Section 1 : Choosing the right database

Redis  :-  Cache

BLOB : s3 + CDN

Text search : elastic search + solr

Time series d/b :- open tsdb

**In system** design how  good it is design that very much depend on database we use

Database does not effect the functional requirement

Non functional requirement are impacted by the database

**Normally choice of database depends on couple of factors**

**1-**structure of data we have whether it is structure or non-structure data

2- query pattern we have

3- amount of scale we need to handle

**Point 1- Caching solution :** whatever the system we design we have to use some caching solution

There could be lot of use case for caching

Ex:-

key :- query param

Value : response will get from api call

**Redis**---- is the solution for caching

**Memcache**

**Hazlecast**

**Point 2-      File storage system**

Note :

Where we have **videos and image**need to store there is something called **blob storage**

**These** are not d/b. d/b means where we can query on.

Now there are lot of providers for that

**One is common one :--- amazon s3**

**CDN (Content delivery network)**

**CDN** is use for distributing the  same images geographically to lot of places

**Point 3-    Requirement :**

Product is uploaded in amazon and want  user to search that using some title or text

Or want to search movie name using netflix

Or uber where we want text searching capability

So all of these type of use cases we usually use **text search engine**

Provided by **elastic search and solr** and both are build on top of **apache lucene**

**What is Apache Lucene used for?**

Apache Lucene™ is a high-performance, full-featured search engine library written entirely in Java. It is a technology suitable for nearly any application that requires structured search, full-text search, faceting, nearest-neighbor search across high-dimensionality vectors, spell correction or query suggestions

**Note**:

Elastic search and solr are not d/b. These  are search engines

The diff b/w search engines and d/b that:

d/b give guarantee that data in d/b wouldn't  be lost. But search engine not guarantee

So primary source must be d/b and could load data in elastic search for search engine

**Graphana and Prometheus**  : application metrics tracking system

Here will do sequential update and not random update

But while in d/b we can do random update

**Point 4- What is Time Series database?**

A time-series database (TSDB) is **a computer system that is designed to store and retrieve data records that are part of a “time series,” which is a set of data points that are associated with timestamps**. The timestamps provide a critical context for each of the data points in how they are related to others.

**Point –5 Why is analytics used?**

Why is data analytics important? Data analytics is important **to understand trends and patterns from the massive amounts of data that are being collected**.

Ex :

Analytics like how many order i m having

So where want  to do analytic on data for the whole company there we something called data warehouse

That is basically a large d/b where we dump all that data and  and provides various queryinf capability on the top of data to serve lot of report. These are not generally use for transaction

System. These are generally use for offline reporting.

Then we will use **hadoop : where** we put lot of data from various transaction system and then build lot of system that will provide reporting on top of that data

**Point 6 -Choose between relational and non relational d/b**

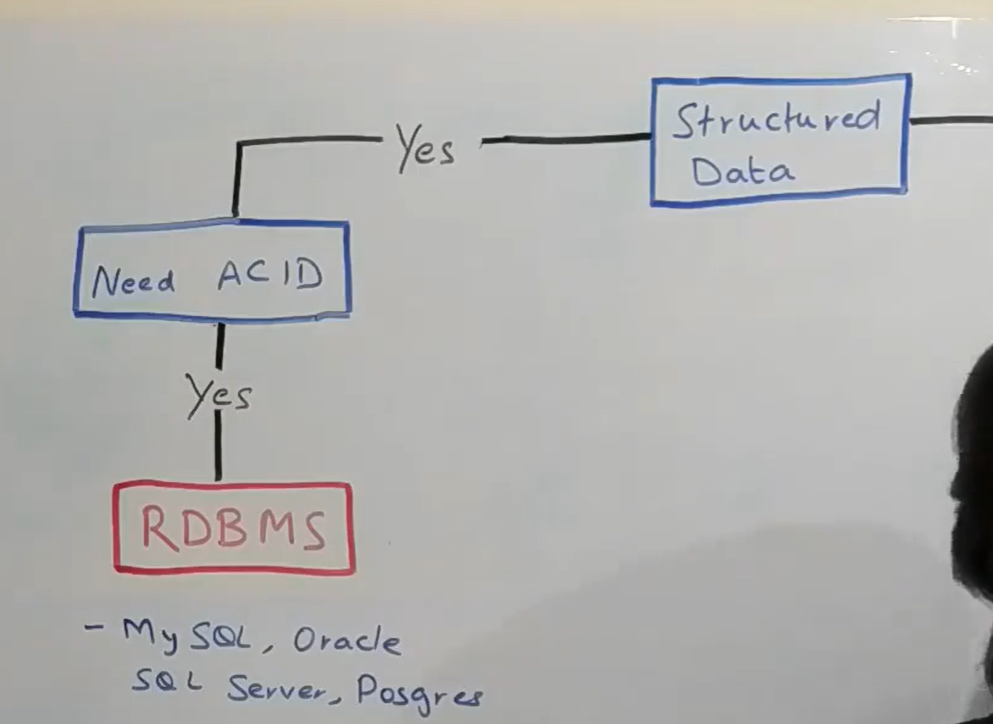
 If we have very **structured info then will use relational** d/b

**Structure info :** it would be an info that could be easily model in form of table.

Table would have row and column of information

**Ex:** if we want to store of user info in any social n/w like name email etc

Like in payment system we need ACID property so we should use relational d/b but there are multiple providers for that.



Now we dont have acid property but still it is relational data then we can use both relational and non relational d/b

**Now let assume we do not have structured data and making catalogue system like e commerce where all the item are available**

Where lot of data is there and wide range of querying is there like

**In ecommerce site there are lot item and each item would have certain attribute**

**Shirt :-** size, colour

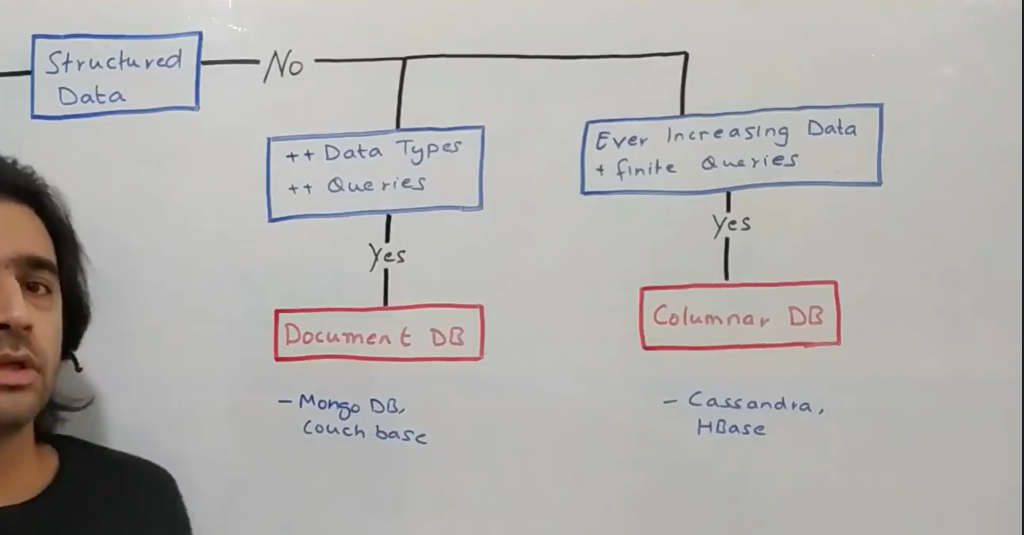
**Refrigerator :-**volume, power

**Milk**: quantity and expiry data

These are d/b where we have lot of data not in terms of volume but in terms of structured.

Lot of attributes coming and wide variety  of query coming then there is something we should use **document db**

**There** is lot of providers  for document db:-**mongo db**



Now let say data is non relational and also we dont have complex query

**If we have less query but large amount of data**

Example

Uber driver

Query : tracking

Data :- increasing daily as uber driver increasing

Then we can use**columnar d/b**

**Cassandra , hbase**

Suppose there is only one item left and 10 users are there to buy  for inventory management system in **amazon** then it should follow ACID property and we should use relational d/b

**Data of amazon is ever increasing**

**What it means :-** it can use combination of d/b