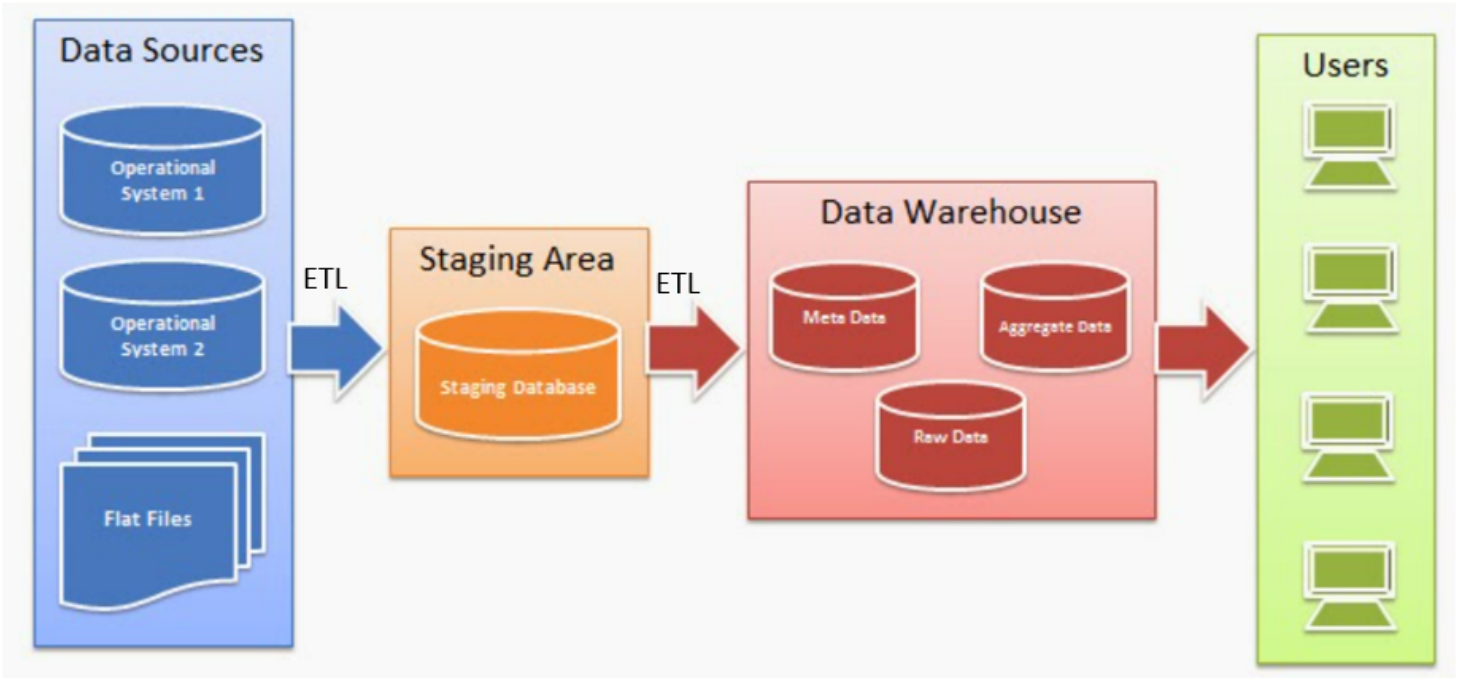
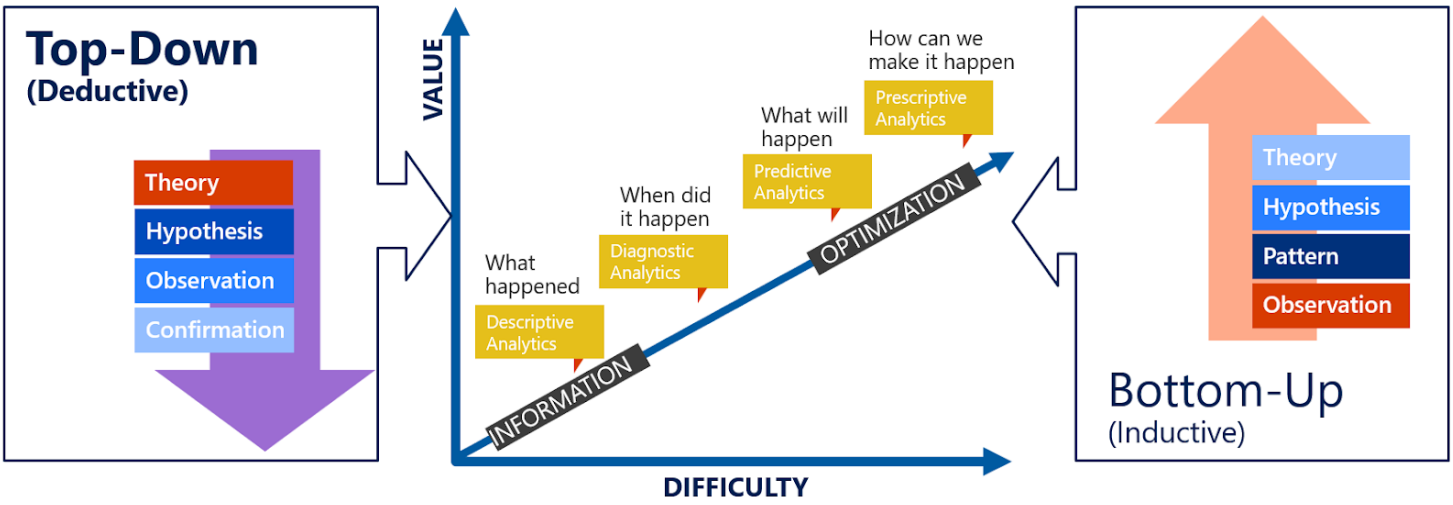
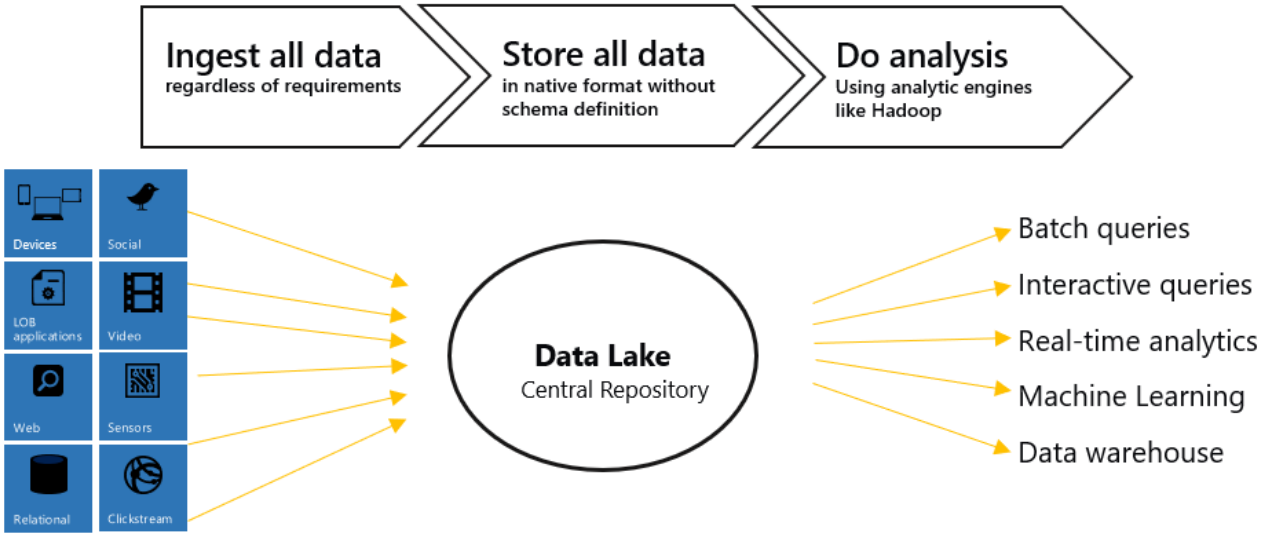
Agenda

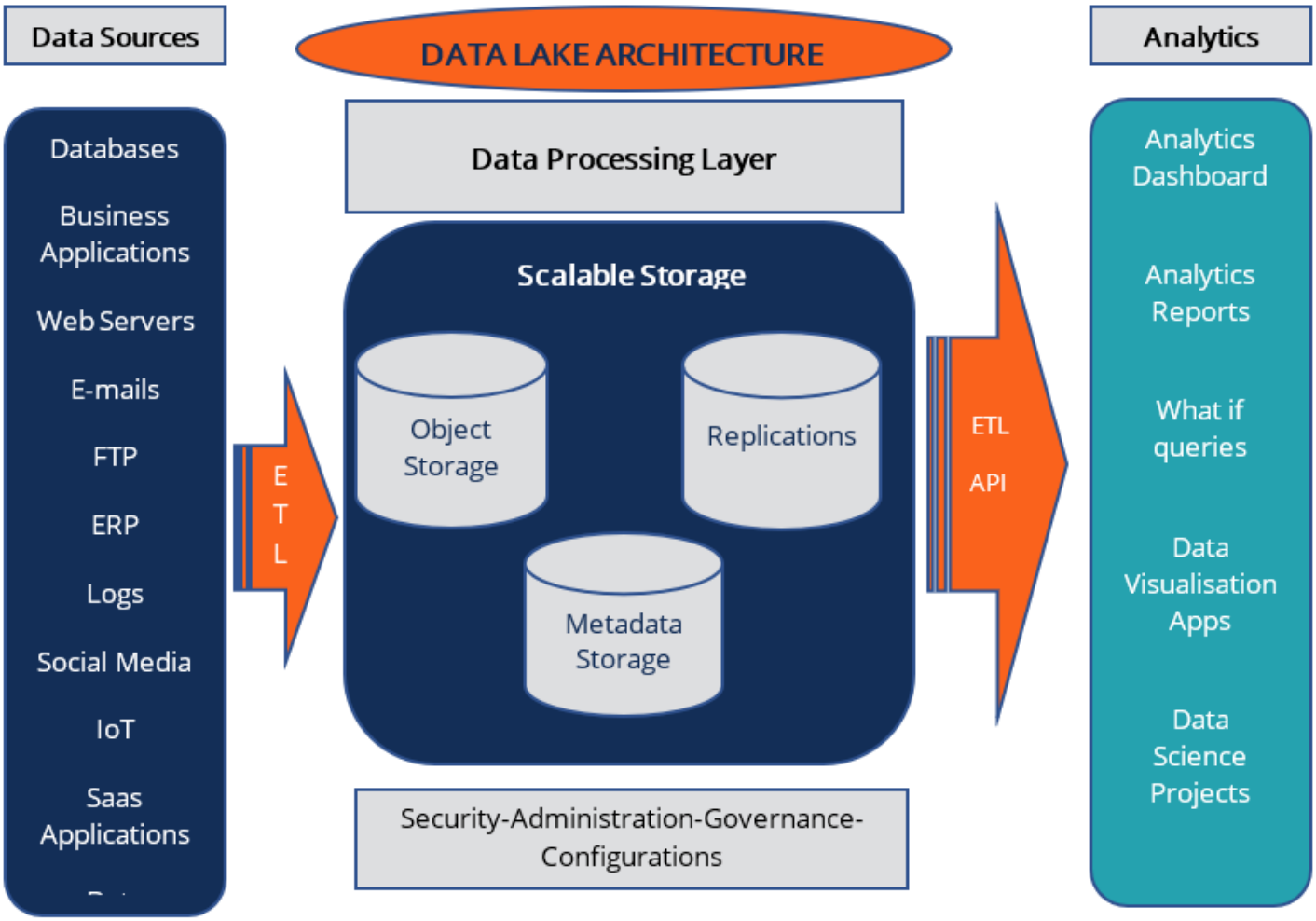
Problem Statement
Existing Problem
So what is Data Lake ?
How are they different from Data warehouses?
Why Netflix choose data lake
What are the components involved in Data Lake Architecture
How to create a Successful Data Lake
How to form a Data lake with AWS Components

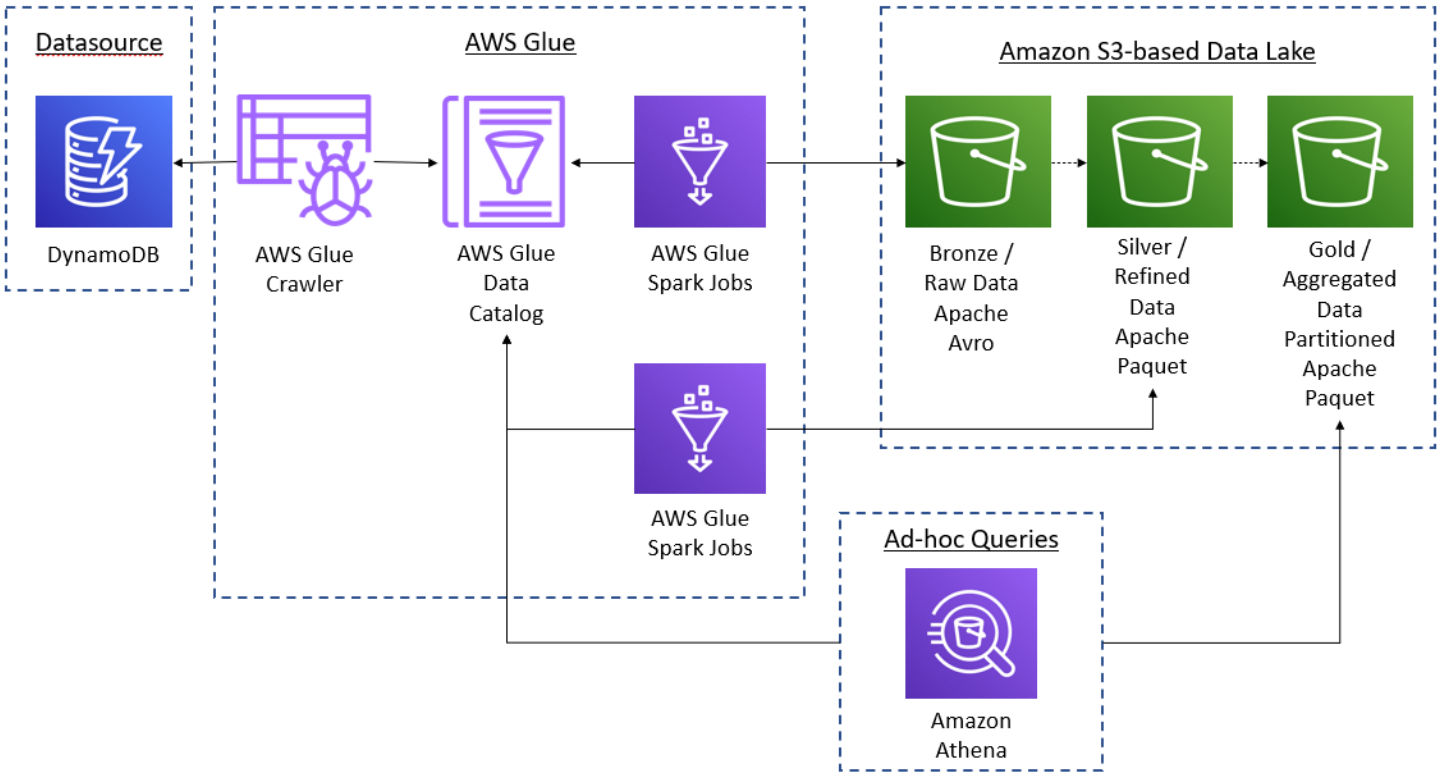

Data Sources
Operational System 1
Operational System 2
Flat Files
ETL
Staging Area
Staging Database
ETL
Data Warehouse
Meta Data
Aggregate Data
Raw Data
Users

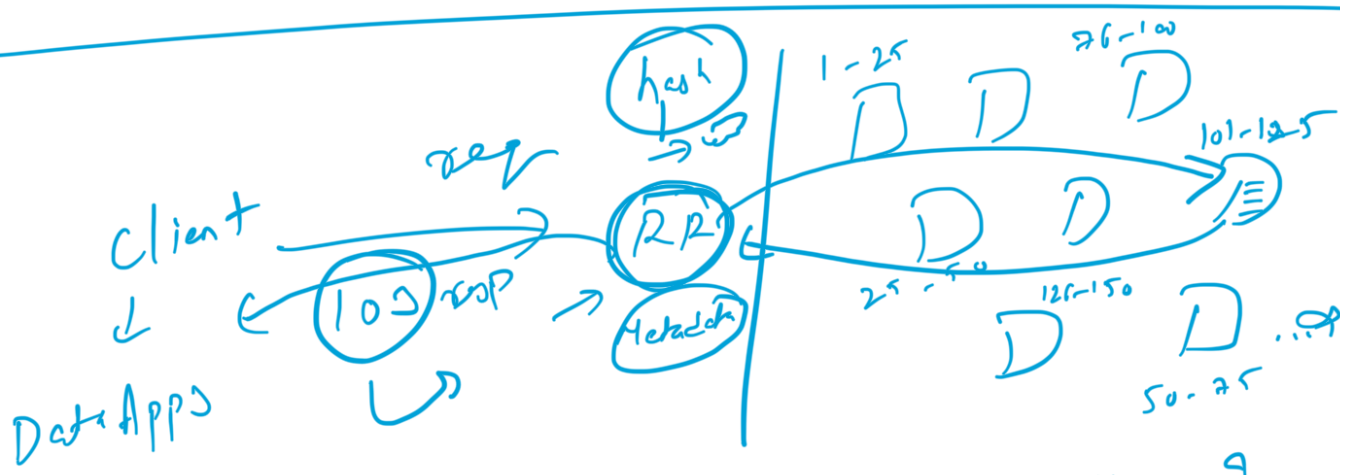

Top-Down
(Deductive)
Theory
Hypothesis
Observation
Confirmation
VALUE
What happened
Descriptive Analytics
When did it happen
Diagnostic Analytics
What will happen
Predictive Analytics
How can we make it happen
Prescriptive Analytics
INFORMATION
OPTIMIZATION
DIFFICULTY
Theory
Hypothesis
Pattern
Observation
Bottom-Up
(Inductive)

The Data Lake Uses A Bottom-Up Approach






Datasource
DynamoDB
AWS Glue
AWS Glue Crawler
AWS Glue Data Catalog
AWS Glue Spark Jobs
AWS Glue Spark Jobs
Amazon S3-based Data Lake
Bronze / Raw Data Apache Avro
Silver / Refined Data Apache Paquet
Gold / Aggregated Data Partitioned Apache Paquet
Ad-hoc Queries
Amazon Athena



→ How Data is actually Distributed?

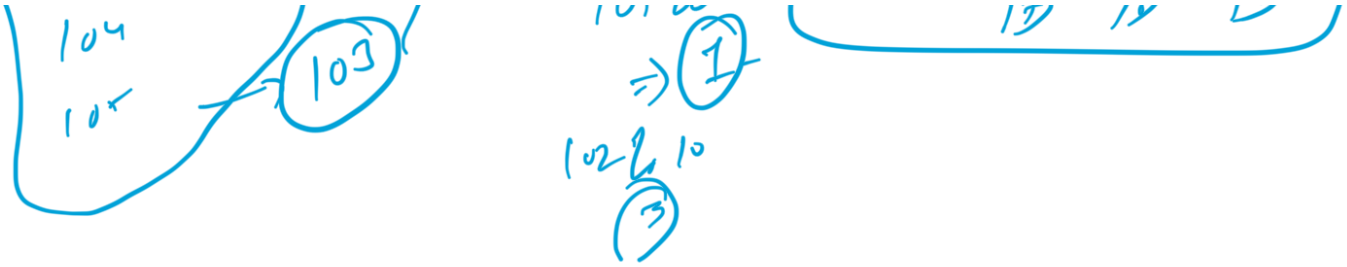
→ Distribution key / Partition key

→ Sort key

(a) Distribution key :-

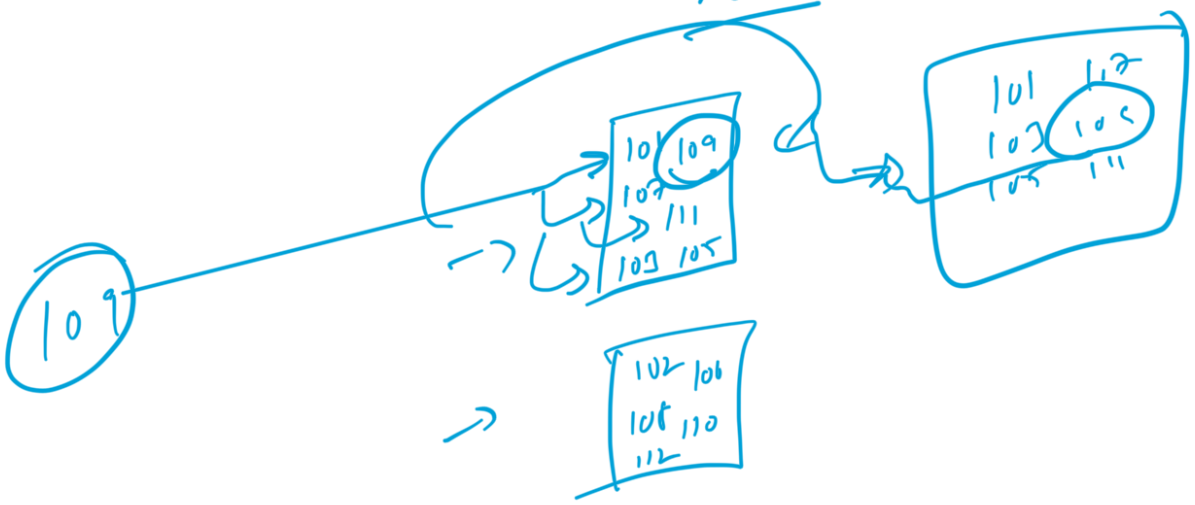
↳ determines how data is distributed across different nodes in your Redshift cluster.

PK → DK ⇒ High Cardinality Columns help in distribution



② Sort Key

↳ how the data within each node is sorted on disk.



⇒ Distribution style

↳ how the data in a table is distributed across nodes in a cluster.

① Even

→ Dist style Even

→ Data is distributed evenly across all in the cluster

② Key

→ Dist style key

→ DistKey (Function)

⇒ Data is distributed based on value of

③ All

→ Dist style All

⇒ A full copy of entire table is stored

nodes

→ Each node will get an approximate equal portion of data.

specified col.

node in cluster

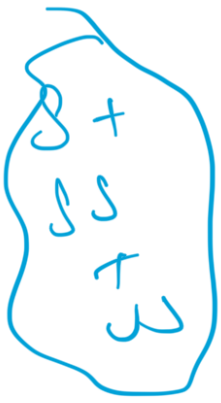
⇒ Which one to choose

↳ Key : use when you have a clear join col that can act as dist key particularly for large fact table.

↳ Even : use when you want to ensure even data distribution across nodes and there is no clear DK.

↳ All : use for small tables that are frequently joined with large table, to avoid data movement during joins.

Data Lake



Netflix → PBs

↳ Personalization

↳ Performance analytics

↳ experimentation

Issues
- ↳ To store all data
- ↳ No early access to Data
- ↳ Scaling
- ↳ Price

Solutions :-
- ↳ Data ~~Streaming~~
- ↳ No Sql DB
- ↳ Hive

Data lakes

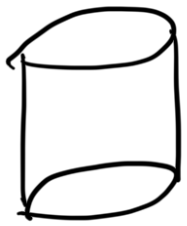
↳ are Central data repositories used to store any data at any scale.

Different from Data warehouse

	Data lakes

Data Warehouse

1. Store Structure Data
2. Top down Approach



→ CDB
→ CT
→ Load the data
→ query

3. ETL
4. BA | DA
5. TBs | PBs
6. Expensive

1. Store Structure, Semi Structure or Unstructure.
2. Botton up approach
3. ELT
4. DS
5. PBs | EBs | HBs
6. cheapest

Yulu BroadCom NetApp Arista CDAC

Paytm

S3 → Azure blob → DL

DL = E L T

AWS
↳ Glue
↳ S3
↳ DynamoDB
↳ Athena

GCP
↳ Dataproc
↳ cloud sql
↳ Bigtable
↳ IoT

---

① Data Sources :- provides the real time business data to DL.
↳ IOT | devices | DB | KafKa | weblogs | FTP | CSV...

② Data storage :- ⓐ object storage
↳ objects (large volumes of unstructured data)

ⓑ Metadata storage
↳ provides properties about the object.

* ⓒ Replication
↳ replicates objects across multiple locations of DL for DR and backup

(3) Data processing layer

↳ retrieved data is processed to produce info that is applicable to business.

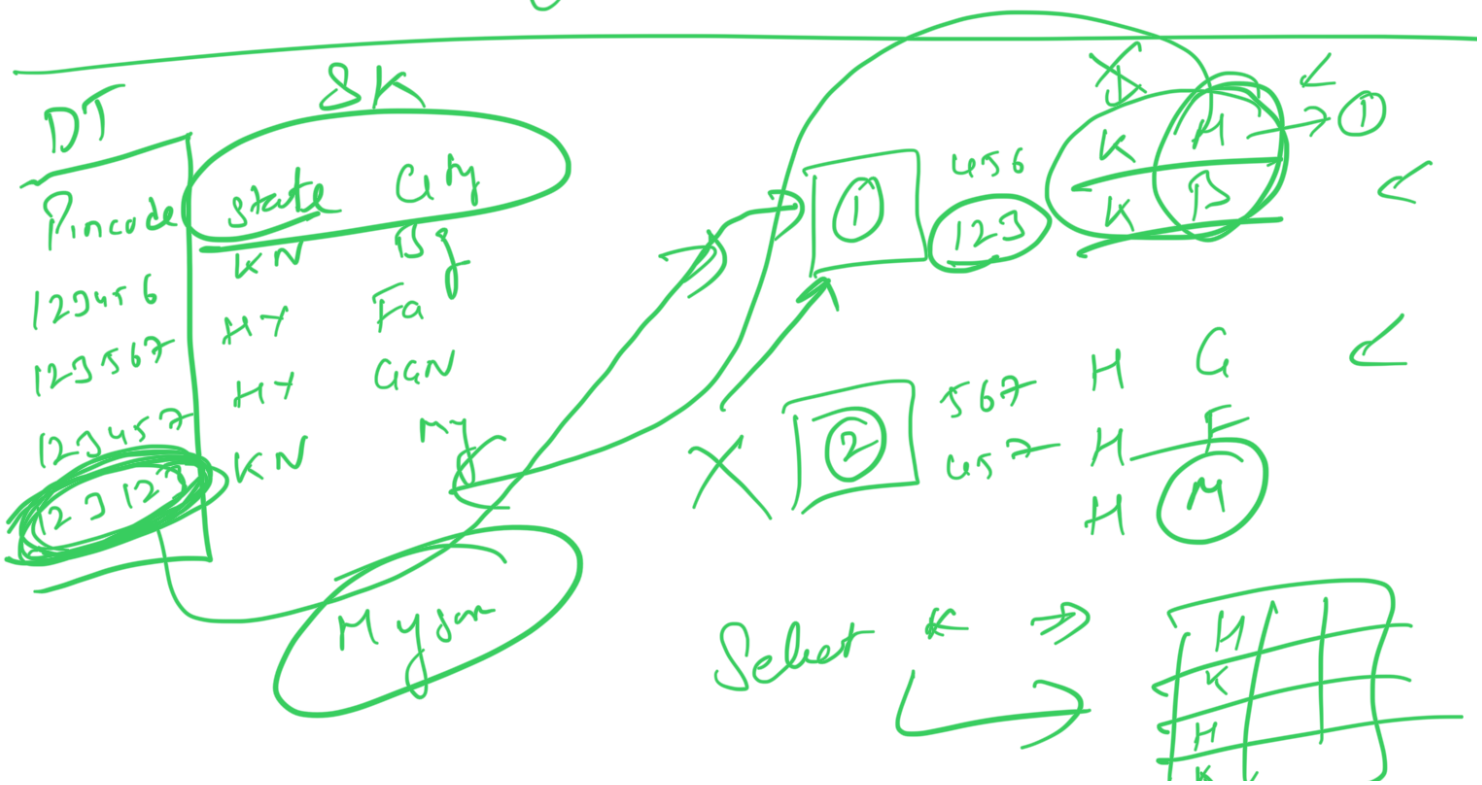
(4) Data Analytics layer:- Dashboards + Reports

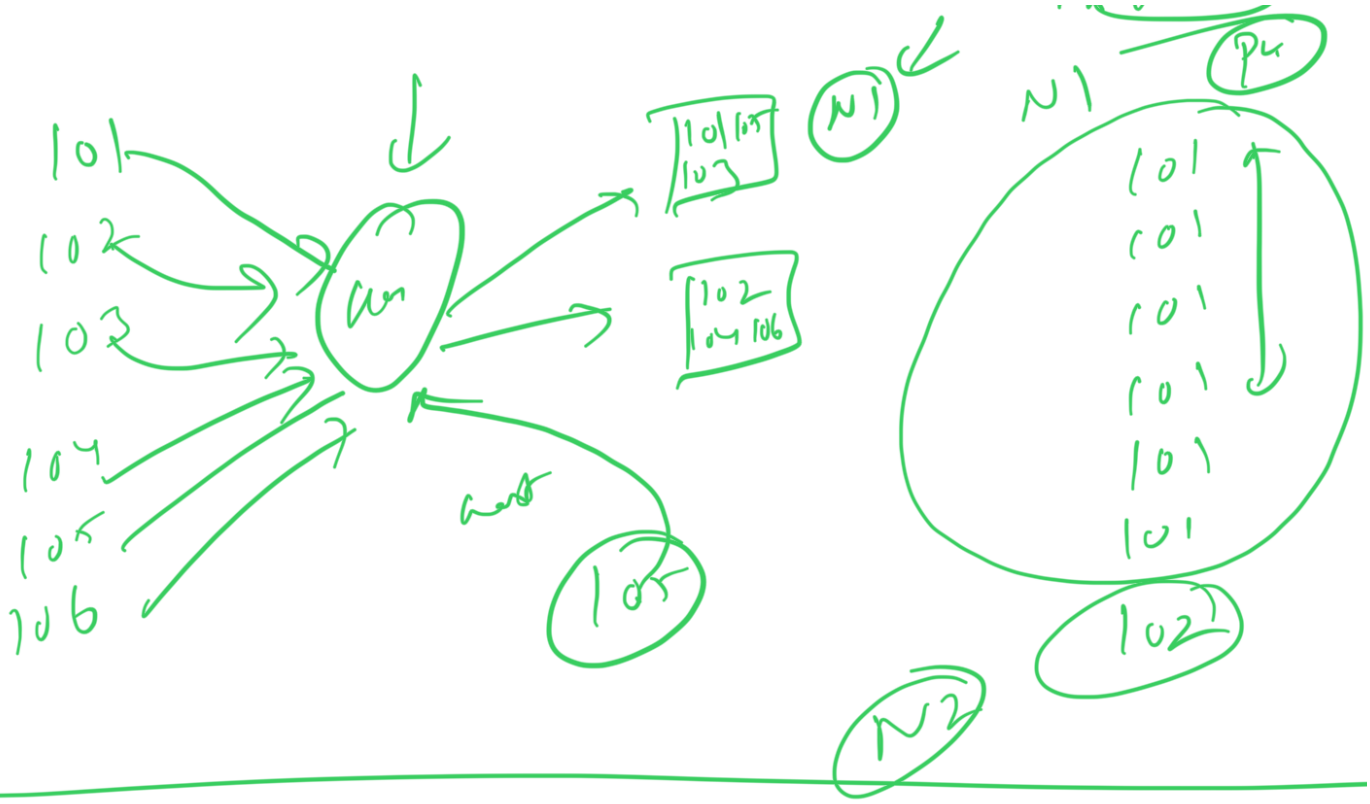
Use Cases:-

Raw Data (bronze)

↓

Curated (Gold)





DCS

Sales
- txn_id (PK)
- C id
- txn_date
- P id
- amount

→ DK

txn_id	C id	txn date	
1	1	02 dept	iPR
2	1	05 dept	Cover
3	2	02 dept	i'od
4	3	06 dept	cover

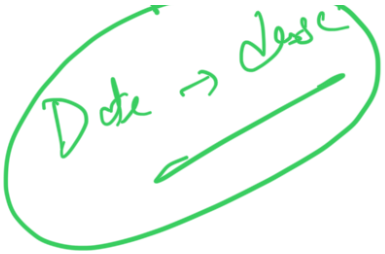
1 Billion Row

Sort key Data	
1	02 dept
2	04 dept
3	03 dept
4	06 dept

RR → Hashing

3	03
1	02

1 3	N1
2 4	N2


Date → Desc