Q1. LIMITATIONS OF NO SQL:

→ There is no standard that defines rules and roles of no sql database. The design and query languages of NO SQL databases vary widely between different NO SQL products much more widely than traditional SQL database.

→ Backups are drawbacks. Though some databases like mongo DB provide some tools for backup. These tools are not mature enough for complete data backup.

→ When it comes to consistency of data NO SQL doesn't take much consideration so makes it little insecure as compared to relational databases.

ADVANTAGES OF NO SQL:

→ Flexible Data Model - they are flexible as they can store and combine any type of data structured and unstructured.

→ Evolving Data Model - It allows you to dynamically updath the scheme to evolve with changing requirements while ensuring it would cause no interuption.

→ Elastic Scalability - they can scale to accomodath any type of data growth while maintaining low cost.

→ High Performance - In terms of both throughput & latency

→ Open Source - Doesn't require liiencing fees & can run on inexperience hardware.

NOSQL is better database for Big Data because it allows for high-performance, agile processing of information at massive scale. It stores unstructured data across multiple processing nodes, as well as multiple servers. As such, NO SQL has been the solution of choice for some of largest data warehouse.

TYPES OF NOSQL STORES :

(1) DOCUMENT DATABASE - They use key value pairs to store and retrive data from documents. A document is stored in form of XML and JSON
Eg. MongoDB, Apache CouchDB

(2) KEY VALUE STORES - They are simplest type of NOSQL database. It uses key & values to store data. The attribute name is stored in key where as values corresponding to that key will be held in VALUE.
Eg. DynamoDB, Redis, etc..

(3) COLUMN ORIENTED DATABASE - they store the data in set of colums known as column families
Eg. HBase, Cassandra

(4) GRAPH DATABASE - they form & store relationship of data. Each element is stored in a node, that node is linked to another data.
Eg. Facebook, Neo4J

Q2. RDD (RESILIENT DISTRIBUTED DATASET)

It is fundamental data structure of spark. They are immutable distributed collections of objects of any type. As the name suggests is a resilient records of data that resides on multiple nodes.

→ Different ways to create RDDs in spark are:

1. Using parallelized collection
   - basic method to create RDD which is applied at very initial stage of spark
   - It creates RDD very quickly

2. From external datasets
   - this method uses the URL for the file path on machine or database
   - It also reads whole as a collection of lines.

3. From existing Apache Spark RDDs
   - they are immutable
   - this property maintains consistency over cluster

→ Operations that can be conducted on RDDs are:

1. TRANSFORMATIONS : RDD → RDD
   → Examples are map, filter, etc.
   → No communication needed.

2. ACTIONS : RDD → Python object in head node
→ Examples are reduce, collect, count, take, etc.
→ Some communication needed.

3. SHUFFLES : RDD → RDD. shuffle needed
→ Examples are repartition, sortByKey, etc
→ A lot of communication needed.

~~Two operations are as follows :~~

let's assume
val listRdd = spark.sparkContext.parallelize(List(1,2,3,4,5))

1. treeReduce() - It reduces the elements of this RDD in a ~~multiple~~ multi-level tree pattern

Eg. println("treeReduce :" + listRdd.treeReduce(_+_))

2. collect() - It returns the complete dataset as an array

Eg. val data : Array[INT] = listRdd.collect()
data.foreach(println)

Q3. (1) db. residential . find ({})

(2) db. residential . find ({ landmark: "iscon temple" })

(3) db. residential . deleteOne ({ home_id : "123456" })

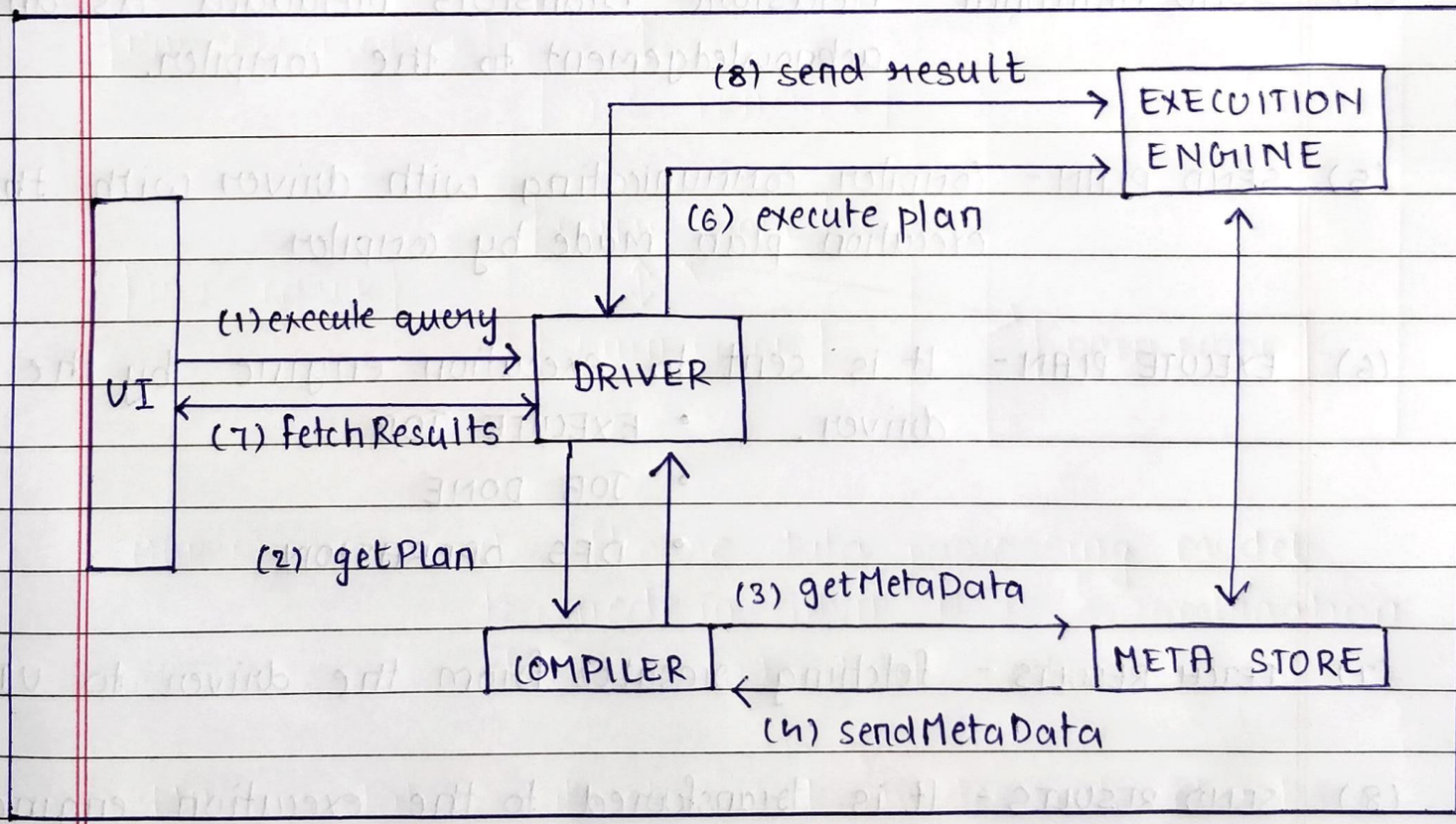
(4) db. residential. updateOne ({ cust_id : "1234567" },
{ $ set : { person_name :
"Rajesh Kapoor" }
})

(5) db. residential . find ( ) . sort ({ soc_land_area_sq_ft
: -1 }) . Limit (1)

Q4. APACHE HIVE IS a data warehouse & an EIL tool that provides on SQL like interface between the user & the hadoop distributed file system (HDFS) which integrates hadoop. It is buit on top of hadoop.

Major components of hive & its interaction are:

- USER INTERFACE
- DRIVER
- COMPILER
- META STORE
- EXECUTION ENGINE



HIVE ARCHITECTURE

(1) EXECUTE QUERY - Interface of the Hive such as cmd line or web interface delivers query to execute. In this UI calls the execute interface to the driver.

(II) GET PLAN - Driver design a session handle for the query and transfer the query to the compiler to make execution plan.

(3) GET METADATA - In this compiler transfer the metadata request to any database & the compiler gets the necessary metadata.

(4) SEND METADATA - Metastore transfers metadata as an acknowledgement to the compiler.

(5) SEND PLAN - Compiler communicating with driver with the execution plan made by compiler

(6) EXECUTE PLAN - It is sent to execution engine by the driver.
- EXECUTE JOB
- JOB DONE
- DFS OPERATION

(7) FETCH RESULTS - fetching results from the driver to UI.

(8) SEND RESULTS - It is transferred to the execution engine from the driver. Sending results to execution engine when results is retrived from data nodes to the execution engine.

Q5. Probability is mathematical foundation of statistics inference which is indespensible for analyzing data affected by chance & hence essential for data science

It's use in data science is mainly because of statistics. Probability in the base of statistics and it is the base of data science. Since statistics involve the analysis of sample data, there is always some degree of uncertainity present.

It also helps in making prediction.
For eg. metorologists use weather patterns to predict probability of rain.

TYPES OF PROBABILITY :

(1) THEORETICAL - It is based on the possible chances of something to happen.

(2) EXPERIMENTAL - It is based on the basis of observations of experiment. or num. of possible outcomes by number of trials.

(3) ARITHMATIC - In this, a set of rules or axioms are set which applies to all types, with this approach chances of occurence of events can be quantified.

18DCS007 (10)

There are six types of probability distribution
- BERNOULLI
- UNIFORM
- BINOMIAL
- NORMAL
- POISSON
- EXPONENTIAL

→ Total number of balls = 10
Let S be the sample space

$$n(S) = {}^{10}C_2$$
$$= \frac{10 \times 9}{2} = \frac{90}{2}$$
$$= 45$$

Let E be event of drawing 2 balls, out of which none of which is blue

$$n(E) = {}^{7}C_2$$
$$= \frac{7 \times 6}{2}$$
$$= 21$$

$$\therefore P(E) = \frac{n(E)}{n(S)} = \frac{21}{45} = \boxed{\frac{7}{15}}$$

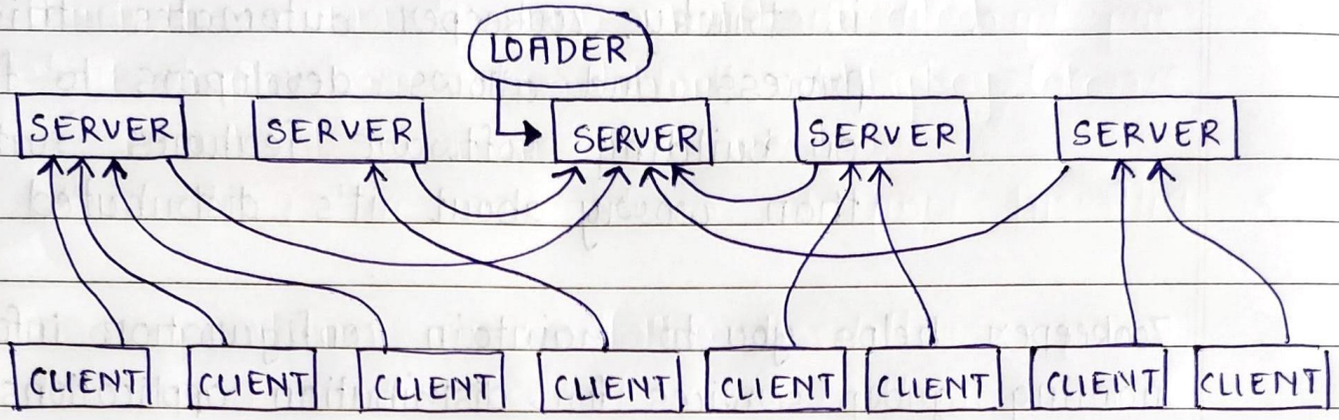
Q6. ZOOKEEPER - It is an open source distributed coordination service that helps to manage a large set of hosts. Management and coordination in a distributed environment is tricky. Zookeeper automates this process and allows developers to focus on building software features rather than worry about it's distributed nature.

Zookeeper helps you to maintain configuration information, naming, group services for distribution applications. It implements different protocols on the cluster so that the application should not implement on their own. It provides a single coherent view of multiple machine.

BENEFITS OF ZOOKEEPER

- Simple distributed coordination process
- Synchronization - mutual exclusion and cooperation between server processes. This process helps in Apache HBase for config. management
- Ordered Messages
- Serialization - encoded messages the data according to specific rules. Ensure your application runs consistently. This approach can be used in Map Reduce to coordinate queue to execute running threads.
- Reliability
- Atomicity - Data transfer either succeed or fail completely, but no transaction is partial.

→ ZOOKEEPER ARCHITECTURE - follows client server architecture
- all system store a copy
- leaders are elected at startup



→ SERVER - server sends an acknowledge when any client connects.

→ CLIENT - It is one of the nodes in the distributed application cluster.

→ LEADER - It gives all the information to the clients as well as an acknowledgement that the server is alive.

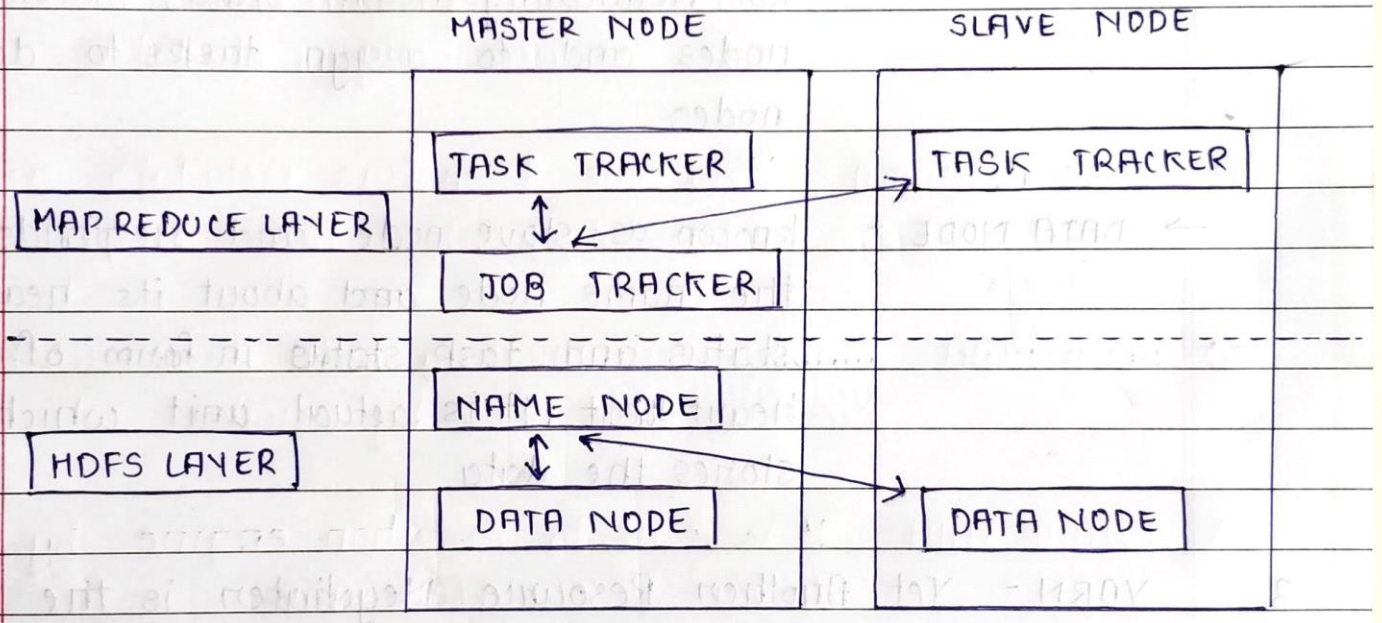
→ FOLLOWER - server node which follows leader instruction is called a follower.

→ ENSEMBLE - group of zookeeper servers is called ensemble. The minimum nodes of that is required to form an ensemble is 3.

Q.8. ADVANTAGES OF HADOOP :

- Varied data sources
- cost effective
- performance
- highly available
- high throughput
- open source

HADOOP ARCHITECTURE :



1. MAP REDUCE - A software data processing model designed in JAVA. It is a combination of two individual tasks

→ MAP - takes datasets and divides into chunks such that they are converted into a new format which is key-value pair.

→ REDUCE - the second part where key-value pairs are reduced to tuples.

2. HDFS - It is primary storge unit in the Hadoop ecosystem. It is the reason behinde the quick data accessing and generous scalability of Hadoop.

→ NAME NODE - known as master and is designed to store the meta data. It is responsible for monitoring health status of slave nodes and to assign tasks to data nodes

→ DATA NODE - known as slave node and responds to the name node ~~and~~ about its health status and task status in form of heart beat. It is actual unit which stores the data.

3. YARN - Yet Another Resource Negotiater is the update to hadoop since its second version. It is responsible for resource management and the job scheduling

It comprises of the following :

- RESOURCE MANAGER
- NODE MANAGER
- APP MANAGER
- CONTAINER

Q9. We need to create 3 bins using equal frequency by smoothing bin boundaries

(A) DATA : 1, 5, 8, 17, 20, 21, 24, 29, 30, 31, 34

→ First of all, we need to sort data to create bins by bin boundaries method. Here the data is already sorted.

→ Now using equal frequency we will divide data into 3 bins

BIN 1 - 1, 5, 8, 17
BIN 2 - 20, 21, 24, 29
BIN 3 - 30, 30, 31, 34

SMOOTHING BY BIN BOUNDARIES
↳ Pick minimum value of bin and put on left side
↳ Pick maximum value of bin & put on right side
↳ Middle values in bin boundaries move to its closest neighbour value with less distance

BEFORE BIN BOUNDARY - BIN 1 : 1, 5, 8, 17
AFTER BIN BOUNDARY - BIN 1 : 1, 1, 1, 17
(Here 5 and 8 are close to 1, so they will be treated as 1)

BEFORE BIN BOUNDARY : BIN 2 - 20, 21, 24, 29
AFTER BIN BOUNDARY : BIN 2 - 20, 20, 20, 29

BEFORE BIN BOUNDARY :- BIN 3 - 30, 30, 31, 34
AFTER BIN BOUNDARY :- BIN 3 - 30, 30, 30, 34

(B) DATASET - 10, 48, 57, 62, 89, 111, 10, 48, 89, 10, 53.4, 52.5, 10, 101

→ MEAN = $\frac{\text{sum of terms}}{\text{total terms}}$

$= \frac{750.9}{14}$

$= 53.6$

→ MEDIAN

sorted dataset - 10, 10, 10, 10, 48, 48, 52.5, 53.4, 57, 62, 89, 89, 101, 111

Median for even no. = $\frac{(n/2)^{th} \text{ obs} + (n/2 + 1)^{th} \text{ obs}}{2}$

$= (7^{th} + 8^{th}) \text{ obs} / 2$

$= \frac{52.5 + 53.4}{2}$

$= 52.95$

→ MODE = 10 (It has highest freqeuncy i.e. 4 times)

→ RANGE = highest obs - lowest obs

$= 111 - 10$

$= 101$

Q10. Important features of good data visualization :

→ VISUALLY APPEALING - the advent of more sophisticated visual creation tools and the high quality of mobile apps have raised the bar very high

→ SCALABILITY - others will want to use & leverage data so be sure to build your visualization on a system thats scalable for accesibility and for future maintainance & modifications

→ RIGHT INFORMATION - Its problem when user's focus on visual & not on what they really want. before creating vintualization, define exactly how it will be used.

→ ACCESSIBLE - An accesible visualization is easy to use and can be modified easily when necessary. It is called critical adoption

→ RAPID DEVELOPMENT & DEPLOYMENT - User need their information today & if you can't provide they will find other ways.

SCATTER PLOT - It is a chart type that is normally used to observe & visually display the relationship between variables.

The value of variables are represented by dots. The positioning of dots on vertical & horizontal axis will inform the value of respective data point.

The type of data for which scatter plot is used are:

→ We use scatter plot to determine if or not two variables have a relationship or correlation
Eg. If we are running icecream business & you're curious to see pattern of sales which are low

→ We use scatter plot when our independent variable has multiple values for our dependent variable
Eg. If we're trying to determine if height & weight have a correlating, the height will placed on X axis & weight on Y axis

→ We use a scatter plot when we have two variable that pair well together
Eg. think about birth weight vs gestational age, longer the baby in utoras's, the more the weight

Q11. BIG DATA - It is a collection of large datasets that cannot be processed using traditional computing techniques. It is related to extracting meaningful data by analyzing complex data.

There are 3 types of BIG DATA
(1) STRUCTURED
(2) UNSTRUCTURED
(3) SEMI-STRUCTURED

CHARACTERISTICS OF BIG DATA
→ Volume
→ Velocity
→ Variety
→ Veracity
→ Value

→ VOLUME - Amongst the 5V's of Big Data, the first one is volume. It refers to the amount of data that exists. If the volume of data is large enough, it can be considered big data.

→ VELOCITY - It refers to how quickly data is generated & how quickly the data moves. This is an important aspect for companies dealing with big data to flow their data quickly.

→ VARIETY - It refers to the diversity of data types. We might obtain data from a number of different data sources, which may vary in value.

→ VERACITY - It refers to the quality & accuracy of data. Gathered data could have missing pieces, may be inaccurate or may not be able to provide real, valuable insight. It refers to level of trust there is in collected data.

→ VALUE - It refers to the value that big data provide & it relates directly to what we can do with that collected data. ~~We can do wit~~ Being able to pull value from big data is a requirement, as the value of big data increases significantly depending on insights ~~significantly~~ that can be gained from them