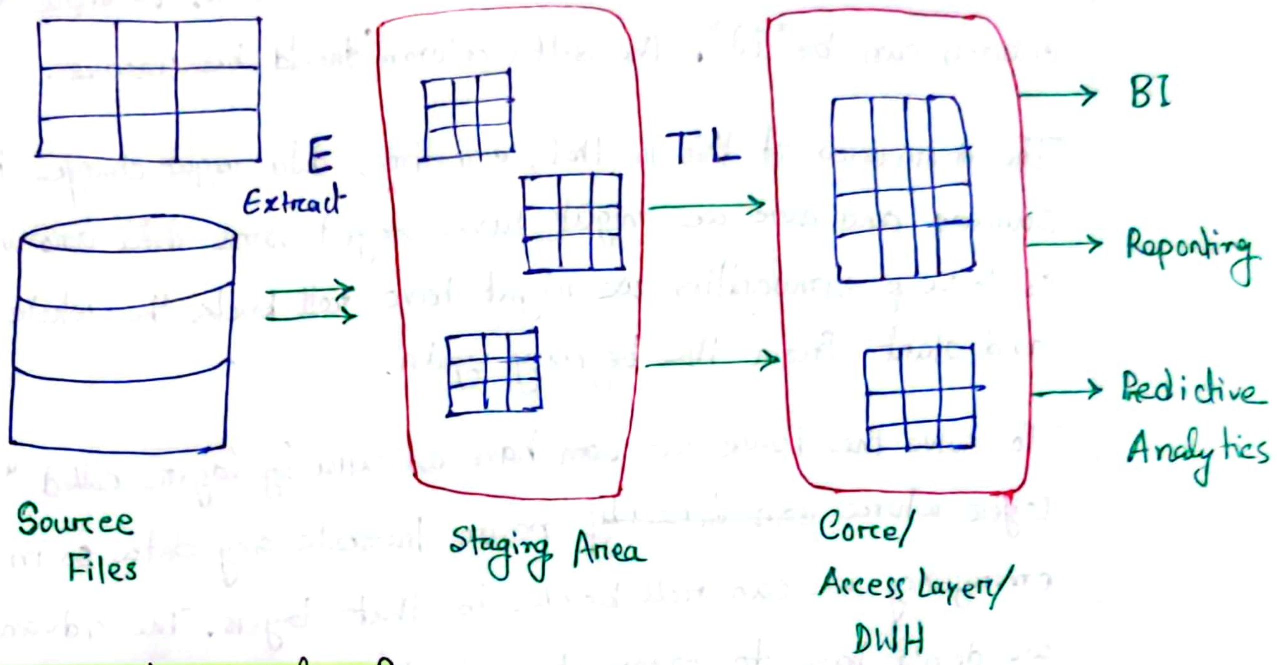
TOPIC NAME : DWH Architecture

DAY :

TIME : DATE : / /



Why we need staging layer?

When we extract data from the operational systems, we should not engage with those systems for a long time which can make them performing low. To avoid this we have to engage with them as quick as possible.

Then in the source files data can be in different format like csv, xlsx, NoSQL and many more. We extract the data and store them in relational database table format in the staging area.

Then in staging area, where everything is in table, we can start to define our transformation and apply them using ETL.

After every ETL operation we will truncate our staging layer, so that we can extract newer data from the sources.

TOPIC NAME : DAY : TIME : DATE : / /

We use delta column to check for newer data. Example of delta column can be "id". The delta column should be unique.

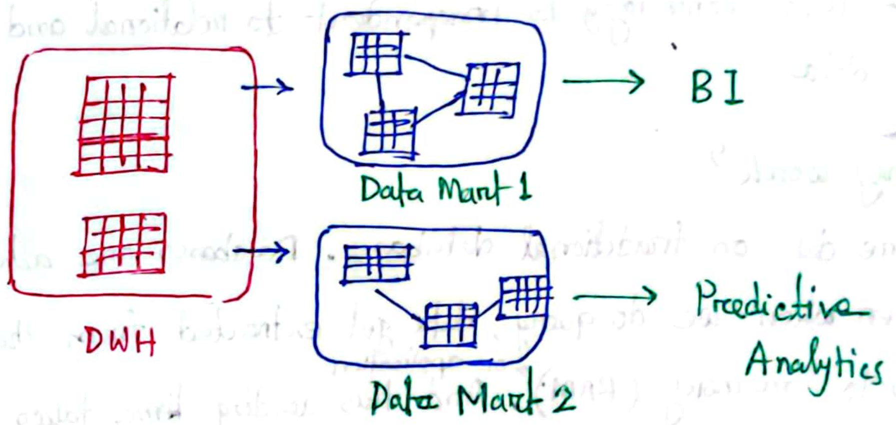
The drawbacks of this is that, sometimes data might change in the sources and also we might have staged some data as well. So to keep similarities we might have roll back the whole process and start from the begining again.

To solve this issue we can have an extra ~~ly~~ layer called "Persistant" layer where ~~can directly~~ never truncate any data so in case of emergency we can roll back to that layer. The advantage is we don't have to interact with the operational systems and lower their performance.
But rarely we have to use it.

Summary:

- Staging layer is the landing zone of extracted data
- Data are in tables or in seperate database
- As little "touching" as possible
- We don't change the source systems
- Temporary or Persistant layers have to be added if need.

Data Marts: Data marts are subset of DWH. Sometimes for different use case we don't need all the tables/databases from DWH. In that case we further make additional layers extracted from DWH where we keep only those table which will solve our needs.



These data marts are the same as power BI or QlikSense data model that we use.

Summary:

- Subset of DWH
- Dimensional Modeling
- Can be further aggregated
- Usability + Acceptence
- Better performance
- Used under BI tools

TOPIC NAME : DAY : TIME : DATE : / /

In memory databases :

- Highly optimized for query performance.
- Good for Analytics / High Query ~~performance~~ volume
- Usually used for data marts.
- This technology is independent to relational and non relational data

How they work ?

We store data on traditional database. Databases are actually HDD, SSD etc. Then when we do query, data get extracted from those disks and load into memory (RAM) $\xrightarrow{}$ or application. And this loading ~~time~~ takes time (depends on how much data has been asked for or how complex the query is)
As response time is greater here, it's not a high quality optimal performance.
To solve this, in memory databases are built. It eliminates the response time coming from the disk and load data in memory. For that, we get a much better query performance.

Different algorithms and tools are there for it.

- Columnar storage
- Parallel query
- many others

TOPIC NAME : DAY : TIME : DATE : / /

Problems can be faced in in-memory storage too.

① Durability: Lose all information when device loses power or get reset

② Cost factor: This technology is more costly.

Examples of in-memory databases:

① SAP Hana

② MS SQL- SERVER in memory Tables

③ Oracle - in memory table

④ Amazon memoryDB.

OLAP Cubes:

In a cube, data is not organized in tables with relations, but in a non relational way and into dimensions. So we can have multiple dimensio in an OLAP cube. Data in those cubes are not organized into rows and columns, rather in arrays.

Main reason to use: Fast query performance.

→ Works well with many BI solutions

The language it ~~is~~ uses → MDX (MultiDimensional expression)

→ Developed by Microsoft

TOPIC NAME : DAY : TIME : DATE : / /

Some reecommendation while using cubes →

- Build for a specific use case
- Try to minimize dimensions as much as possible

Advantages:

- Built for specific use case
- More efficient and less complex with seperate data marts.
- Good for interactive queries with hierarchies.

Disadvantage:

- Little bit complex and technical
- Less important today with advancement of hardware.

Operational Data Storage (ODS)

The difference between DWH and ODS is ; ODS is used for operational decision making. It is not used for analytical or strategic decision making which DWH is used for.

In ODS →

① We make quick operational decisions

② : We need no long history of data. Only need the current state data (real time data)

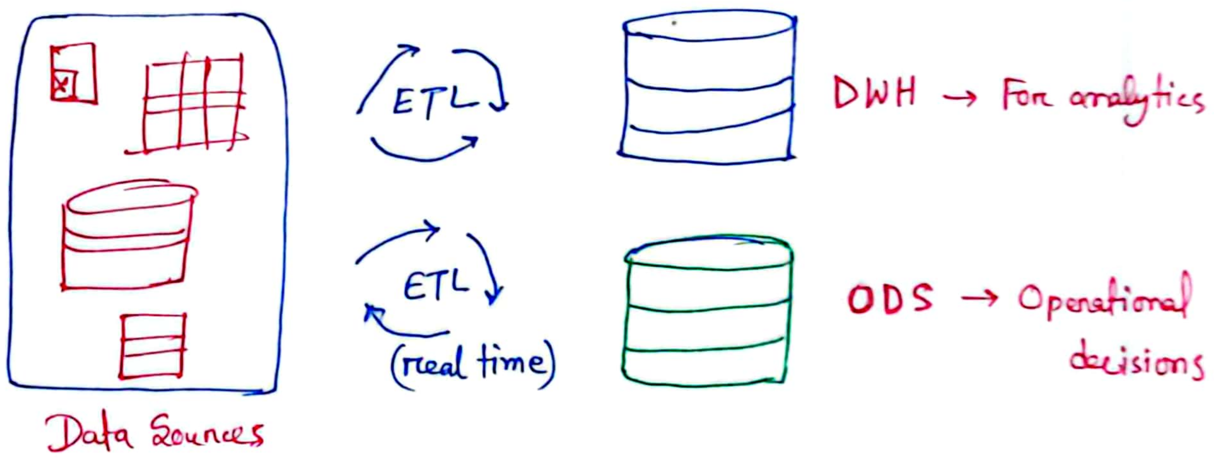
③ Transaction systems (Like Bkash, Nagad)

TOPIC NAME : DAY : TIME : DATE : / /

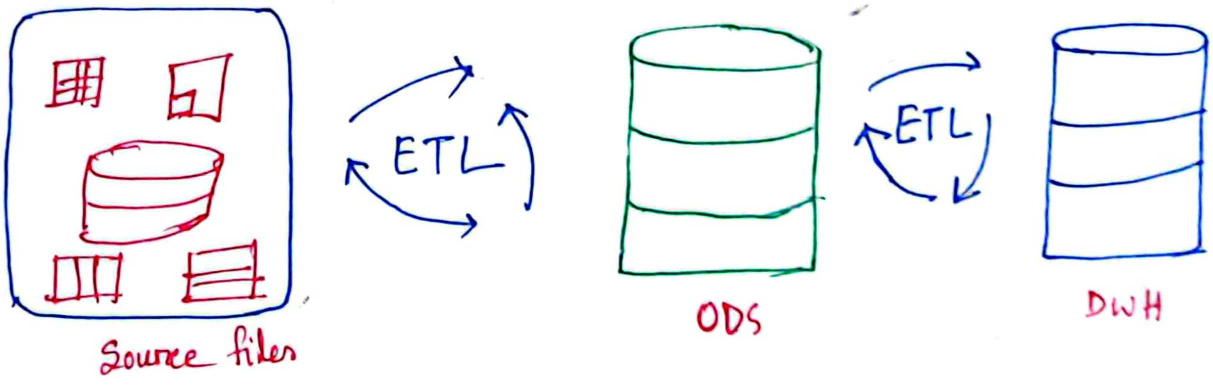
- Any platform where we ~~can~~ provide offers/discounts based on the customer's immediate transaction data.

So, we can have and keep both ODS and DWH in our company as both of their goal and ~~work~~ purpose is really very important for a company.

Parallel integrations:



Sequential Integration:


ETL
ETL
Source files
ODS
DWH