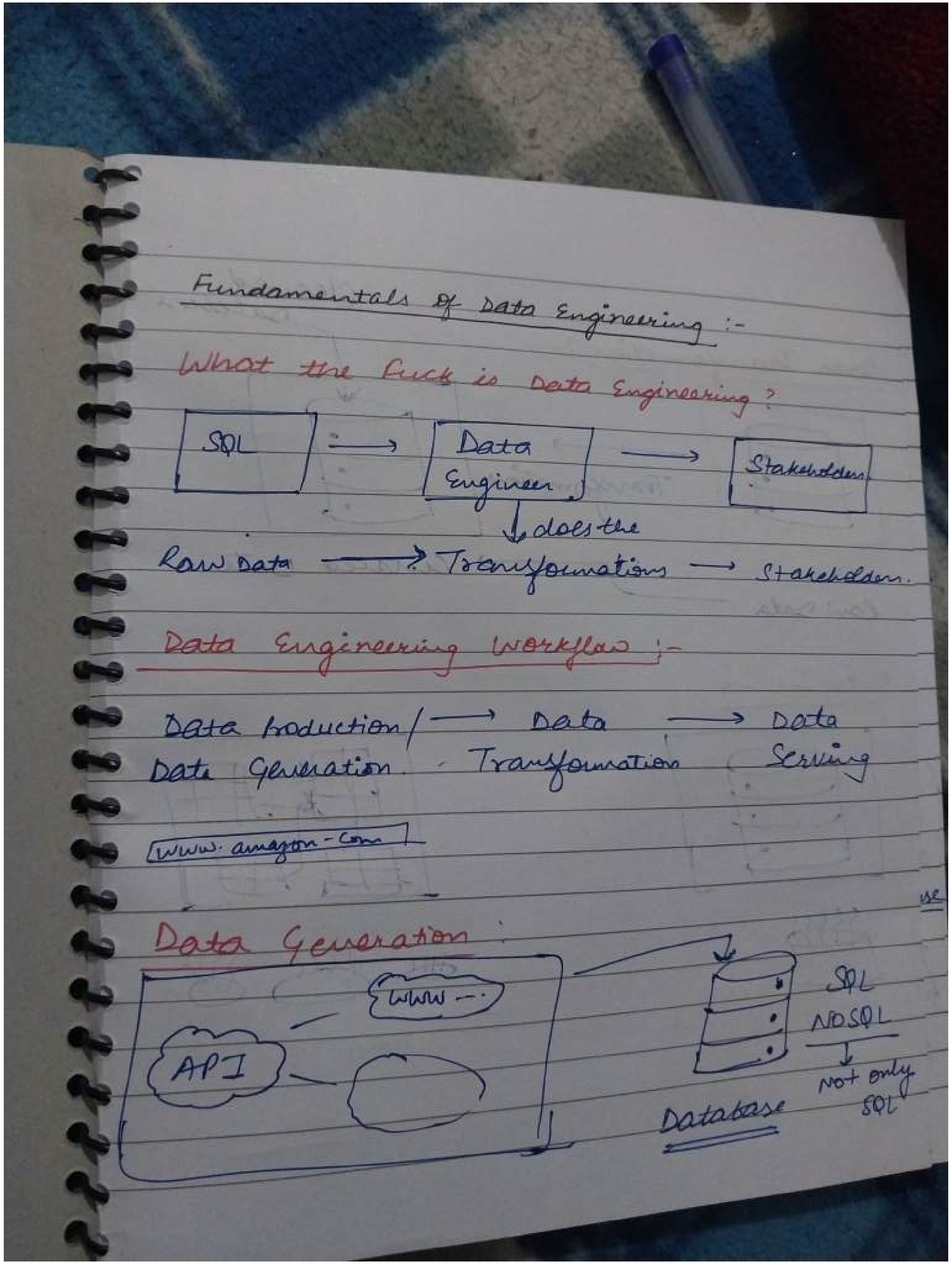


Fundamentals of Data Engineering :-

What the fuck is Data Engineering ?

SQL → Data Engineer → Stakeholders

↓ does the

Raw Data → Transformations → Stakeholders.

Data Engineering Workflow :-

Data Production/ Data Generation → Data Transformation → Data Serving

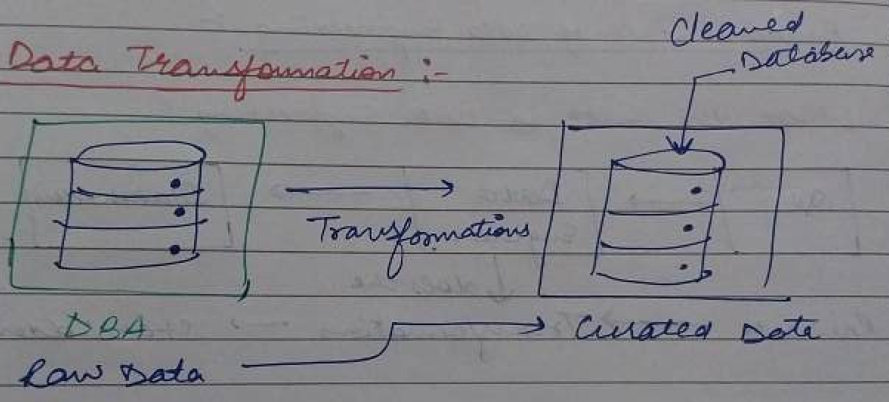
www. amazon-com

use

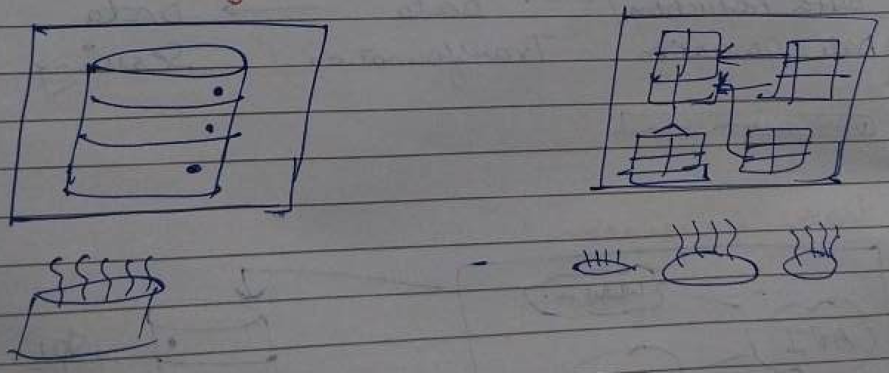
Data Generation :

www --
API
Database
SQL
NOSQL
Not only SQL

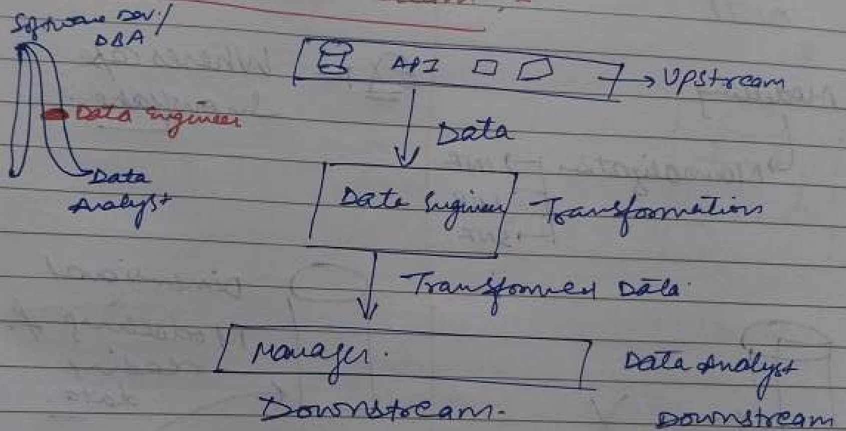
Data Transformation :-



Data Serving



Upstream - Downstream :-

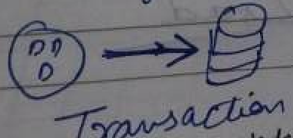


OLTP

- Online Transaction Processing

EX : MySQL, PostgresSQL

- Managed by DBAS.



- Writes and Updates Efficient

OLAP.

Online Analytical Processing.

- Also called Data warehouse

EX :

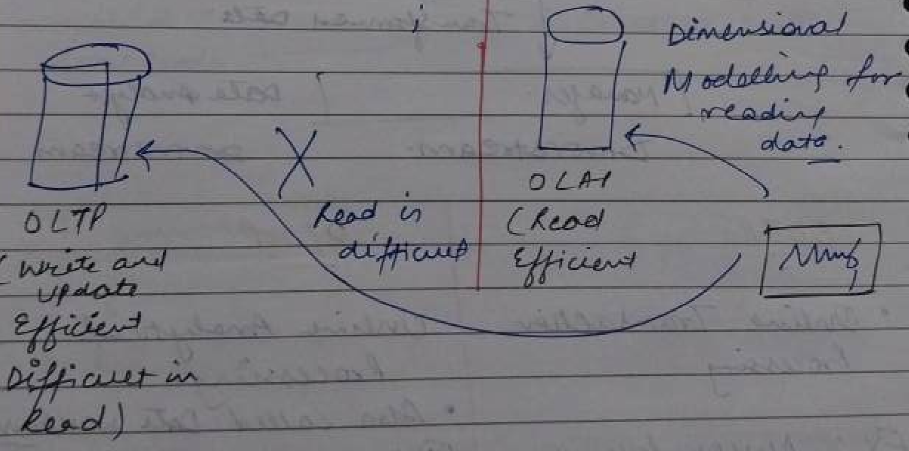
- Read Efficient.

OLTP.

Modelling
- Normalization
  - 1NF
  - 2NF
  - 3NF
  - ⋮

OLAP.

EX :- Wherescape
Snowflake.

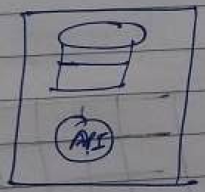


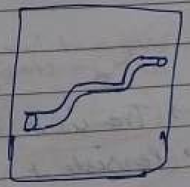
Dimensional Modelling for reading data.
X
Read is difficult
OLTP
(Write and update Efficient Difficult in Read)
OLAP
(Read Efficient

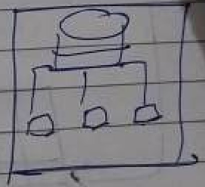
OLTP → ETL → OLAP.

Extract Transform Load

ETL :-







Extract

Transform (Pipeline Creation)

Load

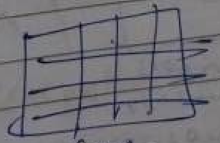
↓

OLTP

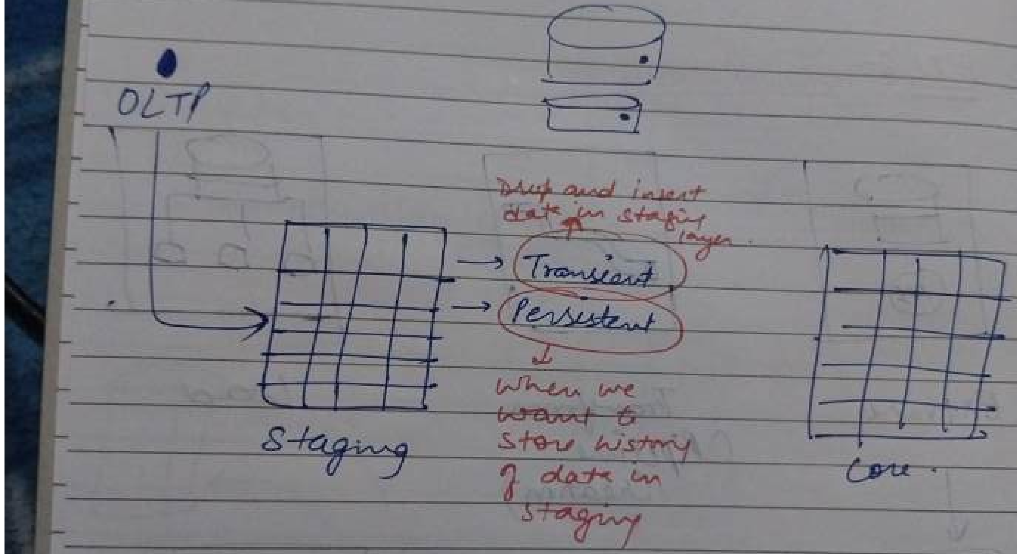
OLAP

Data Warehouse

Data Warehouse and Layers :-



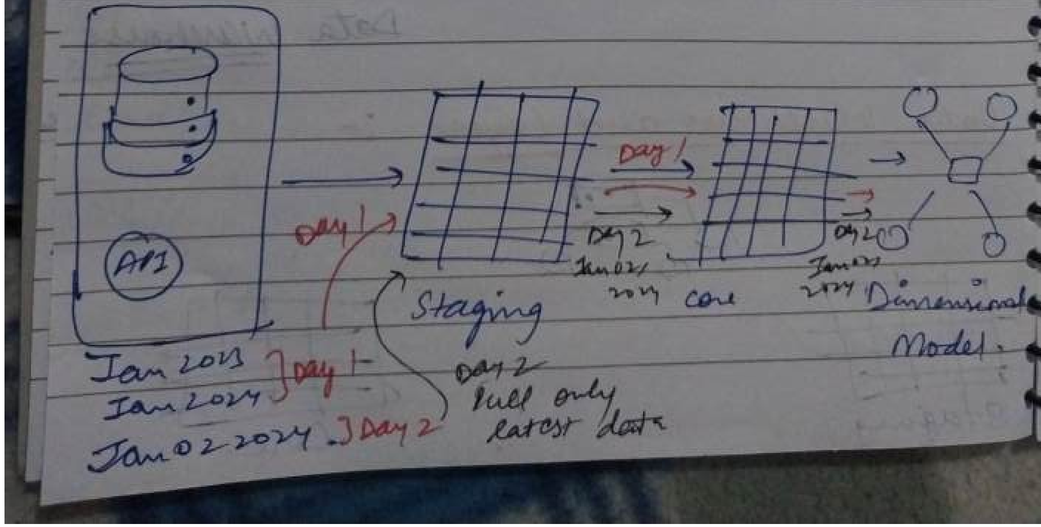
Core



Drop and insert data in staging layer.

When we want to store history of data in staging

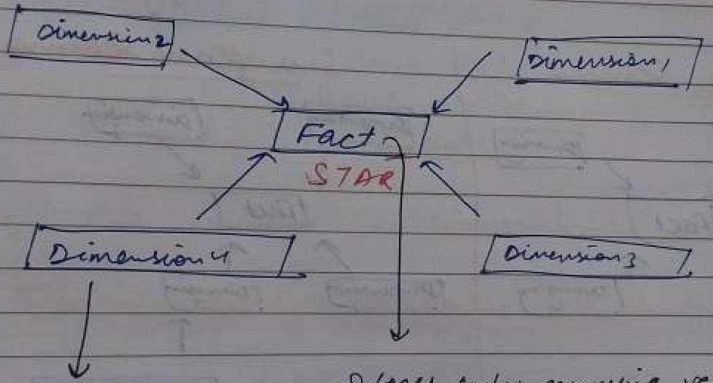
Incremental loading :-



Jan 2023 } Day 1 -
Jan 2024
Jan 02 2024 . } Day 2

Day 2
Pull only latest data

Dimensional Modelling :-



Stores only numeric values.

Rest of things will go to dimension

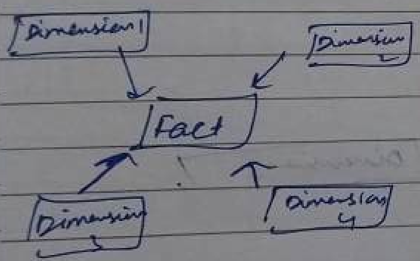
Ex : Dimension 1 :- Details of customers. (Customer Dimension)

Dimension 2 :- Details of Product (Product Dimension).

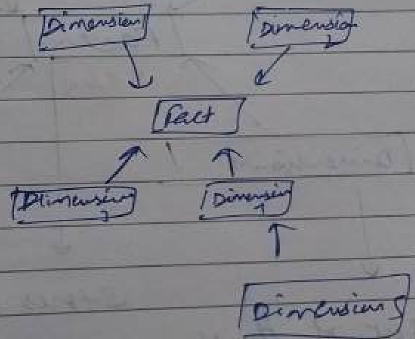
→ Data is Denormalised/less normalised in Fact/Dimension tables.

Dimensional Modelling

Star Schema



Snowflake Schema



// Heirarchy of Dimensions

Slowly changing Dimensions :-

1. Type - 0
2. Type - 1
3. Type - 2
4. Type - 3

Type-0 Slowly changing Dimensions :-

Product_ID	Name	Prod_cat
1	Honey	Food
2	Shirt	clothing
3	Comb	clothing

// No change in Dimensional Table.

Type-1 slowly changing Dimensions :-

↓

UPSERT (update + insert)

Product_ID	Name	Prod_cat
1	Honey	Food
2	Shirt	Clothing
3	Comb	clothing
4	Shoes	Footware

Before

Update → 1

Update → 2

Update → 3

Insert → 4

Product_ID	Name	Product_cat
1	Honey	Food
2	Shirt	clothing
3	Comb	Hair
4	Shoes	Footware

After

CHANGE HISTORY IS NOT KEPT IN THIS DIMENSION

Preserving history.

Type - 2 Slowly Changing Dimension :-

Before

Product_ID	Name	Prod_Cat	Eff StartDate	Eff EndDate	Inuse
1	Honey	Food	1/1/2024	1/1/3000	Yes
2	Shirt	clothing	1/1/2024	1/1/3000	Yes
3	Comb.	clothing	1/1/2024	1/1/3000	Yes

Add 3 more columns to handle change history.

1. Effective start date
2. Effective End date
3. Inuse.

Comb Prod_Cat is changed to Hair.

After

Product_ID

1
2
3
4

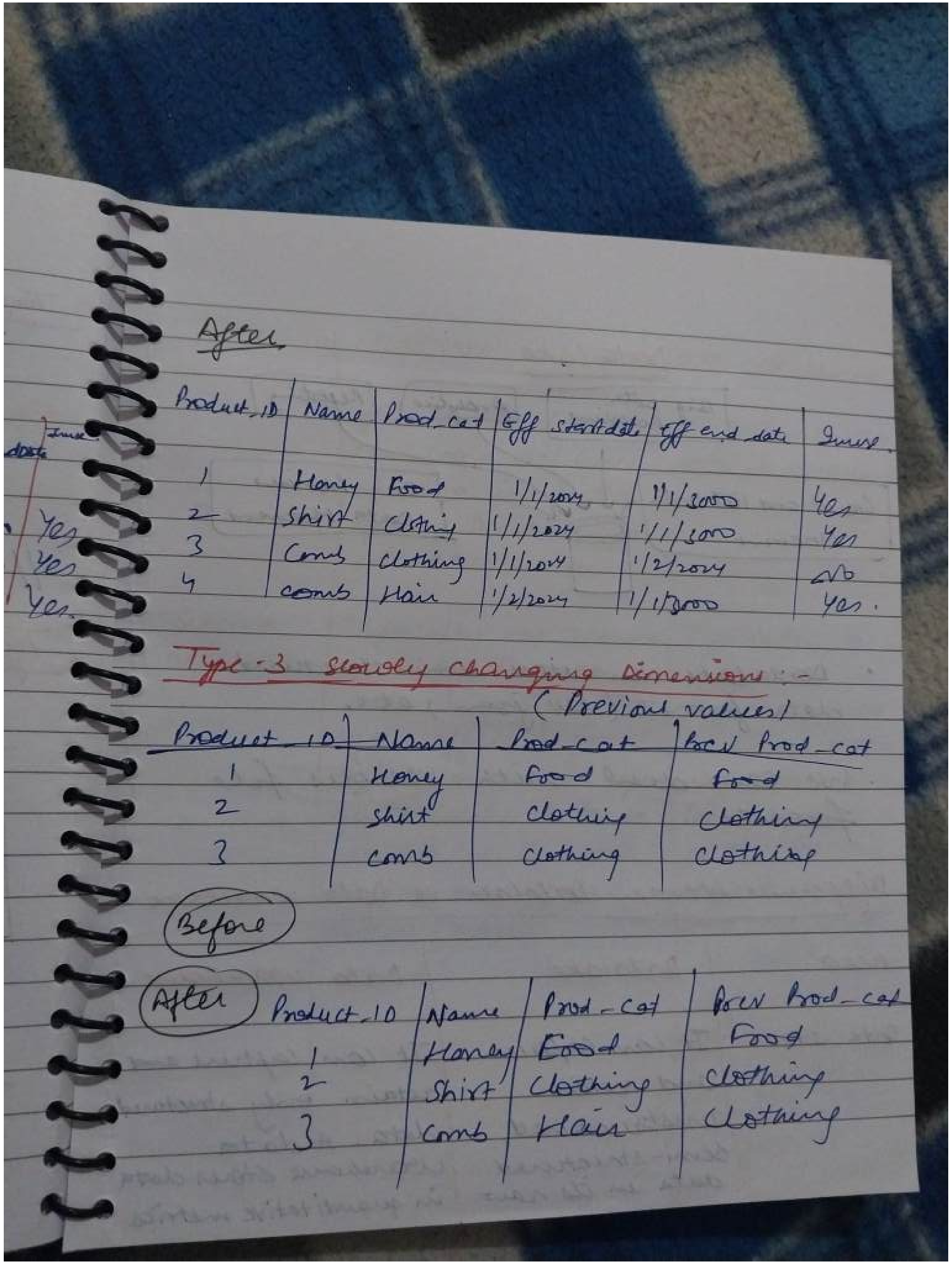
Type -

Produ

1
2
3

Befo

After



Iscurr
date

Yes
Yes
Yes.

After

Product_ID	Name	Prod_cat	Eff startdate	Eff end date	Iscurr.
1	Honey	Food	1/1/2024	1/1/3000	Yes
2	Shirt	Clothing	1/1/2024	1/1/3000	Yes
3	Comb	Clothing	1/1/2024	1/2/2024	No
4	comb	Hair	1/2/2024	1/1/3000	Yes.

Type-3 slowly changing dimensions :-

(Previous values)

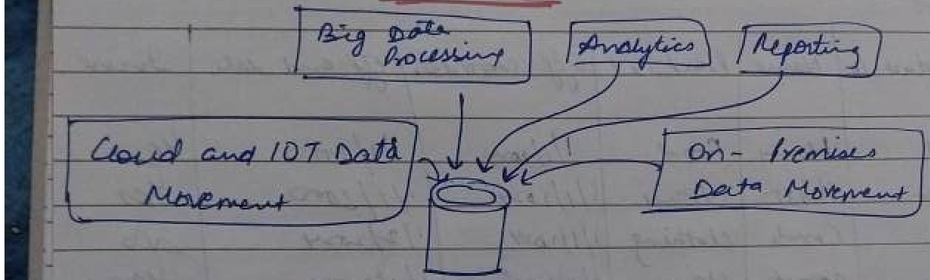
Product_ID	Name	Prod_cat	Prev Prod_cat
1	Honey	food	food
2	Shirt	Clothing	Clothing
3	comb	Clothing	Clothing

Before

After

Product_ID	Name	Prod_cat	Prev Prod_cat
1	Honey	Food	Food
2	Shirt	Clothing	Clothing
3	Comb	Hair	Clothing

Data Lake



- Datalake can store semi-structured data formats (csv, json, etc.

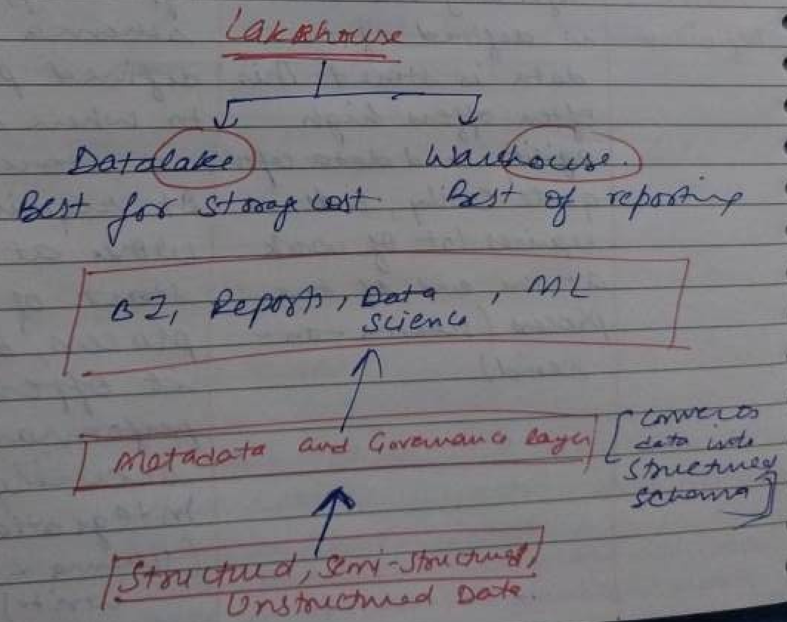
- We can deal with various file formats.

Difference between Datalake vs Data warehouse

Area	Datalake	Data warehouse
Data store.	It can capture and retain unstructured, semi-structured data in its raw	It can capture and retain only structured data. A data warehouse stores data in quantitative metrics

Area.	Datalake	Data warehouse.
	format. A datalake can store all types of data irrespective of the sources and structure.	with their attributes Data is transformed and cleansed.
Schema Definition.	Typically the schema is defined after data is stored. This often offers high agility and data capture quite easily, but it requires lot of work at the end of the process (schema-on-read)	Typically, a schema is defined prior to when data is stored. It requires work at the start of the process, but it offers performance, security and integration (schema-on-write)

Area	Datalake	Data warehouse
Price and Performance	The storage cost is relatively low, Compared to data warehouse and querying result is better.	The storage cost is high, and querying results is time consuming.



Lakehouse
Datalake
Best for storage cost
Warehouse.
Best of reporting
BI, Reports, Data science, ML
Metadata and Governance layer
[Converts data into Structured schema]
Structured, Semi-Structured, Unstructured Data.

File-

① Row-
② Colu

Row-Ba

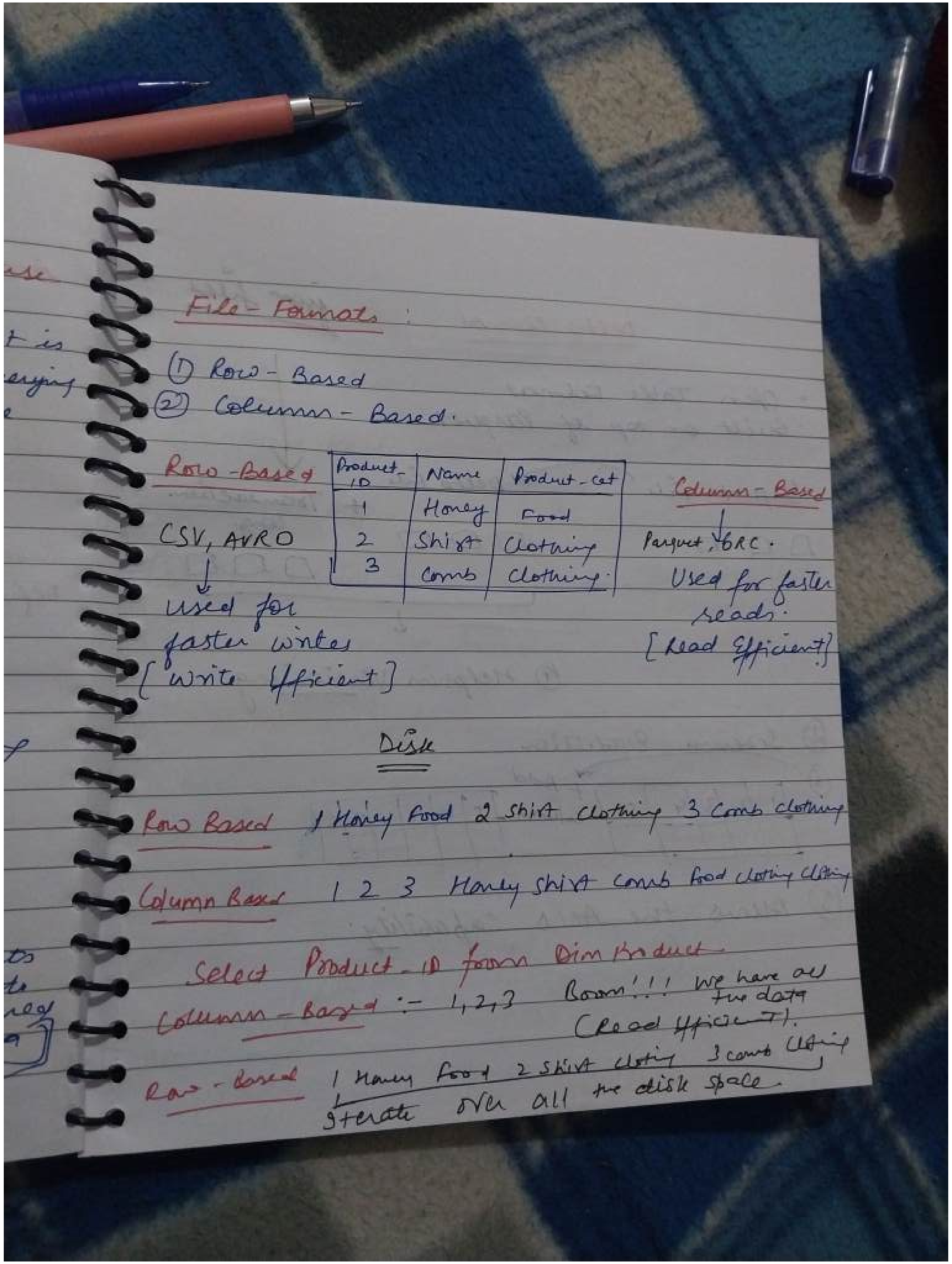
CSV, AVR
↓
used f
faster
{write

Row Based

Column Bas

Select
Column

Row-bas



File-Formats :

① Row-Based

② Column-Based.

Row-Based

Product_ID	Name	Product_cat
1	Honey	Food
2	Shirt	Clothing
3	Comb	Clothing.

Column-Based

CSV, AVRO
↓
Used for faster writes
[Write Efficient]

Parquet, ORC.
↓
Used for faster reads.
[Read Efficient]

Disk

Row Based   1 Honey Food 2 Shirt Clothing 3 Comb clothing

Column Based   1 2 3 Honey Shirt Comb food clothing clothing

Select Product_ID from Dim Product

Column-Based :- 1,2,3   Boom!!! We have all the data
(Read Efficient).

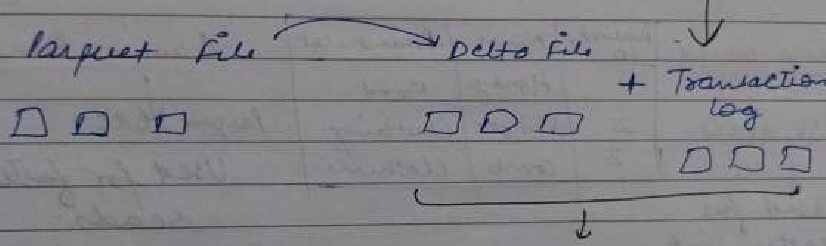
Row-Based   1 Honey Food 2 Shirt clothing 3 comb clothing
Iterate over all the disk space.

Delta Format

json files

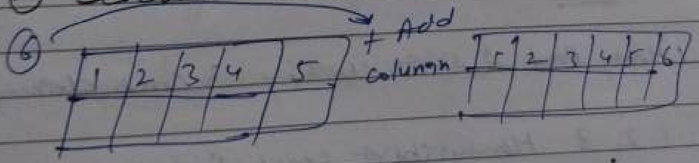
- Open Table Format
- Built on top of Parquet.

Parquet File → Delta File + Transaction log



① Helps in Versioning

② Schema Evolution



③ Allows the ACID Capability.

Big Da

① Apac

② Apac

③ Apac

④ Da

Distri

Ask
we c

Node=
M

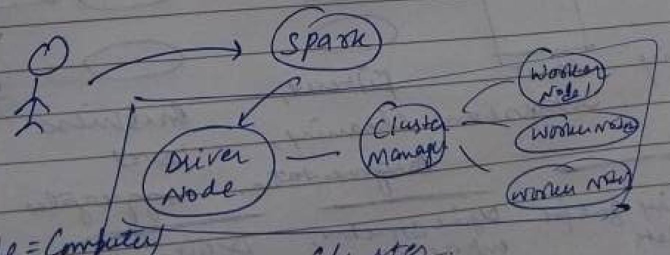
Big Data

Big Data frameworks:-

1. Apache kafka → Used for streaming Data, Real Time Data
2. Apache Airflow → Used for Data orchestration
3. Apache spark → "Discuss in future"
4. Databricks → Processing by using cluster

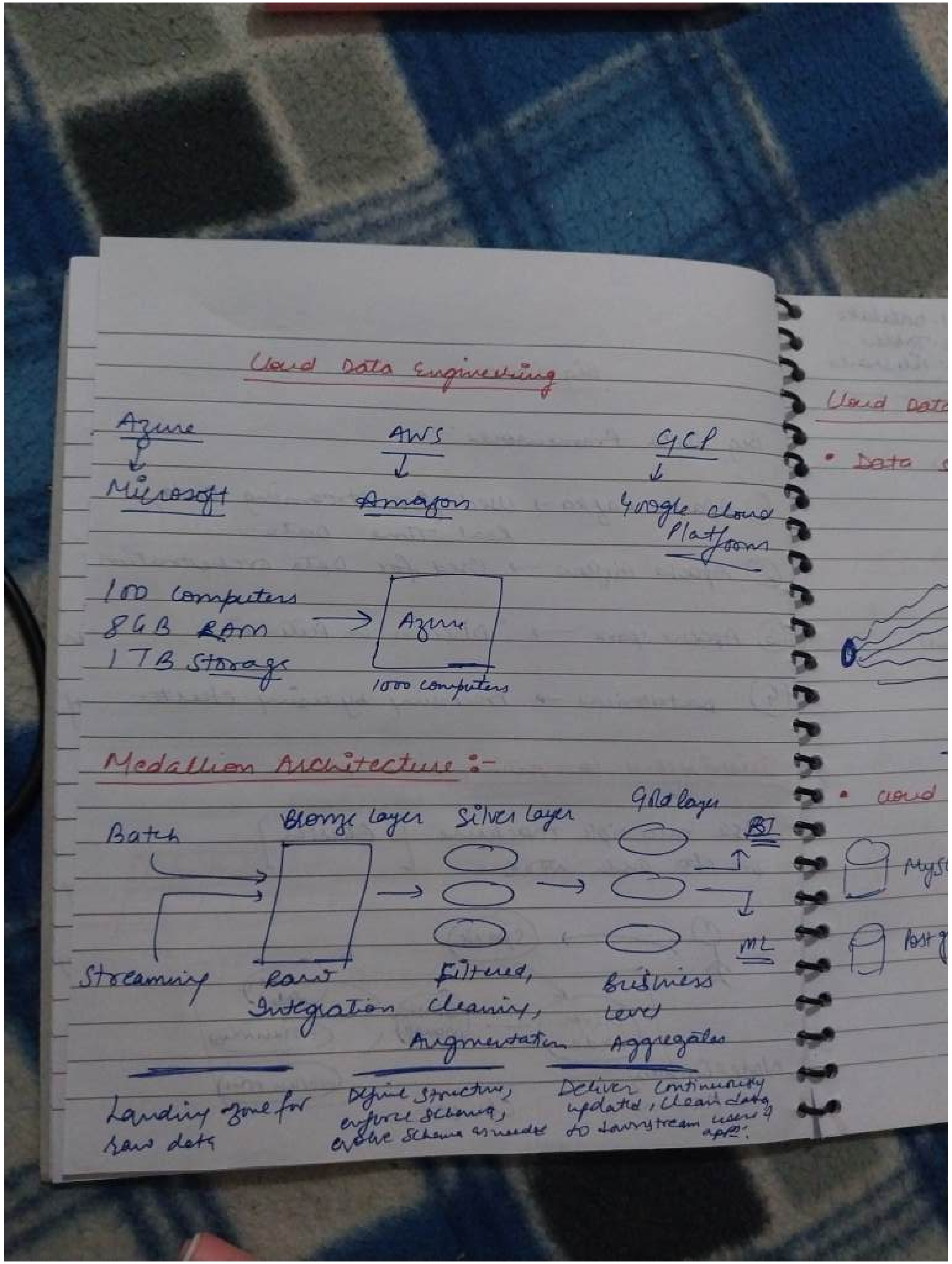
Distributed computing with spark:-

Ask multiple machine [Cluster] we do our work



Node = Computer Machine

Cluster.



Cloud Data Engineering

Azure → Microsoft

AWS → Amazon

GCP → Google Cloud Platform

100 computers
8GB RAM
1TB Storage
→ Azure
1000 computers

Medallion Architecture :-

Batch

Streaming

Bronze layer

Raw Integration

Silver layer

Filtered, Cleaning, Augmentation

Gold layer

Business level Aggregates

BI

ML

Landing zone for raw data

Define structure, enforce schema, evolve schema as needed

Deliver continuously updated, clean data to downstream users & apps.

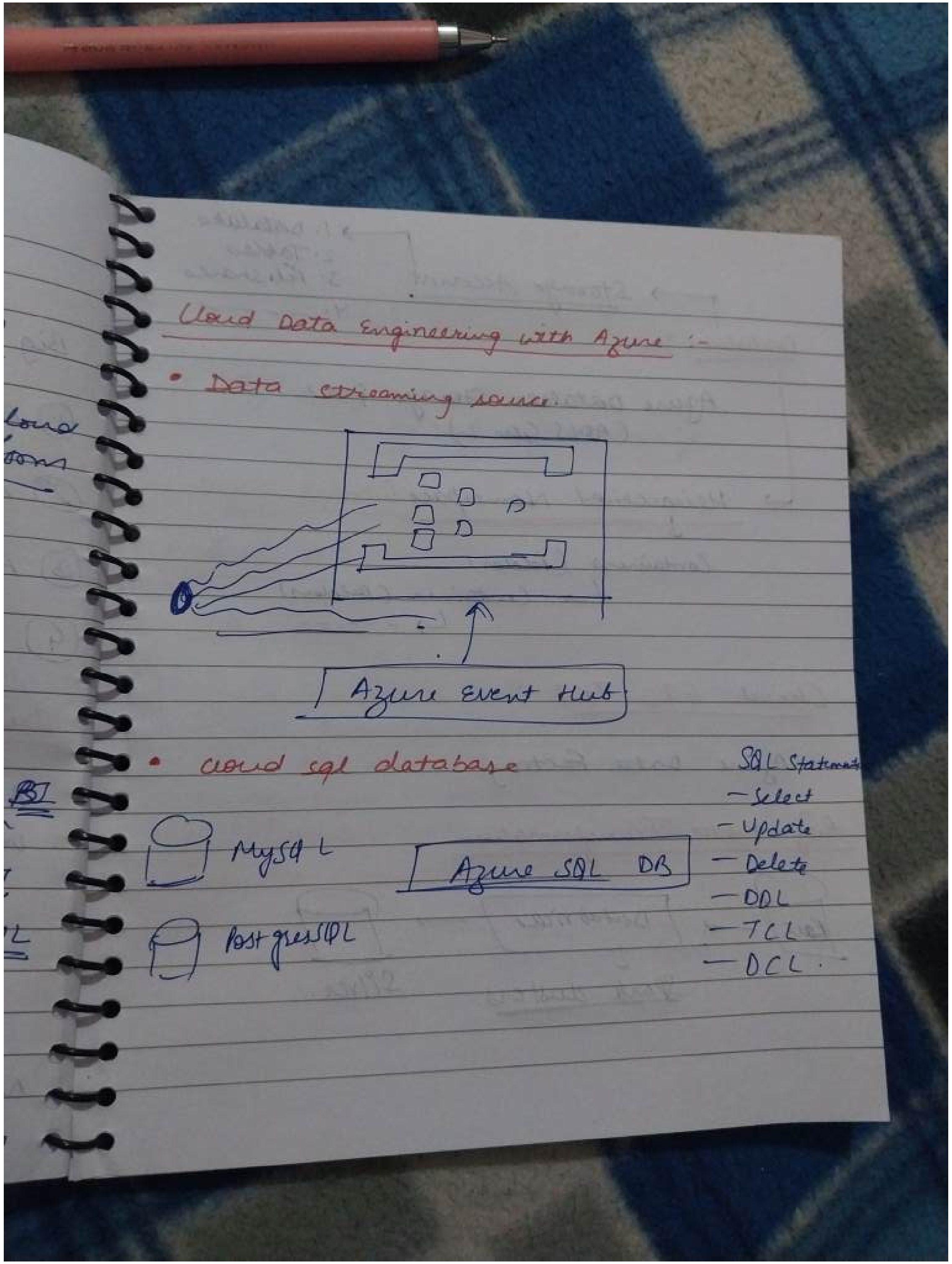
Cloud Data

• Data s

• Cloud

MySQ

Postg



Cloud Data Engineering with Azure :-

- Data streaming source

- cloud sql database

Azure SQL DB

SQL Statements

- Select
- Update
- Delete
- DDL
- TCL
- DCL.

→ Storage Account → 1. Datalake
2. Tables
3. File shares
4. ----

Datalake :-

Azure Datalake Storage Gen 2
(ADLS Gen 2)

→ Heirarchical Namespace.

↓

Containers (Folder)

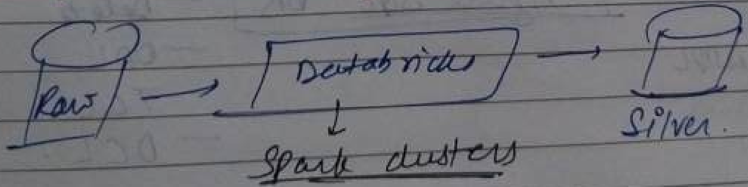
↳ Container (Folders)

↳ ----.

Cloud ETL Tool :-

- Azure Data Factory.

Big Data Transformation :-



ake
s
ores

Cloud Data warehouse :-

- Azure Synapse Analytics
  Ex : Snowflake, Redshift.

Data Reporting

- Power BI

Imp Terms

- Azure Purview :- Data Governance Tool.
- Azure Devops :- CI/CD.
- Azure Key Vault :- Store confidential information
- Microsoft Entra ID :- Records all the users.
- Azure Monitor :- Monitor Production Runs.
- Cost Management :-