**ASSIGNMENT 9.3**

Explain the below concepts with an example in brief.

**1) Nosql Databases**

Solutions:

NoSQL is an approach to databases that represents a shift away from traditional relational database management systems (RDBMS). To define NoSQL, it is helpful to start by describing SQL, which is a query language used by RDBMS. Relational databases rely on tables, columns, rows, or schemas to organize and retrieve data. In contrast, NoSQL databases do not rely on these structures and use more flexible data models. NoSQL can mean “not SQL” or “not only SQL.” As RDBMS have increasingly failed to meet the performance, scalability, and flexibility needs that next-generation, data-intensive applications require, NoSQL databases have been adopted by mainstream enterprises. NoSQL is particularly useful for storing unstructured data, which is growing far more rapidly than structured data and does not fit the relational schemas of RDBMS. Common types of unstructured data include: user and session data; chat, messaging, and log data; time series data such as IoT and device data; and large objects such as video and images.

**2) Types of Nosql Databases**

Solutions:

Several different varieties of NoSQL databases have been created to support specific needs and use cases. These fall into four main categories:

Key-value data stores: Key-value NoSQL databases emphasize simplicity and are very useful in accelerating an application to support high-speed read and write processing of non-transactional data. Stored values can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key. The application has complete control over what is stored in the value, making this the most flexible NoSQL model. Data is partitioned and replicated across a cluster to get scalability and availability. For this reason, key value stores often do not support transactions. However, they are highly effective at scaling applications that deal with high-velocity, non-transactional data.

Document stores: Document databases typically store self-describing JSON, XML, and BSON documents. They are similar to key-value stores, but in this case, a value is a single document that stores all data related to a specific key. Popular fields in the document can be indexed to provide fast retrieval without knowing the key. Each document can have the same or a different structure.

Wide-column stores: Wide-column NoSQL databases store data in tables with rows and columns similar to RDBMS, but names and formats of columns can vary from row to row across the table. Wide-column databases group columns of related data together. A query can retrieve related data in a single operation because only the columns associated with the query are retrieved. In an RDBMS, the data would be in different rows stored in different places on disk, requiring multiple disk operations for retrieval.

Graph stores: A graph database uses graph structures to store, map, and query relationships. They provide index-free adjacency, so that adjacent elements are linked together without using an index.

**3) CAP Theorem**

Solutions:

a) Consistency - This means that the data in the database remains consistent after the execution of an operation. For example after an update operation, all clients see the same data.

b) Availability - This means that the system is always on (service guarantee availability), no downtime.

c) Partition Tolerance - This means that the system continues to function even if the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

**4) HBase Architecture**

Solutions:

HBase is composed of three types of server in master slave type of architecture:

a) Region Server:

HBase Tables are divided horizontally by row key range into “Regions.” A region contains all rows in the table between the region’s start key and end key. Regions are assigned to the nodes in the cluster, called “Region Servers,” and these serve data for reads and writes. A region server can serve about 1,000 regions.

b) HBase:

Region assignment,DDL (create, delete tables) operations are handled by the HBase Master.

A master is responsible for:

• Coordinating the region servers

• Assigning regions on startup

• Re-assigning regions for recovery or load balancing

• Monitoring all RegionServer instances in the cluster (listens for notifications from zookeeper)

Admin functions

• Interface for creating, deleting, updating tables

c) Zookeeper:

• HBase uses ZooKeeper as a distributed coordination service to maintain server state in the cluster.

• Zookeeper maintains which servers are alive and available, and provides server failure notification.

• Zookeeper uses consensus to guarantee common shared state. Note that there should be three or five machines for consensus.

**5) HBase vs RDBMS**

Solutions:

|  |  |
| --- | --- |
| HBase | RDBMS |
| 1) HBase is a distributed, columnoriented data storage system. | 1) RDBMS is row-oriented databases |
| 2) Hbase tables do not have fixed schema | 2) RDBMS tables have fixed-schema |
| 3) Hbase tables guarantee consistency and partition tolerance | 3) RDBMS tables guarantee ACID  properties |
| 4) Hbase uses Java client API and  Jruby. | 4) RDBMS uses SQL (Structured  query Langauge ) to query the  data |