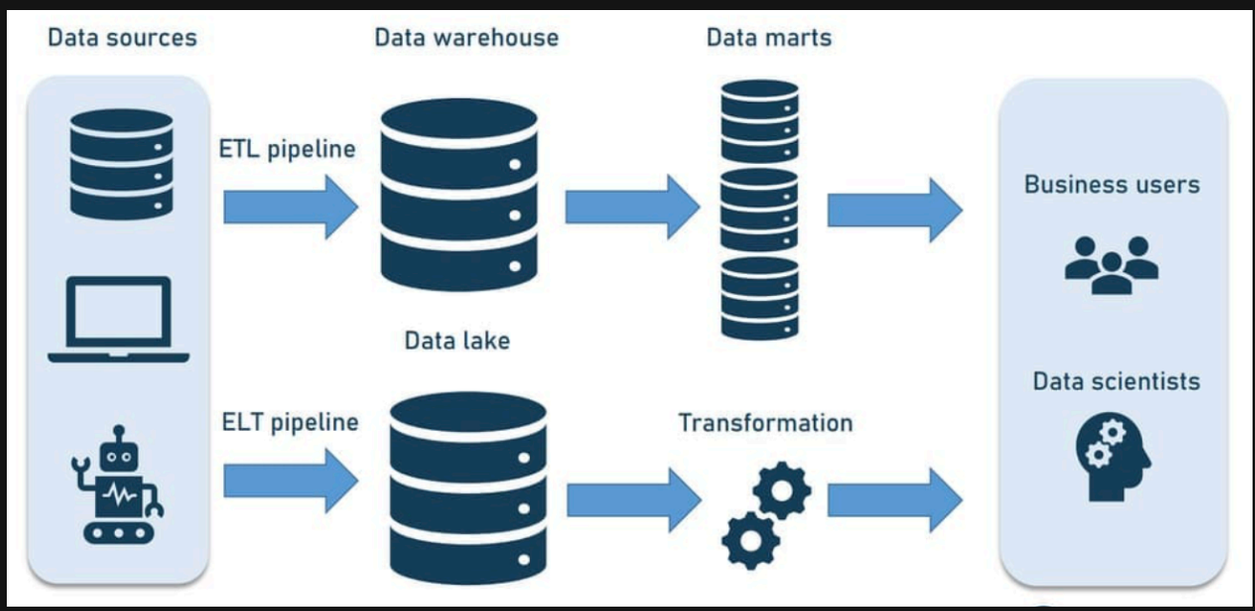
Big data refers to data which is large, complex that traditional systems cannot handle effectively

Big data engineer specialize in tool and technology that can process & manage massive data in distributed system.

ELT + Big Data

Aspect	Data Engineer	Big Data Engineer
Focus	Handles structured and manageable data.	Handles massive datasets (Big Data).
Tools	SQL, Python, Airflow, ETL tools.	Hadoop, Spark, Hive, Kafka, NoSQL databases.
Scalability	Works with traditional systems.	Works with distributed systems for scalability.
Storage	Data warehouses and databases.	Data lakes and distributed storage systems.
Processing	ETL pipelines for structured data.	Parallel processing for large-scale data.
Use Case	Preparing data for a monthly sales report.	Analyzing social media trends in real-time.

Big Data infrastructure with Data Lake



Module 2