-

BDA Assignment No. → 1

Unit → 1

Q1 Give difference between Traditional data Versus Big data Approach.

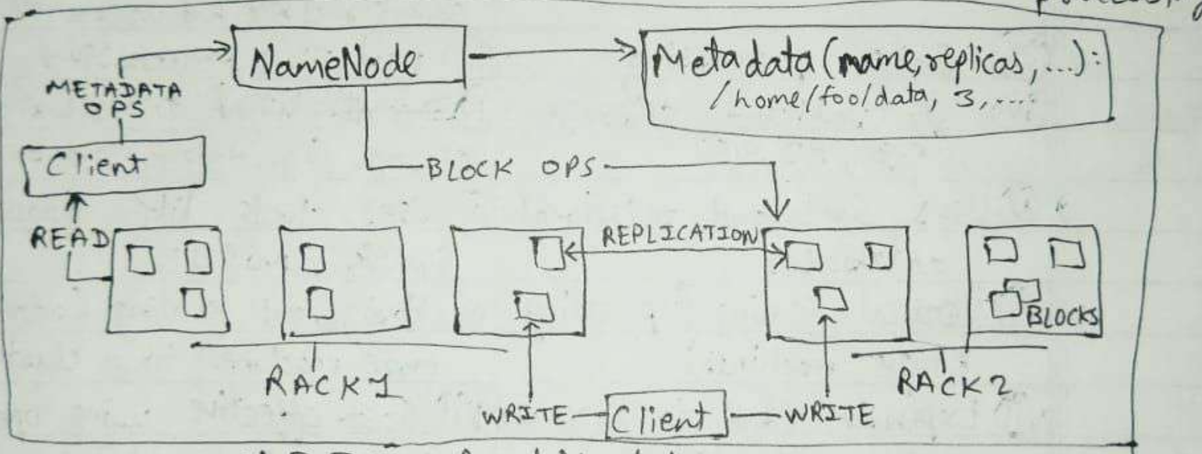
Ans →

Traditional Data Approach	Big Data Approach.
(i) Handles small to moderate data volumes (MBs to GBs)	(i) Handles massive data volumes (TBs to PBs)
(ii) Primary structured data	(ii) Deals with structured, semi-structured and unstructured data.
(iii) Batch processing	(iii) Real-Time Processing.
(iv) Centralized storage (e.g., RDMS)	(iv) ~~Decentralized~~ Distributed storage (e.g., Hadoop, cloud-based)
(v) Uses SQL and relational databases	(v) Uses tools like Hadoop, Spark, NoSQL
(vi) Vertical scaling (upgrading single machines)	(vi) Horizontal scaling (adding more machines to a cluster)
(vii) Expensive due to high-end servers and licenses	(vii) Cost-effective using open-source tools and commodity hardware.
(viii) Fixed schema; hard to ~~modify~~ modify.	(viii) Flecible schema or schema-on-read.
(ix) Clean and consistent data.	(ix) Often messy, noisy and requires cleaning.
(x) Suitable for traditional business applications	(x) Ideal for analytics, AI/ML, IoT, and large-scale insights.

Q.2 Describe HDFS architecture with a diagram.

Ans→ HDFS (Hadoop Distributed File System) architecture is a master/slave design where a single NameNode (master) manages the entire file system namespace and manges the metadata of files, while Multiple DataNodes (slaves) store the actual data blocks of files across the cluster.

Files in HDFS are divided into large blocks, and these blocks are replicated and distributed across multiple DataNodes for fault tolerance and parallel processing.



HDFS Architecture.

Main comoponents of HDFS :-

(i) Name Node → The master node that manages the file system namespace and regulates access to files by clients.

(ii) DataNode → The worker nodes that store actual data. They are responsible for serving read and write requests from clients.

(iii) Secondary NameNode → Assits the NameNode by periodically merging its namespace image with the edit logs to prevent the logs from growing too large. It is not a backup NameNode.

(iv) Client -> An application or user interacting with the HDFS to store or retrieve Data.

How HDFS Works :-

(i) File Splitting -> Files are split into blocks (default 128MB or 256MB)

(ii) Block Replication -> Each block is replicated (usually 3 copies) across different DataNodes for fault tolerance.

(iii) Client Request -> Client contacts NameNode to get metadata and block locations.

(iv) Data Transfer => Client reads / writes directly to / from DataNodes.

Key Feature :-

(i) Fault Tolerance -> Data is replicated across nodes.

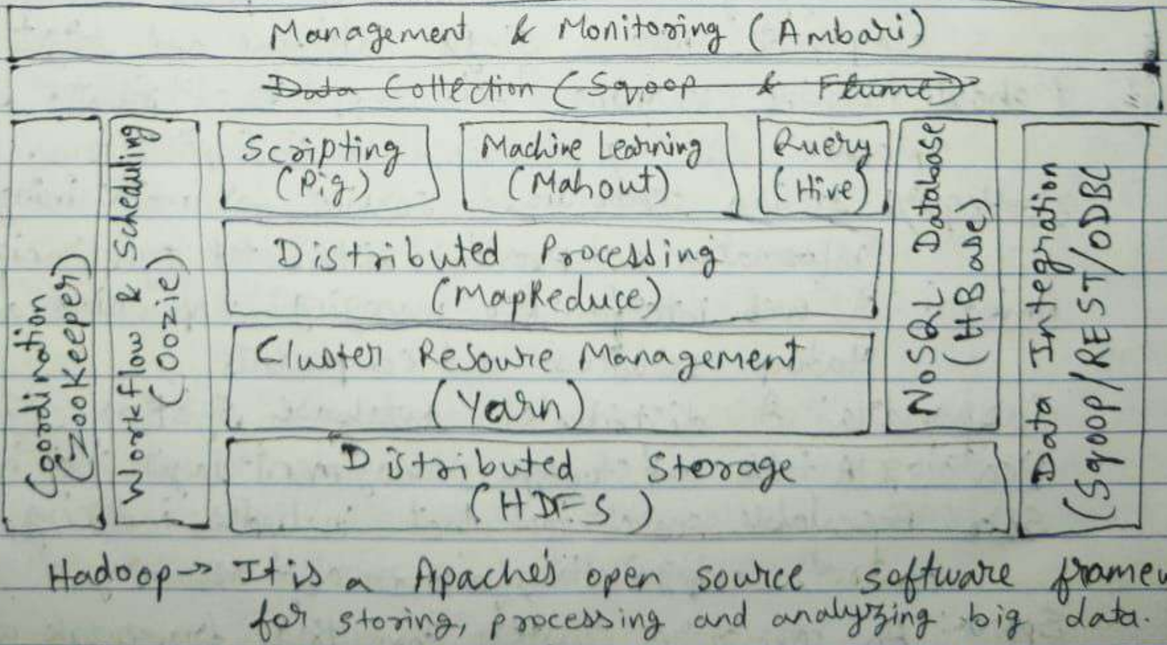
(ii) High Throughput -> Optimized for large-scale batch processing

(iii) Scalability -> Easily scalable by adding more nodes.

(iv) Cost-Effective -> Works on commodity hardware.

Q.3 Draw Hadoop ecosystem and briefly explain its components.

Ans -> Hadoop Ecosystem


Management & Monitoring (Ambari)
~~Data Collection (Sqoop & Flume)~~
Coordination (ZooKeeper)
Workflow & Scheduling (Oozie)
Scripting (Pig)
Machine Learning (Mahout)
Query (Hive)
Distributed Processing (MapReduce)
Cluster Resource Management (Yarn)
Distributed Storage (HDFS)
NoSQL Database (HBase)
Data Integration (Sqoop/REST/ODBC)

-> Hadoop -> It is a Apache's open source software framework for storing, processing and analyzing big data.

Hive: A data warehousing and SQL-like query language that presents data in the form of tables. Hive programming is similar to database programming.

Pig: Pig is an alternative abstraction on top of MapReduce It uses a dataflow scripting language called as PigLatin It's interpreter runs on the client machine.

HBase: HBase is "Hadoop Database". It is a 'NoSQL' data store. It can store massive amount of data.

Flume: It is distributed real time data collection service. It effectively collects, aggregate and move large amounts of data.

Sqoop: It provides a method to import data from tables in relational datatabase into HDFS. It supports ~~easy~~ easy parallel database import/ export.

Oozie: Oozie is a workflow management project. Oozie allows developers to create a workflow of MapReduce jobs including dependencies between jobs.

Hue: An open source web interface that supports Apache Hadoop and its ecosystem licensed under the Apache v2 licence. Hue aggregates the most common Apache Hadoop components into a single interface and targets the UI.

Mahout: Machine learning tool. Supports distributed & scalable machine learning algorithm on HadoopPlatform.

ZooKeeper: It is a centralized service for maintaining configuration information and provides distributed synchronization.

Ambari: A web. interface for managing configuring and testing Hadoop services and components.

Cassandra: A distributed database System.

Hcatalog: A table and storage management layer that helps users share and access data.

Solr: A scalable search tool that includes indexing, reliability, central, configuration, failover and recovery.

Spark: An open-source cluster computing framework with in-memory analytics.

Q.4) Describe 5 characteristics of Big Data in detail.

Ans→ #Volume: (i) Refers to the vast amount of data generated every second from various sources. (ii) Data size ranges from terabytes (TB) to ~~Pentab~~ Petabytes (PB) and even zettabytes. (iii) Requires scalable storage solutions like HDFS or cloud storage. (iv) Examples include data from social media, e-commerce, IoT devices, and sensors.

#Velocity: (i) Refers to the speed at which data is generated collected, processed. (ii) Demands real-time or near real-time processing of incoming data streams. (iii) Technologies like Apache Kafka and Spark streaming are used to handle this. (iv) Common is stock trading systems, online transactions, and live tracking apps.

#Variety: (i) Refers to the different forms of data - structured, semi-structured and unstructured. (ii) Data sources include text, images, video, audio, sensor logs, and web data. (iii) Handling diverse data formats requires flexible storage and processing tools. (iv) NoSQL databases like MongoDB and HBase manage variety effectively.

#Veracity: (i) Refers to the accuracy, reliability, and trustworthiness of data. (ii) Big Data often includes incomplete, inconsistent, or noisy data. (iii) Low-quality data can ~~be~~ lead to incorrect analysis and decisions. (iv) Data cleansing, validation, and filtering are necessary to ensure data quality.

#Value: (i) Refers to the meaningful insights and business benefits extracted from data. (ii) The ultimate goal of Big Data analytics is to create real-world value (iii) Valuable data helps in decision-making, prediction, and optimization. (iv) Example: Personalized marketing based on user behaviour and preferences.

Q.5 Explain types of Big Data.

Ans→ There are three types of Big Data :

- Structured
- Unstructured
- Semi-structured.

(i) <u>Structured</u>: Any data that can be stored, accessed and processed in the form of fixed format is termed as a 'structured' data. Over the period of time, talent in computer science has achieved greater success in developing techniques for working with such kind of data (where the format is well known in advance) and also deriving value out of it. However, nowadays, we are foreseeing issues when a size of such data grows to a huge extent, typical sizes are being in the rage of multiple zettabytes.

Example of Structured Data - An 'Employee' table in a database. ~~database~~ . ~~in an~~

(ii) <u>Unstructured</u> : Any data with unknown form or the structure is classified as unstructured data. In addition to the size being huge, unstructured data poses multiple challenges in terms of its processing for deriving value out of it. A typical example of na unstructured data is a heterogeneous data source containing a combination of simple text files, images, videos, etc. Now day organizations have wealth of data available with them but unfortunately, they don't know how to derive value out of it since this data is in its raw form or unstructured format.

Example :- The output returned by "Google Search'.

(iii) <u>Semi-Structured</u>: Semistructured data can contain both the forms of data. We can see semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS. Example of semi-structured data is a data represented in an XML file or the Personal data stored in an XML file.

Q.6 What are the advantages and limitation of Hadoop?

Ans→ Advantages of Hadoop:-

(i) Scalability – Hadoop can easily scale horizontally by adding more machines (nodes) to the cluster.
- It can ~~had~~handle petabytes of data efficiently.

(ii) Cost-Effective – It uses commodity hardware and open-source software, reducing overall infrastructure costs.
- No expensive licenses are needed.

(iii) Fault Tolerance – Data is replicated across multiple nodes. If one node fails, data is still accessible from others
- Built-in recovery mechanisms ensure high availability.

(iv) Flexibility in Data Handling – Can process structured, semi-structured, and unstructured data.
- Works well with logs, images, videos, social media, and more.

(v) Parallel Processing – Hadoop uses the MapReduce model, which enables high-speed parallel data processing across many nodes.
- Increases performance on large data sets.

Limitations of Hadoop:-

(i) Security Concerns – Hadoop lacks strong built-in security features like encryption and user authentication.
- Requires third-party tools to implement robust security measures.

(ii) Vulnerable by Nature – As an open-source system running on commodity hardware, it can be more vulnerable to cyberattacks or hardware failures.
- Needs careful configuration and monitoring.

(iii) Not fit for small data — It is inefficient and overkill for small data sets or simple processing tasks.

— Hadoop is optimized for ~~ta~~ larger-scale data processing.

(iv) Potential Stability Issues — Running hadoop at large scale can sometimes result in instability or node failures.

— Cluster management and monitoring tools are necessary for stability.

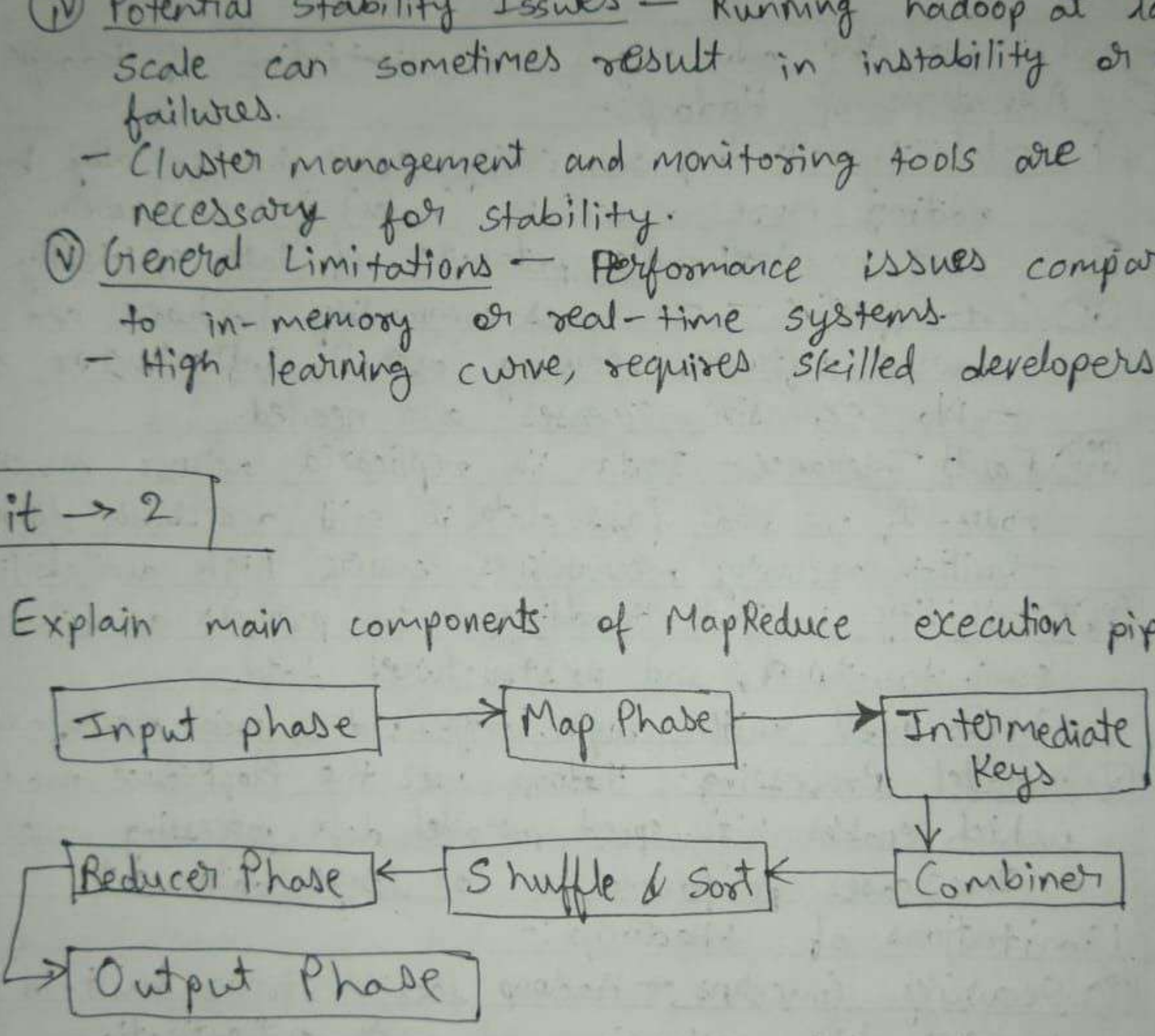
(v) General Limitations — Performance issues compared to in-memory or real-time systems.

— High learning curve, requires skilled developers.

Unit → 2

Q) Explain main components of MapReduce execution pipeline.

Ans →



Input Phase :— Here we have a Record Reader that translates each record in an input file and sends the parsed data to the mapper in form of key-value pairs.

Map Phase :— Map is a user defined function, which takes a series of key-value pairs and processes each one of them to generate zero or more key-value pairs.

Intermediate Keys :- the key-value pairs generated by the mapper are known as intermediate keys.

Combiner :- A combiner is a type of local Reducer that groups similar data from the map phase into identifiable sets. It takes the intermediate keys from the mapper as input and applies a user-defined code to aggregate the values in a small scope of one mapper. ~~It is not a part of the main MapReduce~~

Shuffle and Sort :- The Reducer task starts with the Shuffle and Sort step. It downloads the grouped key-value pairs onto the local machine, where the Reducer is running. The individual key-value pairs are sorted by key into a larger data list. The data list groups the equivalent keys together so that their values can be iterated easily in the Reducer task.

Reducer :- The Reducer takes the grouped key-value paired data as input and runs a Reducer function on each one of them. Here, the data can be aggregated, filtered, and combined in a number of ways, and it requires a wide range of processing. Once the execution is over, it gives zero or morekey-value pairs.

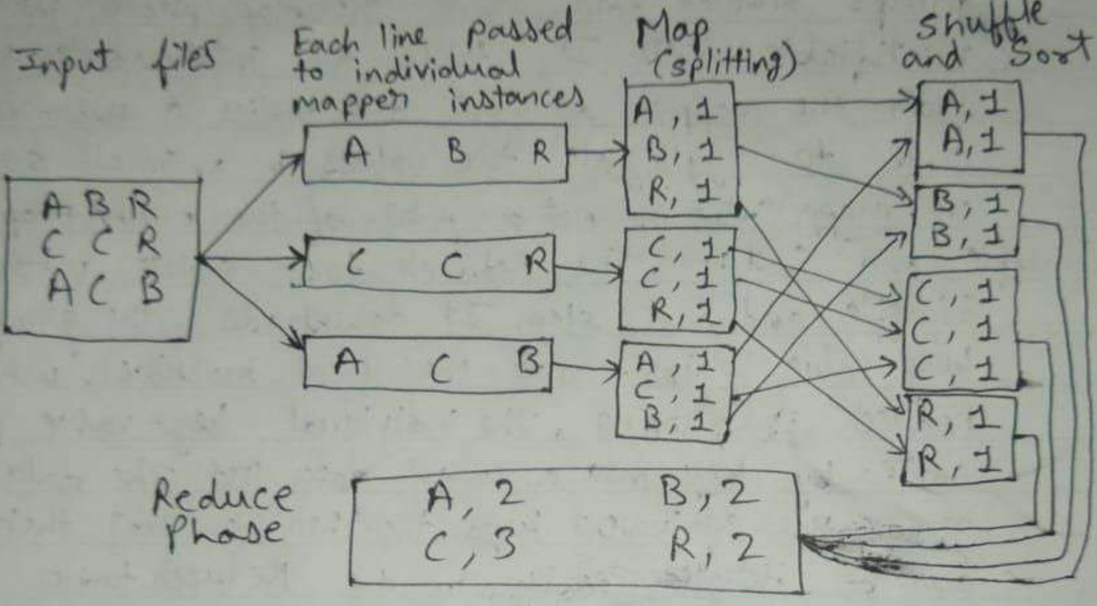
Output Phase: - In the output phase, we have an output formatter that translates the final key-value pairs from the Reducer function and writes them onto a file using a record writer.

Q2 Explain the concept of Map-Reduce with word count with example.

Ans MapReduce Word Count is a framework which splits the chunk of data, sorts the map outputs and input to reduce tasks. A File-system stores the output and input of jobs.

Re-execution of failed tasks, scheduling them and monitoring them is the task of the framework.

Architecture/Working of MapReduce with an example :



Q.3 Explain Selection and Projection algebric operations using MapReduce.

Ans→ # Selection :

→ selections really do not need the full power of MapReduce. They can be done most conveniently in the map portion alone, although they could also be done in the reduce portion alone.

→ Apply a condition C to each tuple in the relation and produce as output only those tuples that satisfy C. The result of this selection is denoted $\sigma C(R)$.

→ Here is MapReduce implementation of selection $\sigma C(R)$.

- The Map Function: For each tuple t in R, test if it satisfies C. If so, produce the key-value pair (t,t). That is, both the key and value are t.
- The Reduce Function: The Reduce function is the identity. It simply passes each key-value pair to the output.

Projection :

→ For some subset S of the attributes of the relation, produce from each tuple only the components for the attributes in S. The result of this projection is denoted $\pi S(R)$.

→ Projection is performed similarly to selection, because projection may cause the same tuple to appear several times, the Reduce function must eliminate duplicates.

→ We may compute $\pi S(R)$ as follows:-

The Map Function : For each tuple t in R, construct a tuple t'' by eliminating from t those components whose attributes are not in S. Output the key-value pair (t', t').

The Reduce Function : For each key t' produced by any of the Map tasks, there will be one or more key-value pairs (t', t'). The Reduce function turns $(t', [t', t', \ldots, t'])$ into (t', t'), so it produces exactly one pair (t', t') for this key.

Q.1) Write a MapReduce pseudo code to multiply two matrices. Illustrate with an example showing all the steps.

Ans→

```
for each element $m_{ij}$ of M do
        produce (key, value) pairs as $((i,k), (M, j, m_{ij}))$
        for k = 1, 2, 3, -- up to the no. of columns of N
for each element $n_{jk}$ of N do
        produce (key, value) pairs as $((i,k), (N, j, n_{jk}))$
        for i = 1, 2, 3, ... up to the no. of rows of M
return Set of (key, value) pairs that each key, (i, k), has a
 list with values $(M, j, m_{ij})$ and $(N, j, n_{jk})$ for all possible
 values of j
```

Above pseudo code is Map Function for Matrix-Multiplication.

```
for each key (i,k) do
    sort values begin with M by j in list M
    sort values begin with N by j in list N
    multiply mij and njk for jth value of each list
    sum up mij * njk
return (i,k), $\sum_{j=1} m_{ij} \times n_{jk}$
```

Above pseudo code is Reduce function for Matrix Multiplication.

$$A = \begin{array}{c|cc} & 0 & 1 \;\rightarrow j \\ \hline 0 & 1 & 2 \\ 1 & 3 & 4 \\ \downarrow i & & \end{array}_{i \times j} \qquad B = \begin{array}{c|cc} & 0 & 1 \;\rightarrow k \\ \hline 0 & 2 & 1 \\ 1 & 5 & 1 \\ \downarrow j & & \end{array}_{j \times k}$$

Step I : ~~Ip~~ Input file < key, value>

Mapping for ~~B~~A < $\underset{key}{(i,k)}$, (Matrix, j, $\underset{(A_{ij})}{value}$) >

value

Mapping for B < $\underset{\cancel{key}}{(i,k)}$, (Matrix, j, $\underset{(B_{jk})}{value}$) >

∴ For A , $A_{ij} = A_{00} = 1$ < (0,0) ,(A,0,1)>
< (0,1) ,(A, 0, 1)>

$A_{01} = 2$ < (0,0),(A , 1 ,2)>
< (0,1),(A , 1 ,2)>

$A_{10} = 3$ < (1, 0),(A , 0,3)>
< (1,1),(A, 0 ,3)>

$A_{11} = 4$ < (1, 0), (A ,1 ,4)>
< (1, 1), (A, 1 ,4) >

∴ For B , $B_{jk} = B_{00} = 2$ < (0, 0),(B, 0, 2)>
< (0, 1),(B ,0 , 2)>

$B_{01} = 1$ < (0, 0),(B, 1, 1)>
< (0, 1),(B, 1, 1)>

$B_{10} = 5$ < (1, 0),(B, 0, 5)>
< (1, 1),(B, 0, 5)>

$B_{11} = 1$ < (1, 0),(B, 1 , 1)>
< (1, 1),(B, 1, 1)>

Step II : Combine or Grouping

based on (i, k) key

$(0,0) \rightarrow (A, 0, \boxed{1}) \times + (A, 1, \boxed{2})$
$\qquad (B, 0, \boxed{2}) \quad (B, 1, \boxed{5}) \times$

$(0,1) \rightarrow (A, 0, 1) \; (A, 1, 2)$
$\qquad (B, 0, 1) \; (B, 1, 1)$

$(1,0) \rightarrow (A, 0, 3) \; (A, 1, 4)$
$\qquad (B, 0, 2) \; (B, 1, 5)$

$(1,1) \rightarrow (A, 0, 3) \; (A, 1, 4)$
$\qquad (B, 0, 1) \; (B, 1, 1)$

Step ~~3~~ III : Reduce

C → (i, k)

$(0,0) = (1 \times 2) + (2 \times 5) = 2 + 10 = 12$

$(0,1) = (1 \times 1) + (2 \times 1) = 1 + 2 = 3$

$(1,0) = (3 \times 2) + (4 \times 5) = 6 + 20 = 26$

$(1,1) = (3 \times 1) + (4 \times 1) = 3 + 4 = 7$

$$C_{i \times k} = \begin{array}{c|cc} i \downarrow \; k \rightarrow & 0 & 1 \\ \hline 0 & 12 & 3 \\ 1 & 26 & 7 \end{array}_{\, i \times k}$$

Unit → 3

Q) Differentiate between SQL vs NoSQL.

Ans →