Lambda Architecture Assignment

1. Explain the factors leading to Big Data, List and Explain major sources of Big data?

-> The rise in technology has led to the production & storage of large amounts of Data. & Affordibility of Smart phones.

* Earlier MetaBytes of data were used but now a days PetaBytes of data are used for analysing, discovering new facts & knowledge.

* Traditional Systems were not built to handle large amount of data so there is a need for new Systems to analyse

* The Big Data is a high-volume, high-velocity, high-variety & a high veracity in formation set.

The Major Sources of Big Data are:-

i) Social Networks & web Data:- Such as Facebook, Twitter, Emails blogs & Youtube on an average per day Billions of Bytes of data are Produced by Social Networking Sites making it a large contributer of Big Data.

* Transaction data & Business Processes:- Such as credit card, transactions, flight bookings etc. & medical records

* Increased Data Generation:. As activities like online shopping, Social media produce more data generation.

* Devices and Sensors continuously collect more and transmit data.

* E-commerce & Transaction data:.

online shopping transactions, payment processing data.

* Health Care and Medical Records:-

EHR's & medical imaging

* Telecommunications:- call records, network traffic data

* Government and public records:. census data, public health records

② List and Explain the characteristics of Big Data

→ Big data is defined by several key characteristics that distinguish it from traditional data. these characteristics are often summarized by the ~~five~~ 3 v's, Given below are the characterics list

1. Volume :- It represents the big size of the data. it comes from large piece of data or collection of small data over a period of time.

* The name Big Data itself is related to a size which is enormous.

Volume is a huge amount of data to determine the value of data, Size of data plays a crucial role.

* If the volume of data is very large, then it is actually considered as a 'Big Data'.

Ex: Facebook can generate approximately a billion messages 4.5 billion times that the 'Like' button is recorded, & more than 350 millions new posts are uploaded each day.

2. Velocity :-

* It refers to the high speed of accumulation of data.
* In big data velocity data flows in from sources like machines, networks, social media, mobile phones etc.
* There is a massive & continuous flow of data.
* It contains the linking of incoming data sets speeds, rate of change & activity bursts.

Veracity:- It means how much the data is reliable it has many ways to filter & translate the data veracity is the process of being able to handle & manage data Efficiently.

Ex:- Facebook post with hashtags.

Variety:- Big data can be structured, unstructured & semi. structured that are being collected from different sources.

* In Earlier the data will only be collected from databases & sheets.
* But these days the data will comes in array forms, that are PDFs, Emails, Audios, Photos.

③ List and Explain the major challenges of Big Data systems.

⇒ ① Data Security & Privacy:- ~~Security~~

Security is a big concern for Organizations. Non-Encrypted information is at risk of theft or damage by cyber-criminals. therefore, data security professionals must balance access to data against maintaining strict security protocols.

* Data Quality Issues:- Data is the foundation of data analytics, & its quality directly impacts the insights derived.

* Missing information can lead to skewed results & hinder effective analysis

* Errors in data entry or inconsistencies can lead to misleading conclusions.

*

* Ethical and Bias challenges:- Data & algorithms can inherit biases from the real world, leading to unfair or discriminatory outcomes. here's why it's crucial to be vigilant.

* Algorithms trained on biased data can perpetuate those biases in their outputs.

*

④ Discuss the desired/required properties for Big Data systems

→ Big data systems must meet certain properties to efficiently handle the challanges posed by the large volume, velocity, variety and veracity of data. here are few required properties.

1. Scalability:- Big data systems must be scable to handle the Ever-growing volumes of data. Scalability can be horizontal (adding more machines or nodes to a system) or vertical (adding more resources to Existing model)

2. Fault Tolerance and Robustness:- Big data systems need to be fault tolerant, meaning they should continue to work Even if some components fail. Reliability Ensures that data processing is accurate and consistent, Even after hardware failure.

3. Low latency and High Throughput:- latency is the time delay between data input and the corresponding output. low-latency systems can provide fast response or quires or data processing that High throughput refers to the system's ability to process a large amount of data at a given time.

4. Generalization:- A general system can support a wide variety of applications based on functions of all data, it generalizes to all applications like finance management systems, social networking or computer analysis etc.

5. Extensibility :- Extensible systems allow functionality to be added with a minimal development cost.

Ex:- change / add of a new future requires a migration of old data into a new format but as big data systems are Extensible it is easy and quick to do big migrations.

6. Adhoc Quiries :- Being able to do adhoc quiries on your datasets has increased opportunities for buisness optimization and new applications.

7. Minimal Maintainance :- Maintainance is the work required to keep a system running smoothly. this includes anticipating when to add machines to scale, keeping process up and running, debudding, anything that goes wrong in production.

8. Debuggability :- A big data system must provide the information necessary to debug the system when things go wrong. the key is to be able to trace for each time & value in the system.

⑤ Discuss the problem faced by traditional database systems

i) Scalability Issues:- Traditional databases struggle to scale horizontally, meaning they are designed to run on a single server or cluster of tightly-coupled server.

ii) challenges with Fault Tolerance:- Ensuring fault tolerance in traditional databases, especially in distributed systems can be complex, requiring mechanism like replication, failover systems and backups.

iii) Bigdata is too big for traditional storage.

Though in theory traditional database systems can handle large amounts of data. it simply can't keep up with demands & of mordon data to deliver the efficiency & insigh we need.

iv) Big data is too complex for traditional storage.

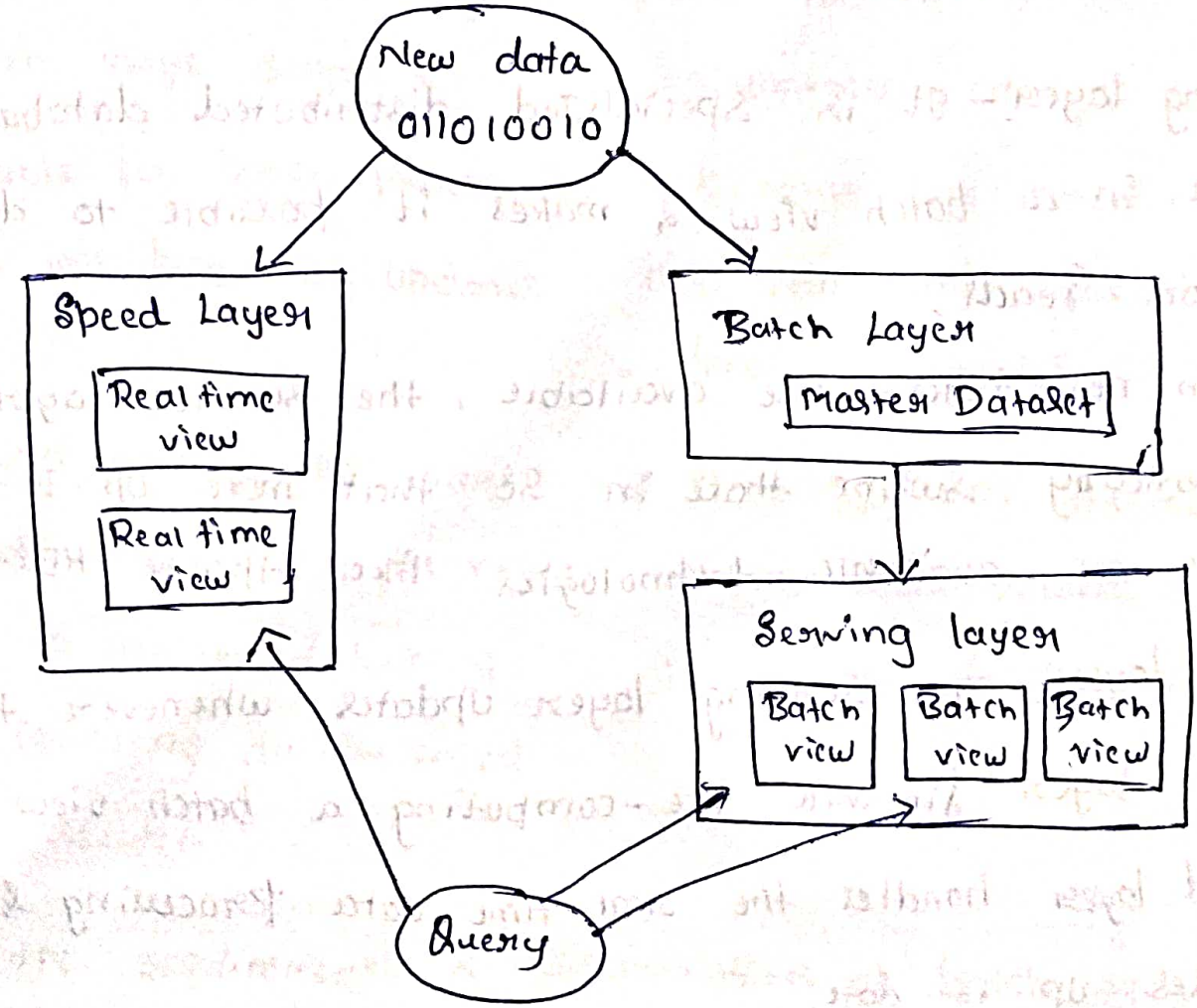
[Though in theory traditional database systems can handle large amount of data. it simply] this is wrong].

Since big data consists of more than rows & columns some may be structured but large part of it is unstructured. which can be an issue as they can properly catagorize it.

v) Big data is too fast for traditional storage.

-> An RDBMS designed for rapid fiucations but Big data grows almost instaneously & anlgris needs to occur in real time. for Ex:- Sensor data. which produce at concurrent rate. which maybe difficult to analyse.

6. Explain different layers of Lambda Architecture.



Batch layer:- The batch layer stores the master copy of the dataset & precomputes batch views on that master dataset which can be thought of as a very large list of records.

-> Batch layer should be able to store an immutable, constantly growing master dataset and,

-> ~~compute~~ This is immutable append-only dataset that stores the Entire historical data.

-> typically using distributed systems like Hadoop or Spark & it produces batch views handles complex computations for dataset analysis.

Ex: Hadoop, Mapreduce, Apache Spark.

Serving layer:- It is specialized distributed database that loads in a batch view & makes it possible to do random reads,

* when new batch's are available, the service layer automatically swaps those in so that more up-to-date results are available. technologies like Apache HBase

Speed layer:- The serving layer updates whenever the batch layer finished pre-computing a batch view.

* Speed layer handles the real-time data processing & provides up-to-date.

realtime view = function (real time view, new data)

batch view = function (all data)

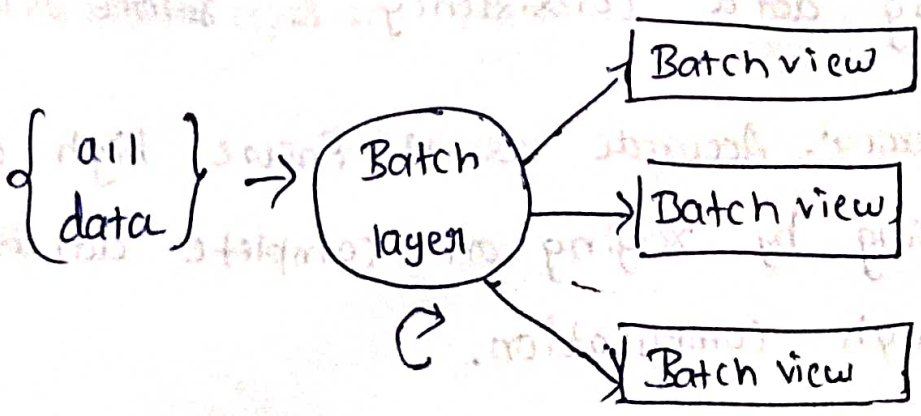
realtime view = function (realtime view, new data)

query = function (batch view, realtime view)

⑦ Differentiate b/w re-compution algorithm and increment algorithm

re-computation algorithm	Increment algorithm
* It re-computes the Entire dataset from Scratch	* It updates only the parts of the data that changed
* Lower Efficiency compared to Increment	* Higher Efficiency compared to re-computation
* High latency, Especially with large datasets	* Low latency, as only incremental-changes etc
* Suitable for batch processing and less frequency updates	* Suitable for real time or near real time systems with frequent updates.
* It will update the count by first appending the new data to master dataset & then counting all the record from Scratch	* It will count the number of new data records & add it to the Existing count.

⑧ List the requirements & responsibilites of batch layer



Requirements of Batch layers

i) Scalability:- It should be horizontally ready to handle massive volumes of data as more data is added over time.

Ex:- Apache hadoop & Apache Spark.

ii) Fault tolerance:- It should be designed to recover Gradually from failures.

iii) Performance:- The System should Efficiently perform large Scale data computations.

iv) Data Integrity:- Ensuring that the results produced are consistent & accurate Even in the face of failures

Responsibility of the Batch layer:

i) Data Processing:- Batch layer processes large volume of data in batches, it is designed to handle comprehensive computations & generate results.

ii) Data Storage:- It is immutable storage systems which helps in Ensuring data consistency & ~~reteri~~ reliability

iii) Consistency & Accuracy:- Accurate results Ensure high accuracy in data Processing by relying on complete datasets & Executing through computation.

@ Explain the requirements of serving layer in lambda Architecture.

→ Batch writable: The batch views for a serving layer are produced from scratch. when a new version of a view becomes available, it must be possible to completely swap out the older version.

Scalable:- A serving layer database must be capable of handling views of arbitary size

Random reads:- A serving layer database must support random reads, with indexes providing directories.

Fault Tolerance: The serving layer must be fault tolerant, Ensuring that even in the case of node or system failures, data remains available for querying. It should replicate data across multiple nodes @ regions to Ensure high availability & resilience against failures.

(19) with Example show how low latency & high throughput can be achieved in serving layer of lambda Architecture.

-> i) Low latency in serving layer.

Ex:- Suppose we're building a recommendation system for a e-commerce site, where customers expect product recommendations as they browse the site.

Batch layer:- Batch layer processes historical data such as computed e-commerce site, customer purchase history, browsing history and product details, generating precomputed recommendation models (batch views)

Speed layer:- The speed layer process real-time data, such as the customer's most recent clicks and page views to adjust the recommendations on the fly.

Query workflow for above Ex using all three layers

* A customer lands on the site and starts browsing
* A query is made to the serving layer for their recommendations based on customer IDs, key techniques for low latency are.

~~Precomputed views~~,

Differentiate b/w batch layer and Speed layer.

Batch Layer	Speed Layer
* Process large volumes of historical data to produce accurate comprehensive results	* It Handles real time to provide low-latency, approximate results
* It has Batch processing model	* It has Stream Processing model.
* It has High latency rate	* It has low latency rate
* Provides accurate & complete results	* Produces approximate results due to real-time nature.
* Scales are large dataset but not suitable for real time needs	* Scales to handle high throughput for real time data.
* typically uses distributed file systems like Hadoop	* uses in memory systems like redis or kafka for fast access
* Suitable for complex, large scale queries on Entire datasets	* Suitable for queries real time queries on recent data.

2. High Throughput in the Serving layer.

Ex:- Consider the financial analytics system that recieves millions of queries per second for stock prices and market trends.

Batch layer:- Batch layer processes historical financial data generating the computed views, such as moving averages, Volume trends & stock price aggregations.

Speed layer:- The speed layer processes real-time stock price updates to provide the latest information.

(12) Requirements & responsibilites of speed layer:-

Low latency processing:- The primary requirement of the speed layer is to process data in near real-time, Ensuring minimal delay from when data is ingested to when it can be queried.

Efficient Real time processing Framework:- It should use a stream processing framework capable of performing computations (Eg- aggregations joining) in real time as data flows through the systems.