Big Data and Analytics

ASSIGNMENT-3.

Course Code : 18CS702.

Ranjitha. C.G
18CS073.
'B' section.

① Differentiate Between RDBMS and MongoDB.

Ans:

RDBMS	MongoDB.
* It is a relational dB.	* It is a non-relational and document-oriented database.
* It is not suitable for hierarchical data storage.	* It is suitable for hierarchical data storage.
* It is a predefined schema	* It has a dynamic schema
* It is row-based	* It is document based.
* RDBMS is slower in processing large hierachical data.	* MongoDB is fast in processing large hierachical data.
* RDBMS do not provide Javascript based client to query database.	* MongoDB provides Javascript based client to query DB.
* RDBMS uses SQL to query database.	* MongoDB uses BSON to query database.

② Explain the features of MongoDB.

Ans:
1. Schema-less Database
2. Document Oriented.
3. Indexing
4. Scalability
5. Replication
6. Aggregation
7. High Performance.

1. Schema-less Database:

It is the great feature provided by the MongoDB. A schema-less database means one collection can hold different types of documents in it.
It is not necessary that one document is similar to another document like in the relational databases. Due to this cool feature, MongoDB provides great flexibility to databases.

2. Document Oriented:

In MongoDB, all the data stored in the documents instead of tables like in RDBMS. The data is stored in key-value pair instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.

3. Indexing:

In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool specified query which takes lots of time and not so efficient.

4. Scalability:

MongoDB provides horizontal scalability with the help of sharding. sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, & these data chunks are evenly distributed across shards that resides across many physical servers.

5. Replication:

MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.

6. Aggregation:

It allows to perform operations on the grouped data and get a single result or computed result ie., aggregation pipeline, map-reduce function, and single-purpose aggregation methods.

7. High Performance:

The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.

③ Illustrate the significance of _id in MongoDB.

Ans: * Each JSON document should have a unique identifier, it is _id key, which is similar to primary key in relational DB.

* This key facilitates search for documents based on unique identifier.

Database :- It is a collection of collections. It is like a container for collections. It gets created the first time that your collections makes a reference to it. This can also be created on demand.

Collection :- A collection is analogous to a table of RDBMS.

* A collection is created on demand it gets created the first time that you attempt to save a document that references it.
* A collection exists within a single DB and a collect^{n} holds several mongoDB documents.

Document :- A document is analogous to row or tuple or record in an RDBMS table.

* It has a document that has dynamic schema.

0 1 2 3	4 5 6	7 8	9 10 11
Timestamp	Machine ID	process ID	counter.

④ What is a cursor? How cursor is implemented in MongoDB? Explain with an example.

Ans: The Cursor is a MongoDB collection of the document which is returned upon the find method execution.

— It is automatically executed as a loop. we can explicitly get specific index document from being returned cursor.

— It is just like a pointer which is pointing upon a specific index value.

— We can call a find method, all the documents which are returned are saved in a virtual cursor.

— If a find method returns for a document then it is mean that the cursor has 0-3 index.

Example:

Database : doc

Collection : student

Documents : Three documents contain the details of the students.

To display all documents present in the student collection

```
db.student.find().pretty()
```

find() method will return a cursor with contain all documents present in the student collection.

```
{       "_id" : objectId("10k"),
  "studentId" : 1,
 "studentName : "Sonu",
 "studentAge : 20
}
{
    "_id"       : objectId("102b"),
   "studentId"  : 2,
   "studentName" : "Param",
   "studentAge" : 22
}
```

```
{
    "_id"          : objectID('103pk"),
   "studentID"     : 3,
   "studentName"   : "Rocky",
   "studentAge"    : 28
}
```

```
var mycursor = db.student.find({studentID : 3}).pretty()
mycursor
```

```
{
    "_id"          : objectId("103pk"),
   "studentId"     : 3,
   "studentName"   : "Rocky",
   "studentAge"    : 28
}
```

⑤. Why do you need Cassandra? Explain the features of Cassandra.

Ans: Apache Cassandra is a distributed database management system that is built to handle large amounts of data across multiple data centers and the cloud.

* Cassandra was originally developed at Facebook for their inbox search feature.
* Cassandra stands out among database systems and offers some advantages over other systems. Its ability to handle high volumes makes it particularly beneficial for major corporations.
* As a result, it's currently being used by many large business including Apple, Facebook, Instagram, Uber, Spotify, Twitter, Cisco, Rackspace, eBay, and Netflix.

Features of Cassandra:

i. Peer-to-Peer Network.
ii. Replication factor
iii. Scalability
iv. Fault-tolerance
v. MapReduce support
vi. Query language.

i. Peer-to-Peer Network:

Each node in the cluster has same role. There's no question of failure and the data set is distributed across the cluster but one issue is there that is the master isn't present in each node to support request for service.

ii. Replication factor:

It determines the number of copies of data (replicas) that will be stored across nodes in a cluster.
The replication factor should ideally be more than one and not more than the no of nodes in the cluster.
Two replication strategies are available:-

1. SimpleStrategy
2. NetworkTopologyStrategy.

The preferred one is NetworkTopologystrategy as it is simple and supports easy expansion to multiple data centers, should there be a need.

iii. Scalability:

It is designed to r/w throughput, Increase gradually as new machines are added without interrupting other applications.

iv. Fault-tolerance:

Data is automatically stored and replicated for fau-lt-tolerance. If a node Fails, then it is replaced within no time.

v. MapReduce support:

It support Hadoop integration with MapReduce support. Apache Hive & Apache Pig is also supported.

vi. Query language:

Cassandra has introduced the CQL (Cassandra Query language). Its a simple interface for accessing the Cassandra.

⑥. Illustrate the following commands in Cassandra.
i. TTL ii. SET iii. LIST iv. ALTER.

Ans: i. TTL

In TTL (Time to Live) an important role is, if we want to set the time limit of a column and we want to automatically delete after a point of time then at the time using TTL keyword is very useful to define the time limit for a particular column.

It is used to set the time limit for a specific period of time. By using TTL clause we can set the TTL value at the time of insertion.

ii. SET

If a table specifies a set to hold data, then either INSERT or UPDATE to enter data.

set values must be unique, because no order is defined in a set internally.

iii. LIST

A list has a form much like a set, in that a list groups and stores values.

Unlike a set, the values stored in a list do not need to be unique and can be duplicated.

A list stores the elements in a particular order and may be inserted or retrieved according to an index value.

iv. ALTER.

It will change the datatype of a columns, add new columns, drop existing columns, renames columns, and change table properties.
The command returns no results.

Restriction: Altering PRIMARY KEY columns is not supported. Altering columns in a table that has a materialized view is not supported.

(7.) Explain Replication strategy in cassandra.

Ans:
The replication factor determines the number of copies of data that will be stored across nodes in a cluster. If one wishes to store only one copy of each row on one node, they should set the replication factor to one. If the need is for two copies of each row of data on two different nodes, one should go with a replicat' factor of two.
The replication factor should ideally be more than one and not more than the no of nodes in the cluster.
A replication strategy is employed to determine which nodes to place the data on.

Two replication strategies are available:

1. SimpleStrategy
2. NetworkTopologyStrategy.

1. SimpleStrategy:

It is recommended for multiple nodes over multiple racks in a single data center.

2. NetworkTopologyStrategy:

It is the strategy in which we can store multiple copies of data on different data centers as per need. This is one important reason to use NetworkTopologySt-rategy when multiple replica nodes need to be placed on different data centers.

(8.) Write the objective of mongoDB queries by considering the following collection.

Students (sroll, sname, grade, hobbies, doj)

i. Db.students.find({grade:{$ne:"vii"}}).pretty();

objective :- To find those documents where grade is not set to 'vii'.

ii. Db.students.find({sname:/s$/}).pretty();

objective :- Find the document from the students collection where sname ends with 's'.

iii. Db.students.find({grade:"vii"}).limit(3);

objective :- Find the first 3 document from the student collection where in the grade is 7.

iv. Db.students.find({},{sroll:1, sname:1, hobbies:1, -id:0});

objective :- Display only the student roll no, student name and hobbies from all the documents of the students collection.

The identifier _id should be supressed and not displayed.

(9.) Write the following queries in MongoDB:

i. To create collection food and insert id and fruit-array with fruit names banana, apple and orange.

Query :- db.food.Insert({_id:1, fruits:['banana','apple','orange']});

ii. Find those documents from food collection which has the fruits array having "banana" as an element.

Query :- db.food.find({fruits:['banana']});

iii. Find those documents from food collection where the size of fruit array is three.

Query :- db.food.find({"fruits" : {$size : 3}});

iv. Find those documents from food collection which the size has the fruits array having "banana" in second index possition.

Query :- db.food.find({'fruits.2': 'banana'});

v. Updata the document with _id=4 by adding an element "grapes" to the fruit array.

Query :- db.food.update({_id:4}, {$set:{'fruits.1':"grapes"}});

(10.) Explain import and export command in Cassandra.