**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**



## LAB REPORT

**on**

# BIG DATA ANALYTICS (20CS6PEBDA)

***Submitted by***

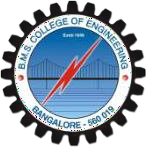
## Sharan S Pai (1BM19CS146)

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

## COMPUTER SCIENCE AND ENGINEERING



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

## BENGALURU-560019

**May-2022 to July-2022**

## B. M. S. College of Engineering,

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “**BIG DATA ANALYTICS**” carried out by **Sharan S Pai (1BM19CS146),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **BIG DATA ANALYTICS - (20CS6PEBDA)** work prescribed for the said degree.

**Antara Roy Choudhary Dr. Jyothi S Nayak**

Assistant Professor Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

`

# Index Sheet

|  |  |  |
| --- | --- | --- |
| **Sl.**  **No.** | **Experiment Title** | **Page No.** |
| **1** | **MongoDB Practice Problems** | **4** |
| **2** | **Employee DB (Cassandra)** | **9** |
| **3** | **Library DB (Cassandra)** | **12** |

**Course Outcome**

|  |  |
| --- | --- |
| CO1 | Apply the concept of NoSQL, Hadoop or Spark for a given task |
| CO2 | Analyze the Big Data and obtain insight using data analytics mechanisms. |
| CO3 | Design and implement Big data applications by applying NoSQL, Hadoop or Spark |

# MongoDB- CRUD Demonstration

Mongo DB Expts

1. Create a collection called students

>db.createCollection(‘student’);



1. Insert an element into the collection which has name, grade, hobbies

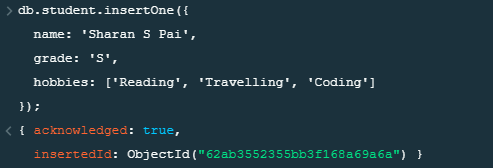
>db.student.insertOne({

name: 'Sharan S Pai',

grade: 'S',

hobbies: ['Reading', 'Travelling', 'Coding']

});

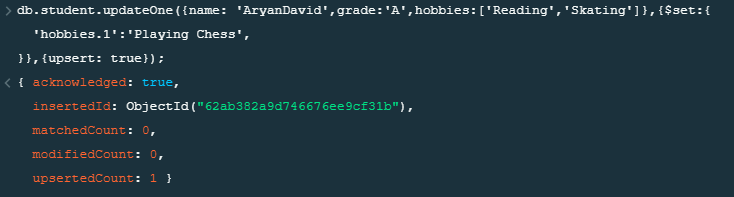


1. Insert the document for “AryanDavid” in to the Students collection only if it does not already exist in the collection. However, if it is already present in the collection, then update the document with new values. (Update his Hobbies from “Skating” to “Chess”.

>db.student.updateOne({name: 'AryanDavid',grade:'A',hobbies:['Reading','Skating']},{$set:{

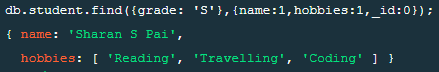
'hobbies.1':'Playing Chess',

}},{upsert: true});

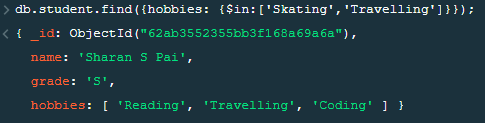


1. Find collection based on some search criteria

>db.student.find({grade: ‘S’},{name:1,hobbies:1,\_id:0});



1. To find those documents from the Students collection where the Hobbies is set to either ‘Travelling’ or is set to ‘Skating’.



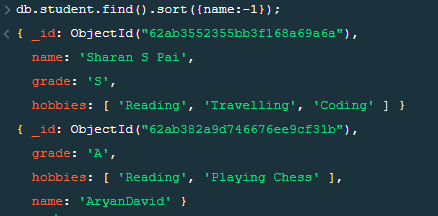
1. To find documents from the Students collection where the StudName begins with “A”.



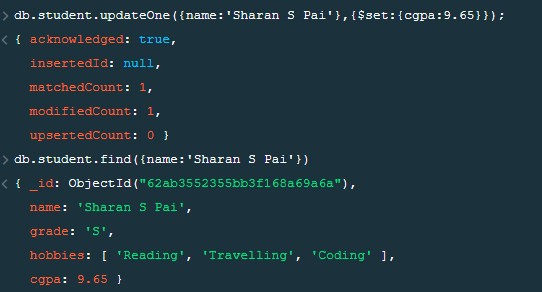
1. To find the number of documents in the Students collection.



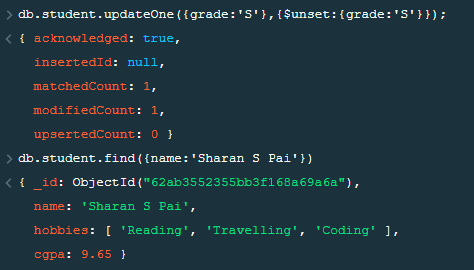
1. To sort the documents from the Students collection in the descending order of StudName



1. Add a new field to existing document



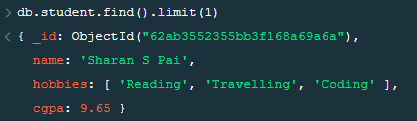
1. Remove a field from existing document



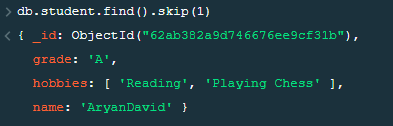
1. Count the document that have travelling as their hobby



1. Read only 1st document



1. Read all documents by skipping over 1st document



# 2. Perform the following DB operations using Cassandra.

**(Employee DB)**

**1. Create keyspace**

CREATE KEYSPACE employee;

**2. create employee table**

create table employee\_in( emp\_id int, emp\_name text, desig text, dateofjoin date, salary float, dept text, PRIMARY KEY(emp\_id,salary));

**3. Insert data interms of batches**

BEGIN BATCH

... INSERT INTO employee\_in(emp\_id,emp\_name,desig,dateofjoin,salary,dept) VALUES (1,'Karan

... ','Manager','2021-06-12',1000000,'HR')

... INSERT INTO employee\_in(emp\_id,emp\_name,desig,dateofjoin,salary,dept) VALUES (2,'Rajesh Meheta','associate SE','2020-04-20',600000,'Tech')

... INSERT INTO employee\_in(emp\_id,emp\_name,desig,dateofjoin,salary,dept) VALUES (3,'Vihnu Chauhan','associate SE','2020-05-20',600000,'Tech')

... INSERT INTO employee\_in(emp\_id,emp\_name,desig,dateofjoin,salary,dept) VALUES (4,'Shweta Tripathi','associate SE','2020-05-20',600000,'Tech')

... APPLY BATCH

... ;

**4. Update one emp\_name and dept in the table**

select \* from employee\_in

... ;

emp\_id | salary | dateofjoin | dept | desig | emp\_name

--------+--------+------------+------+--------------+-----------------

1 | 1e+06 | 2021-06-12 | HR | Manager | Karan\n

2 | 6e+05 | 2020-04-20 | Tech | associate SE | Rajesh Meheta

4 | 6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi

3 | 6e+05 | 2020-05-20 | Tech | associate SE | Vihnu Chauhan

update employee\_in SET emp\_name='Vishnu Chauhan',dept='Technical' where emp\_id=3 AND salary=600000;

select \* from employee\_in ;

emp\_id | salary | dateofjoin | dept | desig | emp\_name

--------+--------+------------+-----------+--------------+-----------------

1 | 1e+06 | 2021-06-12 | HR | Manager | Karan\n

2 | 6e+05 | 2020-04-20 | Tech | associate SE | Rajesh Meheta

4 | 6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi

3 | 6e+05 | 2020-05-20 | Technical | associate SE | Vishnu Chauhan

**5. SORT the entire employee table on Salary**

select \* from employee\_in where emp\_id IN(1,2,3,4) ORDER BY salary DESC allow filtering;

emp\_id | salary | dateofjoin | dept | desig | emp\_name

--------+--------+------------+-----------+--------------+-----------------

1 | 1e+06 | 2021-06-12 | HR | Manager | Karan\n

2 | 6e+05 | 2020-04-20 | Tech | associate SE | Rajesh Meheta

3 | 6e+05 | 2020-05-20 | Technical | associate SE | Vishnu Chauhan

4 | 6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi

**6. Add projects column to the table**

ALTER TABLE employee\_in ADD projects list<text>;

select \* from employee\_in ;

emp\_id | salary | dateofjoin | dept | desig | emp\_name | projects

--------+--------+------------+-----------+--------------+-----------------+----------

1 | 1e+06 | 2021-06-12 | HR | Manager | Karan\n | null

2 | 6e+05 | 2020-04-20 | Tech | associate SE | Rajesh Meheta | null

4 | 6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi | null

3 | 6e+05 | 2020-05-20 | Technical | associate SE | Vishnu Chauhan | null

**7. update the projects in the table**

>update employee\_in SET projects=['CCF','CCD','KMAP'] where emp\_id=3 AND salary=600000;

>update employee\_in SET projects=['AAP','BJP','TMC'] where emp\_id=4 AND salary=600000;

>select \* from employee\_in ;

emp\_id | salary | dateofjoin | dept | desig | emp\_name | projects

--------+--------+------------+-----------+--------------+-----------------+------------------------

1 | 1e+06 | 2021-06-12 | HR | Manager | Karan\n | null

2 | 6e+05 | 2020-04-20 | Tech | associate SE | Rajesh Meheta | null

4 | 6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi | ['AAP', 'BJP', 'TMC']

3 | 6e+05 | 2020-05-20 | Technical | associate SE | Vishnu Chauhan | ['CCF', 'CCD', 'KMAP']

**8. CREATE TTL of 15 seconds to display values of employees**

>update employee\_in USING TTL 15 SET emp\_name='Karan Sharma' where emp\_id=1 AND salary=600000;

//BEFORE 15 seconds

> select \* from employee\_in ;

emp\_id | salary | dateofjoin | dept | desig | emp\_name | projects

--------+--------+------------+-----------+--------------+-----------------+------------------------

1 | 6e+05 | null | null | null | Karan Sharma | null

1 | 1e+06 | 2021-06-12 | HR | Manager | Karan\n | null

2 | 6e+05 | 2020-04-20 | Tech | associate SE | Rajesh Meheta | null

4 | 6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi | ['AAP', 'BJP', 'TMC']

3 | 6e+05 | 2020-05-20 | Technical | associate SE | Vishnu Chauhan | ['CCF', 'CCD', 'KMAP']

//AFTER 15 seconds

> select \* from employee\_in ;

emp\_id | salary | dateofjoin | dept | desig | emp\_name | projects

--------+--------+------------+-----------+--------------+-----------------+------------------------

1 | 1e+06 | 2021-06-12 | HR | Manager | Karan\n | null

2 | 6e+05 | 2020-04-20 | Tech | associate SE | Rajesh Meheta | null

4 | 6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi | ['AAP', 'BJP', 'TMC']

3 | 6e+05 | 2020-05-20 | Technical | associate SE | Vishnu Chauhan | ['CCF', 'CCD', 'KMAP']

(4 rows)

# 3. Perform the following DB operations using Cassandra.

**(Library DB)**

Perform following operation using CASSANDRA on library database

1. Create a keyspace library

> CREATE KEYSPACE library WITH REPLICATION={

‘class’: ‘SimpleStrategy’,

‘replication\_factor’: 3

};

2. Create a column family by name lib\_info:

> CREATE TABLE lib\_info(

sid int,

c\_val counter,

sname text,

bname text,

bid int,

doi date,

PRIMARY KEY(sid,sname,bname,bid,doi)

);

> DESCRIBE lib\_info;

OUTPUT:

CREATE TABLE library.lib\_info (

sid int,

sname text,

bname text,

bid int,

doi date,

c\_val counter,

PRIMARY KEY (sid, sname, bname, bid, doi)

) WITH CLUSTERING ORDER BY (sname ASC, bname ASC, bid ASC, doi ASC)

AND additional\_write\_policy = '99p'

AND bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND cdc = false

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND default\_time\_to\_live = 0;

3) Insert values in batches:

Ans. Not possible to insert (update doesn’t work with batches)

ERROR: InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot include a counter statement in a logged batch"

>UPDATE lib\_info SET c\_val=c\_val+1 WHERE sid=110 and sname=’Sharan’ AND bname=’BDA’ AND bid=120 AND doi=’2022-01-27’ ;

> UPDATE lib\_info SET c\_val=c\_val+1 WHERE sid=112 and sname=’Varun’ AND bname=’CNS’ AND bid=110 AND doi=’2022-01-27’ ;

> UPDATE lib\_info SET c\_val=c\_val+1 WHERE sid=112 and sname=’Varun’ AND bname=’CNS’ AND bid=110 AND doi=’2022-01-27’ ;

> SELECT \* FROM lib\_info;

sid | sname | bname | bid | doi | c\_val

-----+--------+-------+-----+------------+-------

110 | Sharan | BDA | 120 | 2022-01-27 | 1

112 | Varun | CNS | 110 | 2022-01-27 | 2

4. Display details of table and increase counter

> SELECT \* FROM lib\_info;

sid | sname | bname | bid | doi | c\_val

-----+--------+-------+-----+------------+-------

110 | Sharan | BDA | 120 | 2022-01-27 | 1

112 | Varun | CNS | 110 | 2022-01-27 | 2

> UPDATE lib\_info SET c\_val=c\_val+1 WHERE sid=112 and sname=’Varun’ AND bname=’CNS’ AND bid=110 AND doi=’2022-01-27’ ;

> SELECT \* FROM lib\_info;

sid | sname | bname | bid | doi | c\_val

-----+--------+-------+-----+------------+-------

110 | Sharan | BDA | 120 | 2022-01-27 | 1

112 | Varun | CNS | 110 | 2022-01-27 | 3

5. Write a query to show student with id 112 has taken CNS 2 times

> SELECT \* FROM lib\_info WHERE sid=112 AND c\_val>=2 AND sname='Karan' AND bname='CNS' AND bid=110 AND doi='2022-01-27' ALLOW FILTERING;

sid | sname | bname | bid | doi | c\_val

-----+--------+-------+-----+------------+-------

112 | Varun | CNS | 110 | 2022-01-27 | 3

6. Export created column family to csv file

> COPY lib\_info(sid,sname,c\_val,bid,bname,doi) TO ‘./lib.csv’;

Using 1 child processes

Starting copy of library.lib\_info with columns [sid, sname, c\_val, bid, bname, doi].

Processed: 2 rows; Rate: 11 rows/s; Avg. rate: 3 rows/s

2 rows exported to 1 files in 0.827 seconds.

7. Import csv dataset from local FS to cassandra

> COPY lib\_info(sid,sname,c\_val,bid,bname,doi) FROM ‘./lib.csv’;

Using 1 child processes

Starting copy of library.lib\_info with columns [sid, sname, c\_val, bid, bname, doi].

Processed: 2 rows; Rate: 1 rows/s; Avg. rate: 1 rows/s

2 rows imported from 1 files in 0 day, 0 hour, 0 minute, and 1.400 seconds (0 skipped).

> SELECT \* FROM LIB\_INFO;

sid | sname | bname | bid | doi | c\_val

-----+--------+-------+-----+------------+-------

110 | Sharan | BDA | 120 | 2022-01-27 | 2

112 | Varun | CNS | 110 | 2022-01-27 | 6