Class-1 - Big Data

Audio ✓
Video ✓
Screen ✓

Zoom

raise your hand

Other people chat

chat ⟷ Q n A ⟷ Raise hand

(lot active) (very less) clearing doubts

[doubt on concept]

[community chat]

@MayankR

Agenda

→ Who am I ?

→ Introduction & Getting into Big Data

Mayank Aggarwal

→ 2012 → Coding 2013

love maths → coder

NSIT, Delhi → 1st year coded

2nd year Data Science → IIIT-D

3rd → ML, DL, AI, DSA → OYO

4th → PPO +

Professional

OYO → 5 months

Goldman Sachs → Big Data or Data engineering

Mindhickle → Spark + AI/ML

Ineuron

Scaler + Coding Ninjas

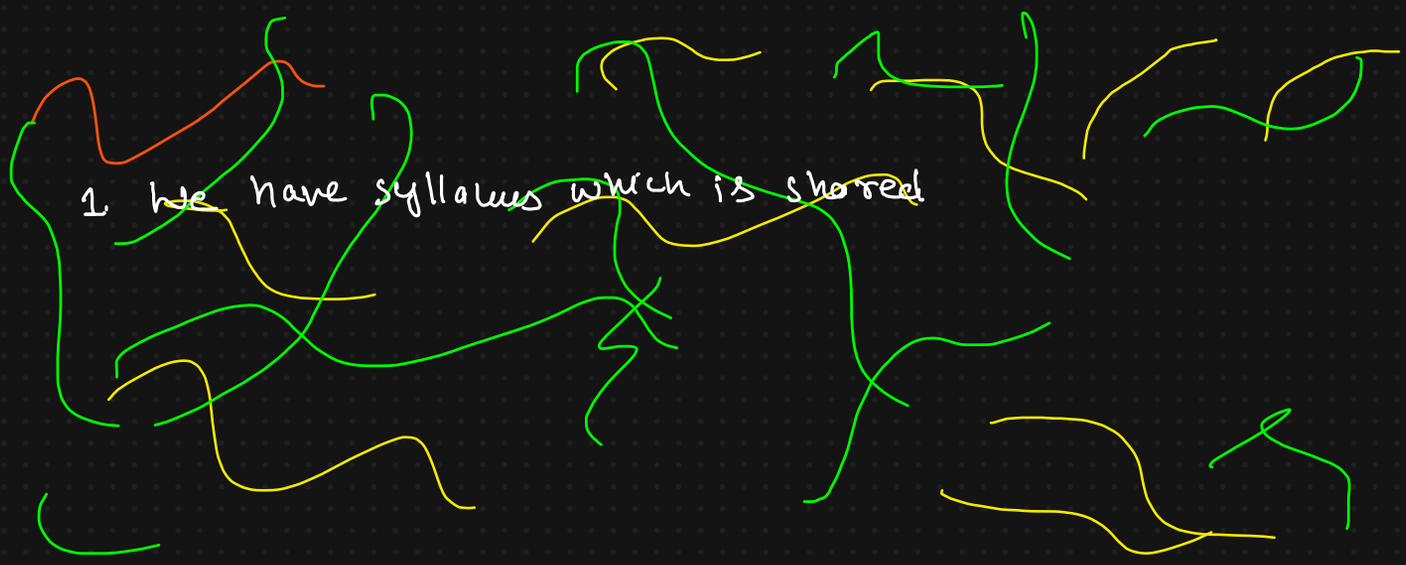
Course

1. Notes + Code + linKs + Recording + Files + pdf
   - → GitHub
   - → Resource Section
2. Class 8 - 11 Doubt Session Till 12

Few Instruction / Good practice

1. Be a blank paper
2. Make your own notes. [use pens]
3. Don't Jump to Future
4. Theory + Practical → Projects
5. Great Teacher → everything + how to learn +
   vs
   Good Teacher → everything
6. Dumb with doubts cleared than smart with having doubts

Feedback → -ve shewed


1. We have syllabus which is shared

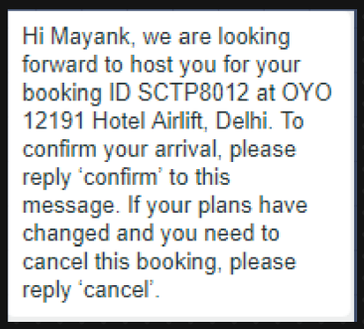
Big Data

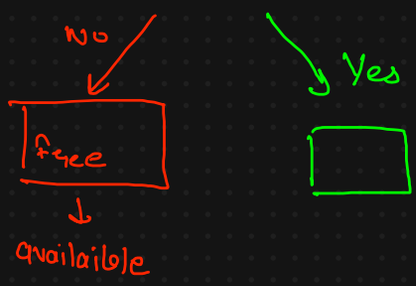
Data Science

Kaggle , mail , pendrive , sql

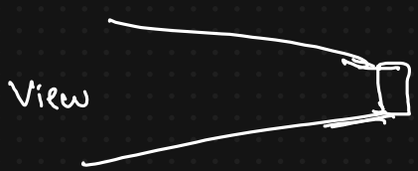
Whatsapp chat Automation

Rdbms
→ transaction
→ small volume

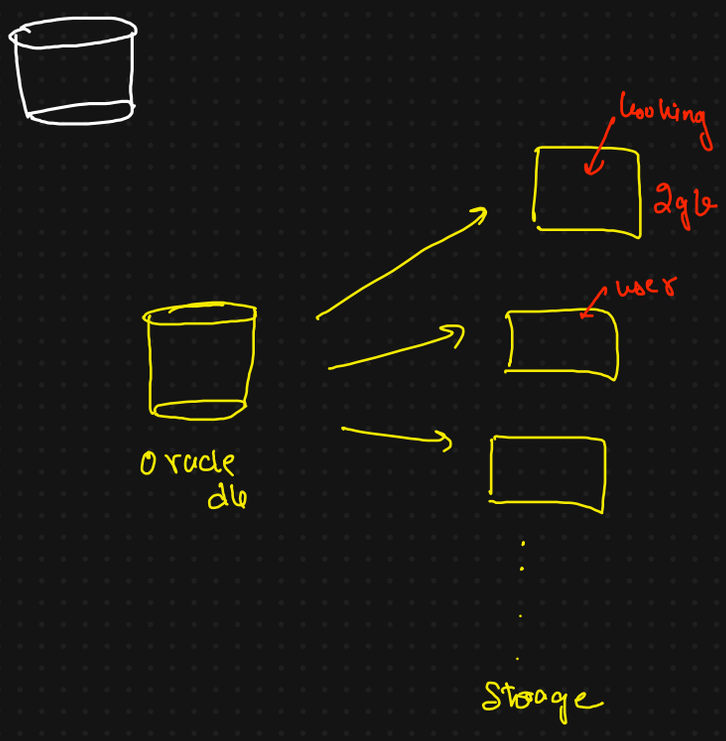




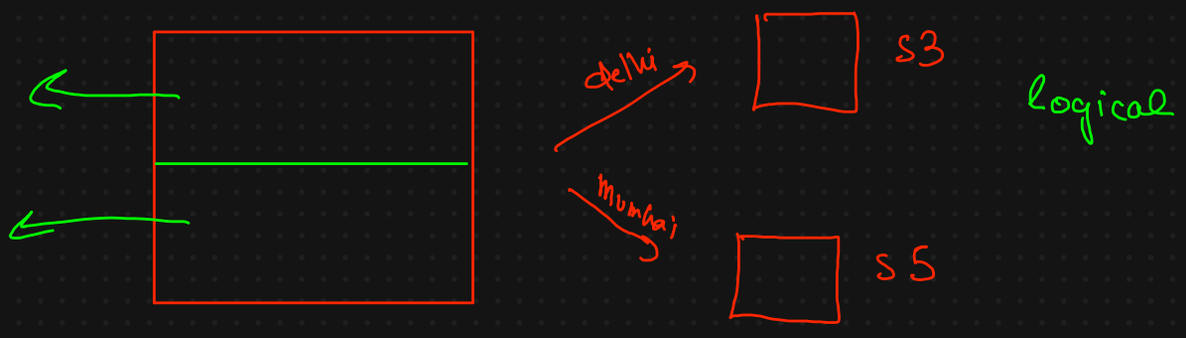
2nd project → Delhi - 2x

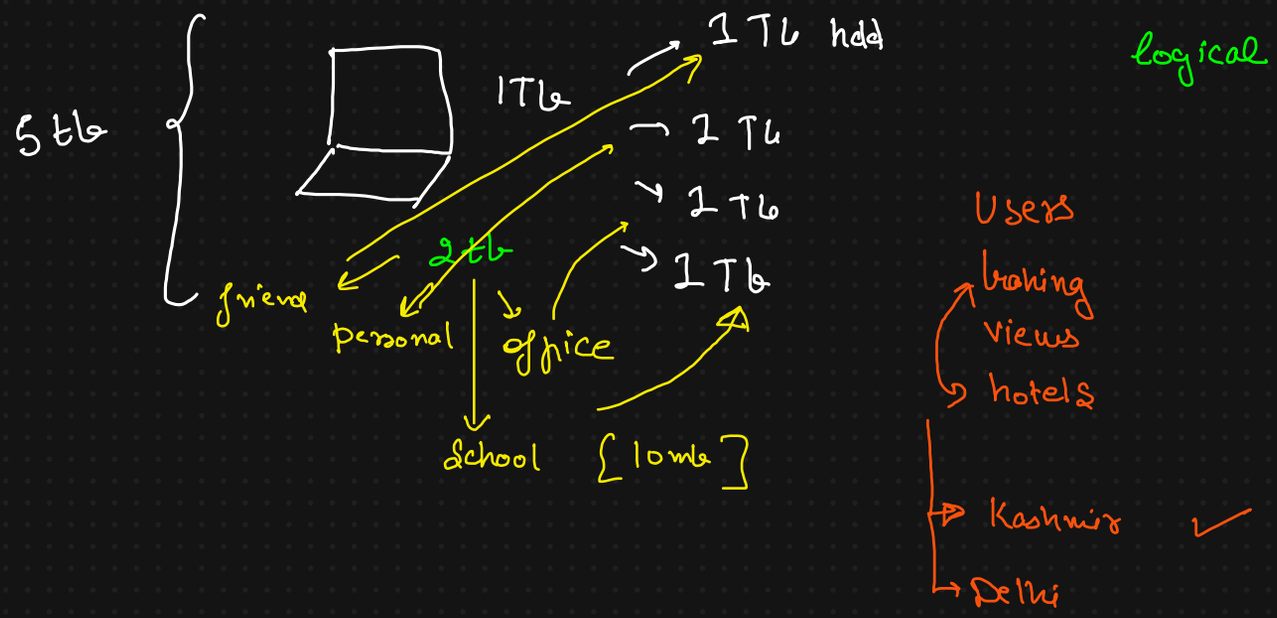


gb/tb can be stored



1. There is still a limit which we can expand
2. as data increases we have to partition our data
3. Unstructured data ✓
4. Costly.





Big data : huge amount of data that cannot be handled by traditional systems.

These systems deal with data that is too large & complex for traditional system to handle

Big Data is problem & solution

Why Big data emerge?

1. Internet
2. Social Media
3. IoT

5 V's of Big Data

5 Vs of Big Data are crucial / provide a framework for understanding challenges & characteristic of Big Data.

1. Volume :- size of data getting generated

OYO vs GS

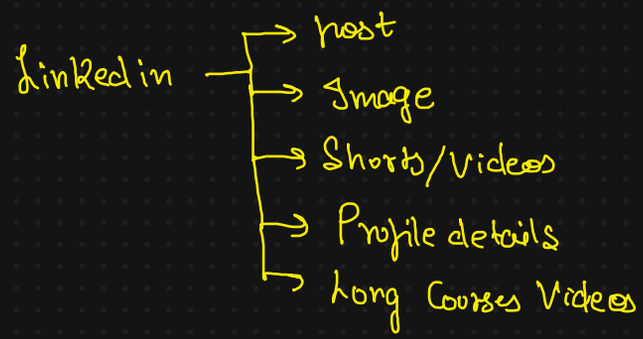
there is no set volume when your problem becomes a big data

2. Velocity: Speed at which data is getting generated & needs to be processed.

Real Time	Near Real Time	Batch
Bank Transaction	data is continuously generated but we are taking 1min, 2min buffer	Credit card
Fraud		[data engineer]
live feed	Amazon → Confirm in 2min	once a month
	live tracking	

Specific time period

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



Structured | Semi structure | Unstructured

Row & columns

Schema is enforced

Parquet

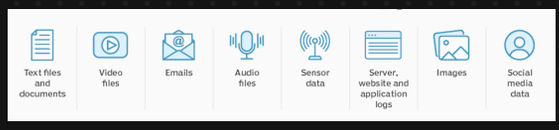
Structured data

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.

Json
xml
csv

```
<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>
```

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



4. Veracity : Trustworthiness or Quality of data
   → messy data needs to be corrected.
   → -ve age, Quality issues

5. Value : data should be useful & provide some value
   → help in business decision
   → provides insights

Volume ≠ Big data

For a problem to be a big data problem, not all V's need to be satisfied. generally a combination of 2~3 is enough

Big Data & Distributed Systems.

gta 3 / San andreas ⟶ gta 5

We need more resources

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

Single system

Monolithic Systems

1

Single, Self contained unit

X → 2X → 10X

P → 2P → 6P

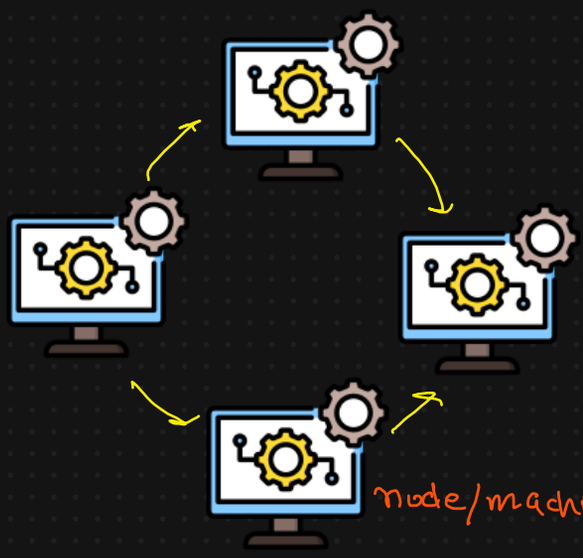
Scalability problem

cost & performance

Vertical Scaling

Multiple Systems

Distributed architectures



node/machine

Cluster

True Scaling

X → 2X → 10X

P → 2P → 10P

they don't have problem of scalability

Horizontal Scaling

All good big data systems are based on distributed architecture

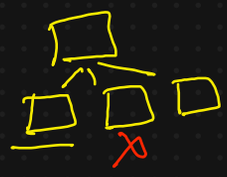
Designing a Good Big Data System
Cloud vs On-Premise
Database vs Data Warehouse vs Data Lake
ETL vs ELT
Data Engineer vs Big Data Engineer
Hadoop Introduction

Class 2 : Terms in Big Data

Designing a good Big data system

1. Scalability: Flipkart Big billion
   (a) Storage scalability: store increased data
   (b) Processing scalability: process large/complex data faster by adding more m/c.

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



3. Cost effectiveness: System should be able to balance performance & scalability with cost.

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

On Premise vs Cloud

When choosing to deploy Big Data solutions, companies face an issue between on-premise vs cloud.

On-Premise

Buying an office

- → space
- → initial investment
- → procure all machines
- → install software, OS, security updates
- → setting up
- → operating expenditure

* 20 = 20 nodes

More machines → Scaling up

+10 more machines

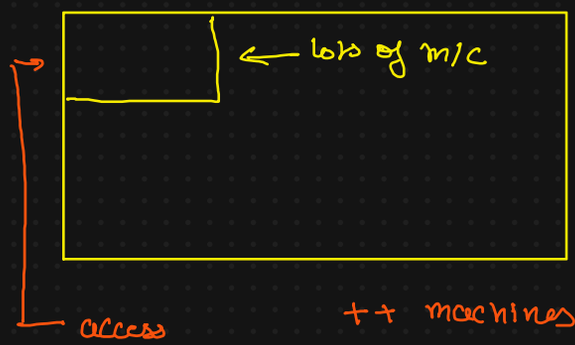
1. Security

Cloud

Renting

aws
gcp
Azure
IBM

pay-as-you-go model



++ machines

-- machines

Infra on rent

Advantages.

1. Scalability
2. Operating expenses : more operating expenses

   Capex vs opex
3. Fault tolerant

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.

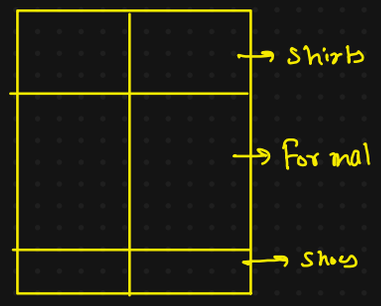
Types of cloud

1. Public : aws, azure, gcp
2. Private: infra set up by company. Compliance & security
3. Hybrid : Private + Public
4. Community : shared by organization with common concern.
   eg: University
   Hospitals

Data Base vs Data warehouse vs Data Lake

Database

→ System which stores the data
→ Organized i.e. has a structure
→ Structured data with defined schema



Transaction data = Day to day operations eg. (updates balance order, inventory)

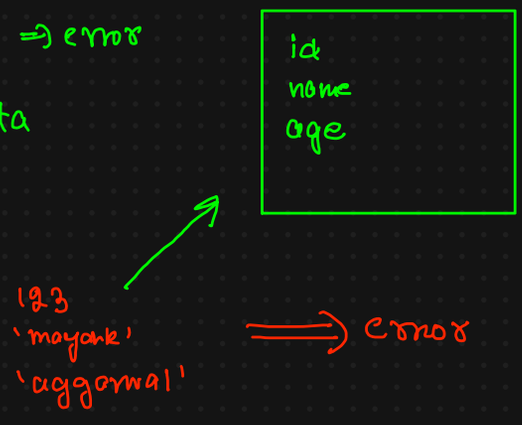
OLTP (Online transaction processing)

don't hold years of data, recent data for perform.

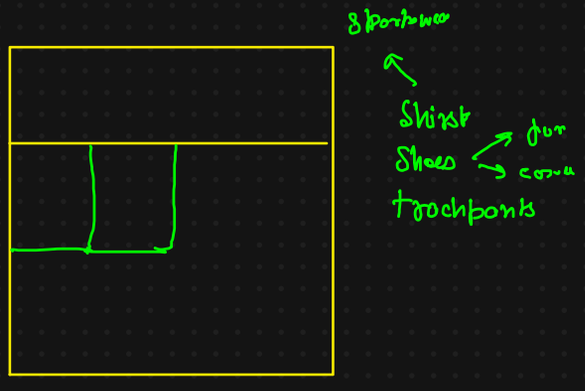
We store just the recent data due to performance & cost

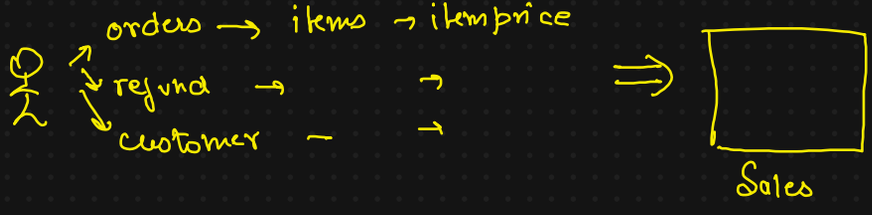
Schema on write — any mismatch ⇒ error

overall cost high ⇒ just for recent data

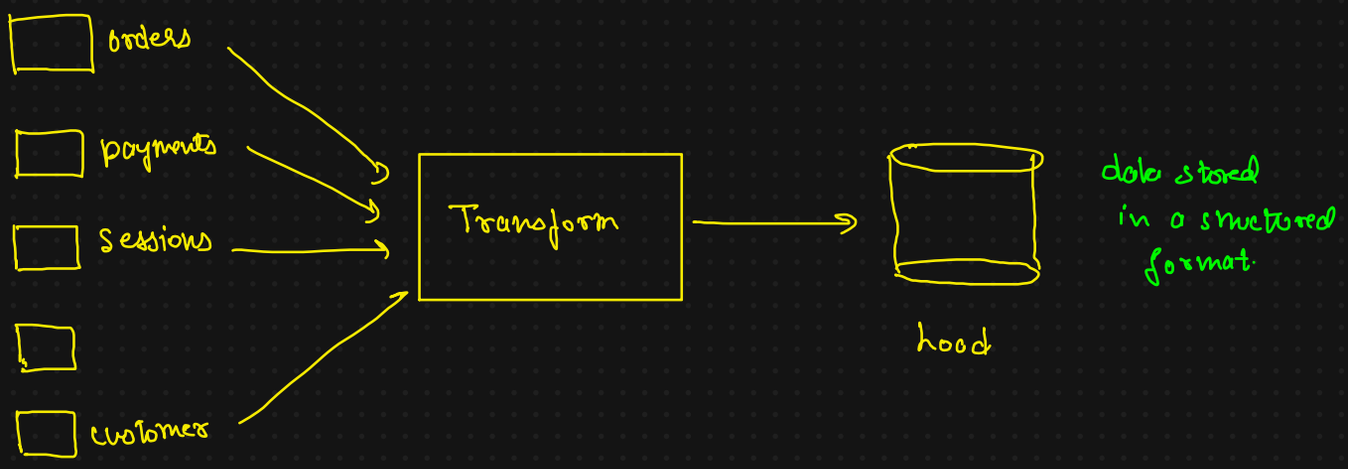


Data Warehouse





Teerradata , Informatica , Talend



Extract Transform load

Q: Why can't I use a database?

1] There are multiple sources of data
2] lots of data ⇒ cost ↑↑
3] We write lots of Queries quite complex on DB

Performance ↓↓

Complex Queries

transaction write

Read Heavy

OLAP ⇒ Online analytical Processing

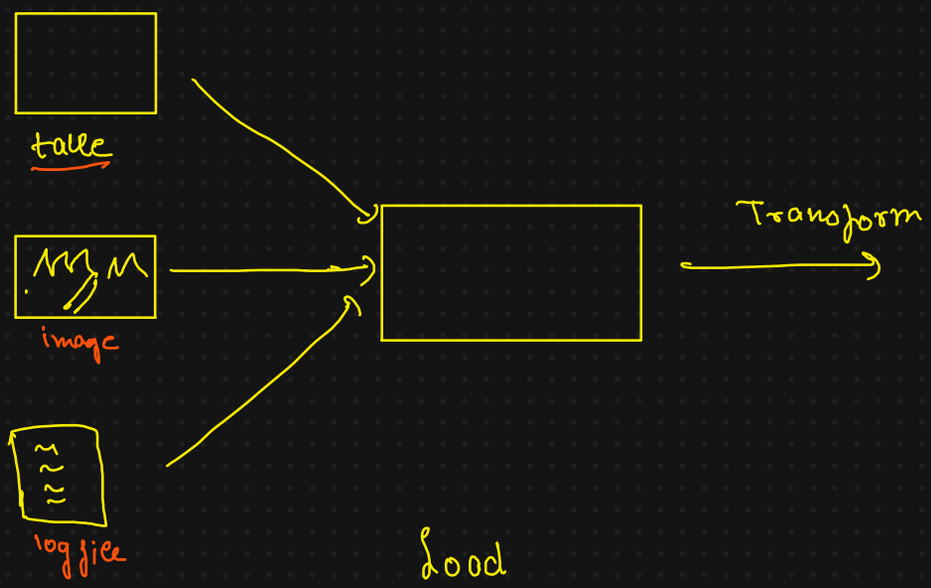
3. Data Lake 23. hdfs

we are now majorly dealing with different kinds of data

Structured Unstructured Semi structured

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

nlp



ELT

Load

Extract Load Transform

⊕ Cost effective
enough flexibility

Many teams can use this

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

Break ⇒ 10min 10:00

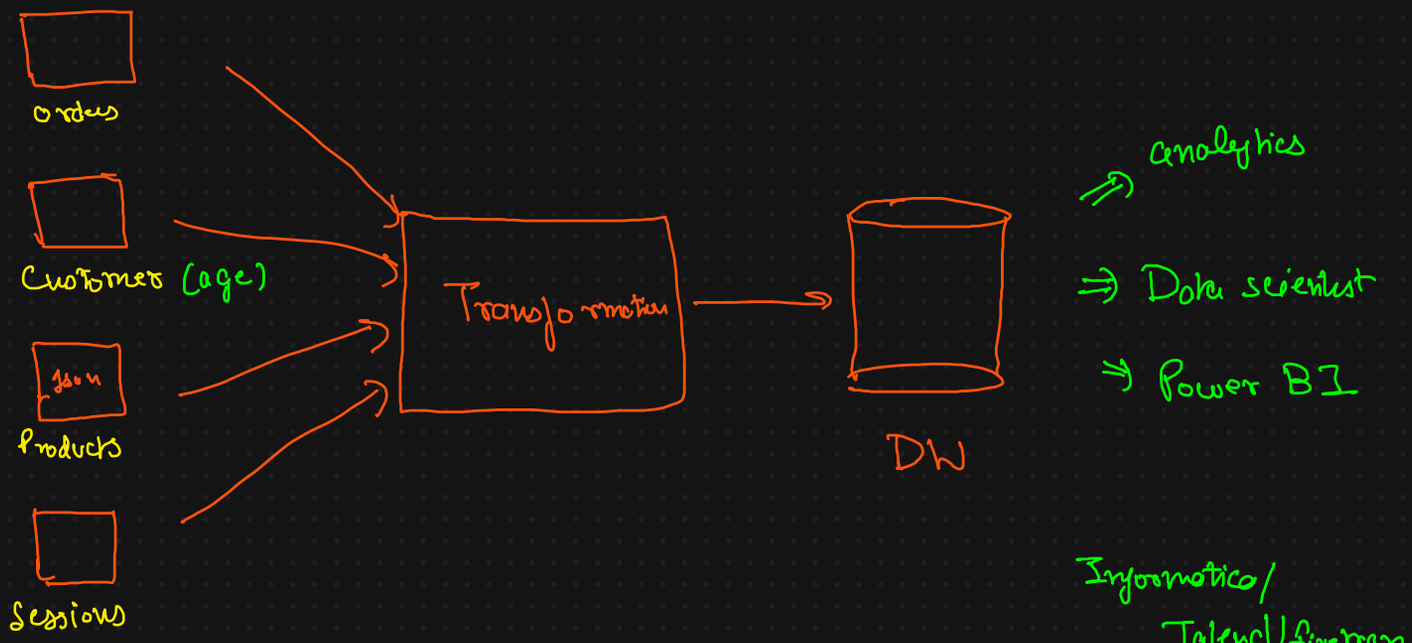
Transformation :-

Aspect	OLTP	OLAP
Purpose	Handles day-to-day transactional data.	Handles historical data for analytical queries.
Data	Real-time, current operational data.	Historical, aggregated, or summarized data.
Operations	Frequent INSERT, UPDATE, DELETE operations.	Primarily SELECT queries for analysis.
Query Complexity	Simple, fast queries (e.g., fetching single records).	Complex queries with joins, aggregations, and filters.
Schema Design	Normalized schema (e.g., 3NF) to minimize redundancy.	De-normalized schema (e.g., star or snowflake schema) for faster querying.
Transaction Type	Short, atomic transactions (e.g., order placement).	Long, batch-oriented queries (e.g., sales trends analysis).
Users	Operational staff, front-end applications.	Data analysts, decision-makers.
Data Volume	Smaller data sets, as it handles current operations.	Very large data sets (terabytes to petabytes).
Performance	Optimized for fast write operations.	Optimized for read-intensive operations.
Examples	Banking systems, e-commerce sites, CRMs.	Data warehouses, BI tools, analytical dashboards.
Tools/Databases	MySQL, PostgreSQL, MongoDB, Oracle (OLTP mode).	Snowflake, Amazon Redshift, Google BigQuery.

ELT vs ETL

Extract Transform Load

homework

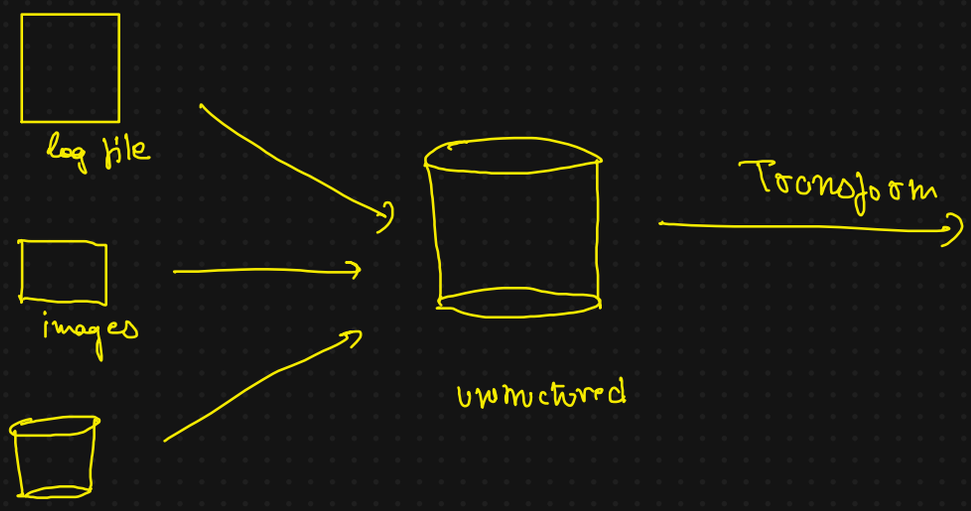


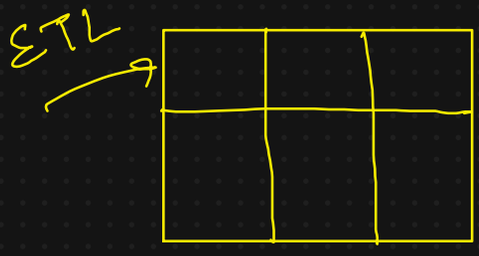
Collect data from multiple sources

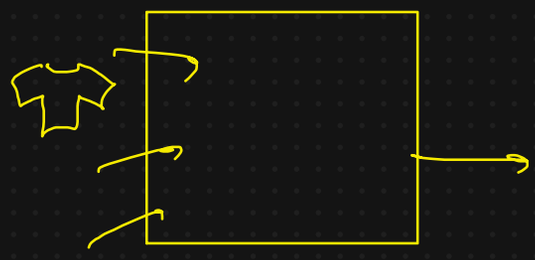
Clean, filter and format data for analysis

Clean data stored in DW

ELT process





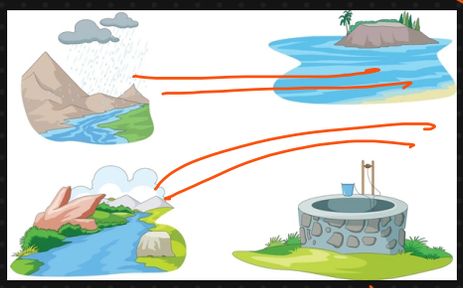


Aspect	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Processing Location	Data is transformed before loading.	Data is transformed after loading.
Target System	Traditional data warehouses with limited compute power.	Modern data lakes or cloud platforms with high compute power.
Data Type	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).

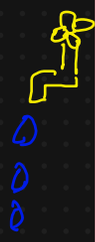
Data engineer vs Big Data engineer

→ data engineer

→ water pipeline
→ restaurant owner



designs & build pipelines
to collect water
→ clean it
→



Water Sources

data analysis
data scientist

⇒ Data pipeline creation
⇒ Data transformation
⇒ ensuring data Quality
⇒ Automation

ETL + DW

Q: Where does now Big data fits in?

massive floods
ocean

⇒ Specialized Dam ⇒

Big data engineer specializes in technology that can process and manage massive data in distributed systems

ELT + DL

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.