Q6 A]

(:|)

When it comes to big data NOSQL scores over RDBMS

- NoSQL is designed for operational needs - real-time application that often interface with customers or parties external to the organization.
- It provides the ability to query the data, so users can drill down into the data as it changes
- NO SQL allows for high-performance, agile processing of information at massive scale.
- It stores unstructured data across multiple processing nodes as well as across multiple servers.
- As such, the NoSQL distributed database infrastructure has been the solution of choice for some of the largest date warehouses.
- NoSQL seems to work better on both unstructured and unrelated data. The better solutions are the crossover databases that have elements of both NOSQL and SQL.
- RDBMSs that use SQL schema-oriented which means the structure of the data should be known in advance to ensure that the data adheres to the schema.
- For example, predefined schema based application that use SQL include Payroll Management System Order Processing and Flight Reservations

Q 6 B]

(ii)

Different Architectural patterns in NOSQL

① Key-Value Databases examples:
→ Riak, Redis, memcached, BerkeleyDB, upscaleDB, Amazon DynamoDB.

② Document Databases Examples:
→ MaggoDB, CouchDB, Terrastore, OrientDB, RavenDB

③ Column Family stores Examples:
→ Cassendra, HBase, HyperTable.

④ Graph Databases examples:
→ Neo4j, infinite Graph, FlockDB

Graph Database.

- Data is stored as a graph and their relationships are stored as a link between them whereas entity acts like a node.
- Examples:
  1. Neo4j
  2. Polyglot.

Column Database

- Instead of storing data in relational tuples (table rows), it is stored in cells grouped in colums.
- It offers very high performance and a high scalable architecture.
- Example:
  1. HBase
  2. Big Table
  3. Hyper Table

AMEY THAKUR BDA 03-09-2021

BE B 50 IAT-I Amey

Q6 B](1)

Job Tracker

1. Job Tracker process runs on a separate node and not usually on a Data Node.
2. Job Tracker is an essential Daemon for Mapreduce execution in MRv1. It is replaced by Resource Manager / Application Master in MRv2
3. Job Tracker receives the request for MapReduce execution from the client
4. JobTracker talks to the NameNode to determine the location of the data

Task Tracker

1. Task Tracker runs on Data Node. Mostly on all Data Nodes.
2. Task Tracker is replaced by Node Manager in MRV2
3. Mapper and Reducer tasks are executed on Data Nodes administered by Task Tracker.
4. Task Trackers will be assigned mapper and Reducer tasks to execute by JobTracker