VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS
(20CS6PEBDA)

Submitted by

KIRAN M K (1BM19CS073)

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 **KIRAN M K(1BM19CS073),** 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 Laboratory - (20CS6PEBDA)** work prescribed for the said degree.

Dr. Pallavi G. B.
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index Sheet

Sl. No.	Experiment Title	Page No.
1	Cassandra Lab Program 1: - Create a Data set either structured/Semi-Structured/Unstructured from Twitter/Facebook etc. to perform various DB operations using Cassandra. (Use the Face Pager app to perform real-time streaming)	1-3
2	Cassandra Lab Program 2: - Create a Data set either structured/Semi-Structured/Unstructured from Twitter/Facebook etc. to perform various DB operations using Cassandra. (Use the Face Pager app to perform real-time streaming)	4-6
3	MongoDB Lab Program 1 (CRUD Demonstration): - Students should be classifying a dataset into one of the standard forms and apply suitable querying rules to obtain suitable results	7-11
4	MongoDB Lab Program 2 (CRUD Demonstration): - Students should be classifying a dataset into one of the standard forms and apply suitable querying rules to obtain suitable results	12-17

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

Cassandra Program 1

Program 1. Perform the following DB operations using Cassandra.

1. Create a key space by name Employee

```
cqlsh> CREATE KEYSPACE employee WITH replication = {'class':'SimpleStrategy', 'replication_factor':1};
cqlsh> describe keyspace
No keyspace specified and no current keyspace
cqlsh> describe Employee;

CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}  AND durable_writes = true;
cqlsh> create table Employee.employee_info(emp_id int Primary Key, emp_name text, designation text, date_of_joining timestamp, salary double,
dept_name text);
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining | dept_name | designation | emp_name | salary
--------+-----------------+-----------+-------------+----------+--------
```

2. Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name,
Designation, Date_of_Joining, Salary, Dept_Name

```
cqlsh> begin batch insert into Employee.employee_info(emp_id, date_of_joining, dept_name, designation, emp_name, salary) values (1, '2021-06-0
3', 'Deploymemt', 'Manager', 'KIRAN', 1500000.50); apply batch;
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining                 | dept_name  | designation | emp_name | salary
--------+---------------------------------+------------+-------------+----------+--------
      1 | 2021-06-02 18:30:00.000000+0000 | Deployment |     Manager |    KIRAN | 1.5e+06

(1 rows)
cqlsh> begin batch
   ... insert into Employee.employee_info(emp_id, date_of_joining, dept_name, designation, emp_name, salary) values (2, '2020-09-03', 'Develop
ment', 'Web Developer', 'VAISHAK', 1700000.50);
   ... insert into Employee.employee_info(emp_id, date_of_joining, dept_name, designation, emp_name, salary) values (121, '2019-05-03', 'Resea
rch and Development', 'Intern', 'VISHVESH', 2000000.50);
   ... apply batch;
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining                 | dept_name                | designation   | emp_name | salary
--------+---------------------------------+--------------------------+---------------+----------+---------
      1 | 2021-06-02 18:30:00.000000+0000 |               Deployment |       Manager |    KIRAN | 1.5e+06
      2 | 2020-09-02 18:30:00.000000+0000 |              Development | Web Developer |  VAISHAK | 1.7e+06
    121 | 2019-05-02 18:30:00.000000+0000 | Research and Development |        Intern | VISHVESH |   2e+06

(3 rows)
```

3. Insert the values into the table in batch

```
cqlsh> update Employee.employee_info SET emp_name = 'PRAMOD', dept_name = 'Testing' WHERE emp_id = 121;
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining                 | dept_name   | designation   | emp_name | salary
--------+---------------------------------+-------------+---------------+----------+---------
      1 | 2021-06-02 18:30:00.000000+0000 |  Deployment |       Manager |    KIRAN | 1.5e+06
      2 | 2020-09-02 18:30:00.000000+0000 | Development | Web Developer |  VAISHAK | 1.7e+06
    121 | 2019-05-02 18:30:00.000000+0000 |     Testing |        Intern |   PRAMOD |   2e+06
```

4. Update Employee name and Department of Emp-Id 121

```
cqlsh> create table Employee.emp(emp_id int, emp_name text, designation text, date_of_joining timestamp, salary double, dept_name text, prim
y key(emp_id, salary) );
cqlsh> begin batch
   ... insert into Employee.emp(emp_id, salary, date_of_joining, dept_name, designation, emp_name) values (1, 1500000.50, '2021-06-03', 'Dep
yment', 'Manager', 'KIRAN');
   ... insert into Employee.emp(emp_id, salary,date_of_joining, dept_name, designation, emp_name) values (2, 1100000.50, '2022-05-03', 'Deve
pment', 'Web Developer', 'VAISHAK');
   ... insert into Employee.emp(emp_id, salary, date_of_joining, dept_name, designation, emp_name) values (121, 1900000.50, '2022-05-05', 'R
earch and Development', 'Intern', 'VISHVESH');
   ... apply batch;
cqlsh> select * from Employee.emp;

 emp_id | salary  | date_of_joining                 | dept_name               | designation   | emp_name
--------+---------+---------------------------------+-------------------------+---------------+----------
      1 | 1.5e+06 | 2021-06-02 18:30:00.000000+0000 |              Deployment |       Manager |    KIRAN
      2 | 1.1e+06 | 2022-05-02 18:30:00.000000+0000 |             Development | Web Developer |  VAISHAK
    121 | 1.9e+06 | 2022-05-04 18:30:00.000000+0000 | Research and Development |        Intern | VISHVESH

(3 rows)
cqlsh> alter table Employee.employee_info add projects set<text>;
cqlsh> select * from Employee.emp;

 emp_id | salary  | date_of_joining                 | dept_name               | designation   | emp_name
--------+---------+---------------------------------+-------------------------+---------------+----------
      1 | 1.5e+06 | 2021-06-02 18:30:00.000000+0000 |              Deployment |       Manager |    KIRAN
      2 | 1.1e+06 | 2022-05-02 18:30:00.000000+0000 |             Development | Web Developer |  VAISHAK
    121 | 1.9e+06 | 2022-05-04 18:30:00.000000+0000 | Research and Development |        Intern | VISHVESH
```

5. Sort the details of Employee records based on salary

```
cqlsh> update Employee.employee_info set projects = projects + {'Hospital Management', 'Payment Interface'} where emp_id = 1;
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining                 | dept_name   | designation   | emp_name | projects                                          | salary
--------+---------------------------------+-------------+---------------+----------+---------------------------------------------------+---------
      1 | 2021-06-02 18:30:00.000000+0000 |  Deployment |       Manager |    KIRAN | {'Hospital Management', 'Payment Interface'} | 1.5e+06
      2 | 2020-09-02 18:30:00.000000+0000 | Development | Web Developer |  VAISHAK |                                              null | 1.7e+06
    121 | 2019-05-02 18:30:00.000000+0000 |     Testing |        Intern |   PRAMOD |                                              null |   2e+06

(3 rows)
cqlsh> update Employee.employee_info set projects = projects + {'Lab', 'Helper'} where emp_id = 121;
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining                 | dept_name   | designation   | emp_name | projects                                          | salary
--------+---------------------------------+-------------+---------------+----------+---------------------------------------------------+---------
      1 | 2021-06-02 18:30:00.000000+0000 |  Deployment |       Manager |    KIRAN | {'Hospital Management', 'Payment Interface'} | 1.5e+06
      2 | 2020-09-02 18:30:00.000000+0000 | Development | Web Developer |  VAISHAK |                                              null | 1.7e+06
    121 | 2019-05-02 18:30:00.000000+0000 |     Testing |        Intern |   PRAMOD |                                 {'Helper', 'Lab'} |   2e+06
```

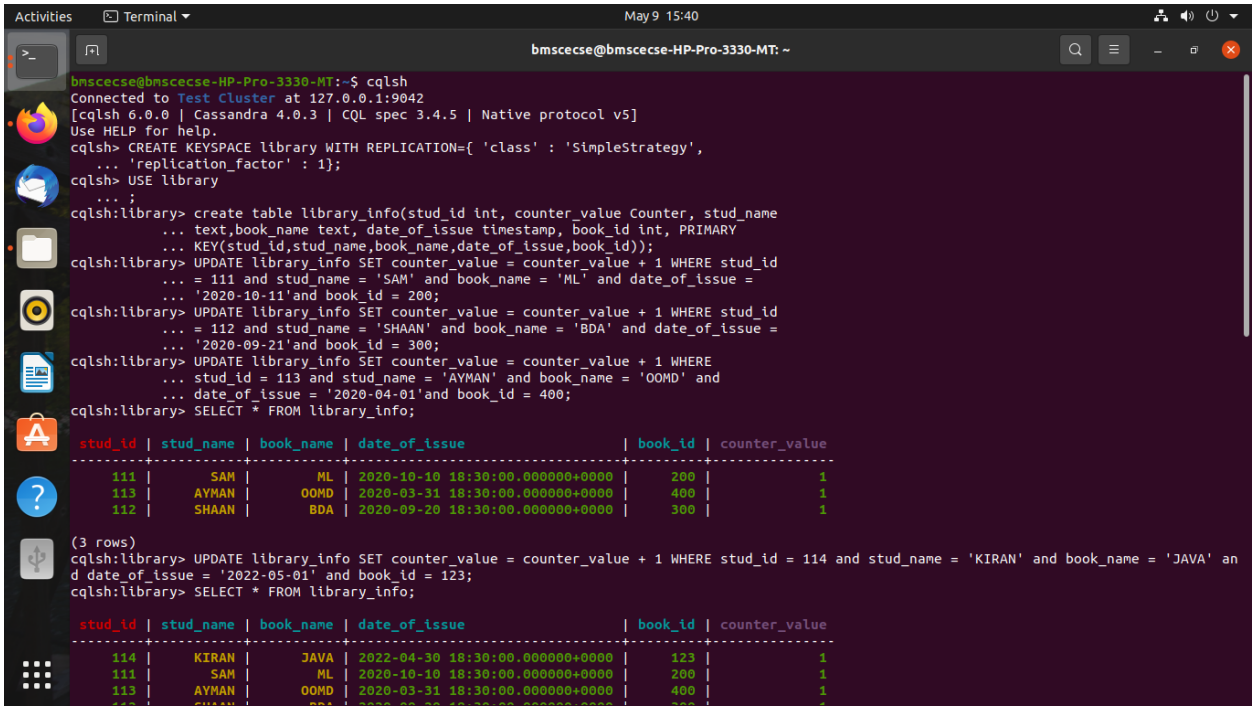
6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

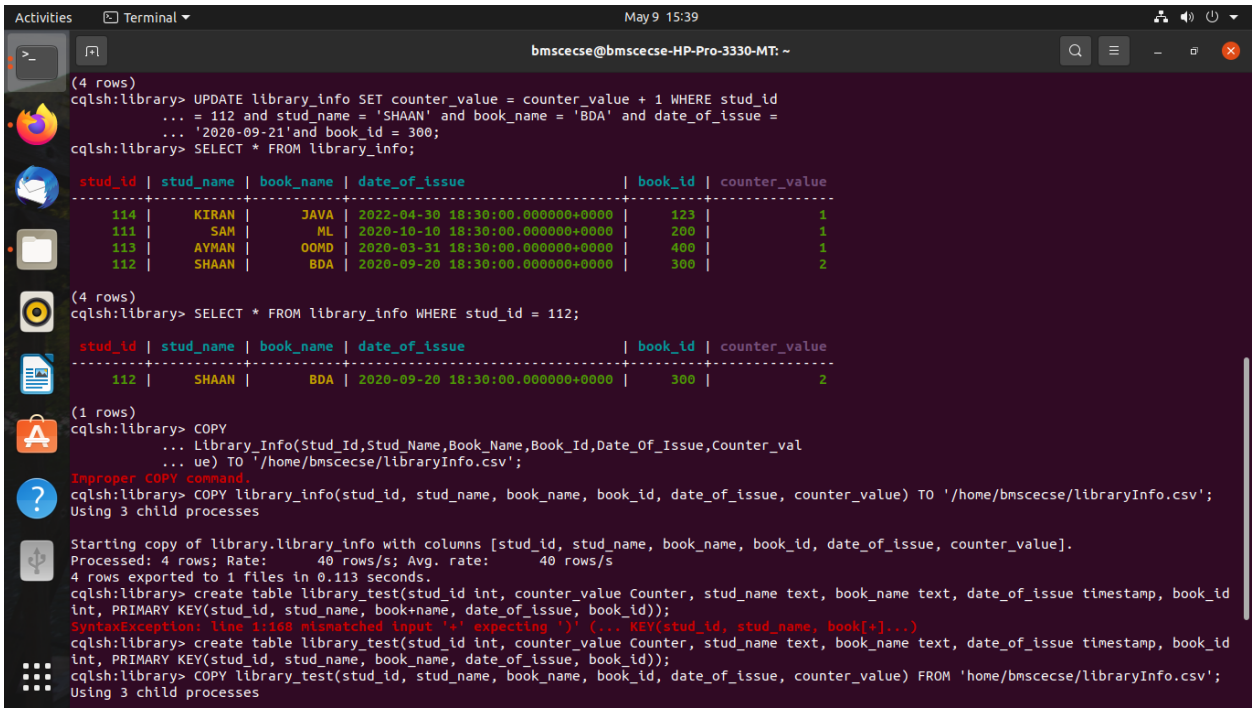
```
cqlsh> insert into Employee.employee_info(emp_id, date_of_joining, dept_name, designation, emp_name, salary) values (11, '2019-05-05', 'Resear
ch and Development', 'Intern', 'SHREYAS', 1000000.50) using TTL 15;
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining                 | dept_name               | designation   | emp_name | projects
 | salary
--------+---------------------------------+-------------------------+---------------+----------+-----------------------------------------
-+---------
     11 | 2019-05-04 18:30:00.000000+0000 | Research and Development |        Intern |  SHREYAS |                                      null
 |   1e+06
      1 | 2021-06-02 18:30:00.000000+0000 |              Deployment |       Manager |    KIRAN | {'Hospital Management', 'Payment Interface'}
 | 1.5e+06
      2 | 2020-09-02 18:30:00.000000+0000 |             Development | Web Developer |  VAISHAK |                                      null
 | 1.7e+06
    121 | 2019-05-02 18:30:00.000000+0000 |                 Testing |        Intern |   PRAMOD |                         {'Helper', 'Lab'}
 |   2e+06

(4 rows)
cqlsh> select * from Employee.employee_info;

 emp_id | date_of_joining                 | dept_name   | designation   | emp_name | projects                                          | salary
--------+---------------------------------+-------------+---------------+----------+---------------------------------------------------+---------
      1 | 2021-06-02 18:30:00.000000+0000 |  Deployment |       Manager |    KIRAN | {'Hospital Management', 'Payment Interface'} | 1.5e+06
      2 | 2020-09-02 18:30:00.000000+0000 | Development | Web Developer |  VAISHAK |                                              null | 1.7e+06
    121 | 2019-05-02 18:30:00.000000+0000 |     Testing |        Intern |   PRAMOD |                                 {'Helper', 'Lab'} |   2e+06
```

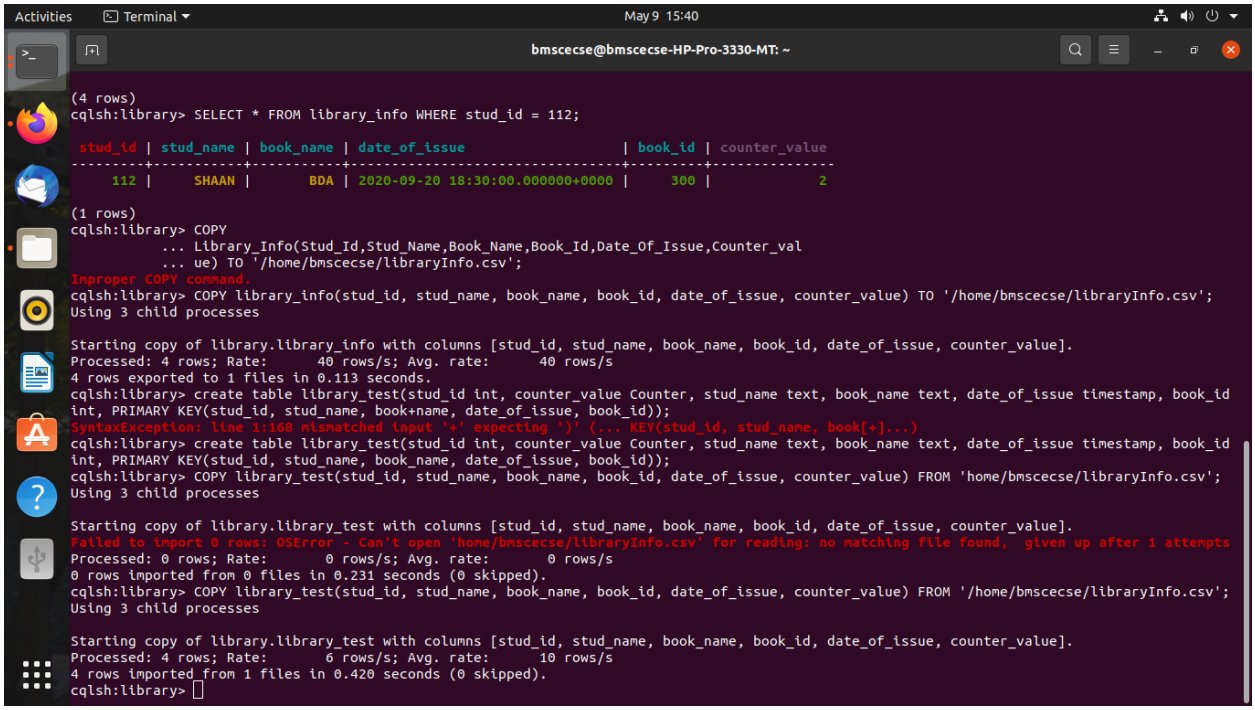
7. Update the altered table to add project names.

```
cqlsh> update Employee.employee_info set projects = projects + {'API', 'Mood detection'} where emp_id = 2;
cqlsh> paging off;
Disabled Query paging.
cqlsh> select * from Employee.emp where emp_id in (1,2,121) order by salary;

 emp_id | salary  | date_of_joining                 | dept_name              | designation    | emp_name
--------+---------+---------------------------------+------------------------+----------------+----------
      2 | 1.1e+06 | 2022-05-02 18:30:00.000000+0000 |             Development | Web Developer |  VAISHAK
      1 | 1.5e+06 | 2021-06-02 18:30:00.000000+0000 |              Deployment |       Manager |     KIRAN
    121 | 1.9e+06 | 2022-05-04 18:30:00.000000+0000 | Research and Development |        Intern | VISHVESH
```

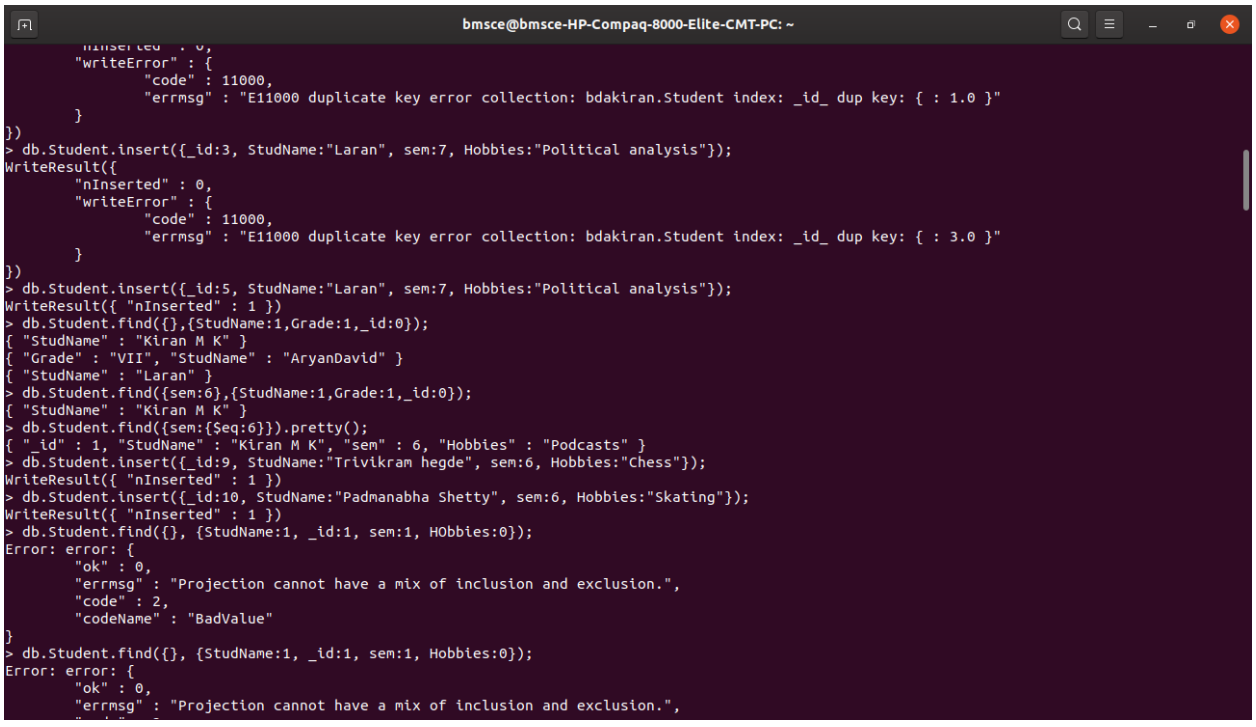
Cassandra Program 2


Activities
Terminal
May 9 15:40
bmscecse@bmscecse-HP-Pro-3330-MT: ~
bmscecse@bmscecse-HP-Pro-3330-MT:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.3 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE library WITH REPLICATION={ 'class' : 'SimpleStrategy',
... 'replication_factor' : 1};
cqlsh> USE library
... ;
cqlsh:library> create table library_info(stud_id int, counter_value Counter, stud_name
... text,book_name text, date_of_issue timestamp, book_id int, PRIMARY
... KEY(stud_id,stud_name,book_name,date_of_issue,book_id));
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id
... = 111 and stud_name = 'SAM' and book_name = 'ML' and date_of_issue =
... '2020-10-11'and book_id = 200;
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id
... = 112 and stud_name = 'SHAAN' and book_name = 'BDA' and date_of_issue =
... '2020-09-21'and book_id = 300;
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1 WHERE
... stud_id = 113 and stud_name = 'AYMAN' and book_name = 'OOMD' and
... date_of_issue = '2020-04-01'and book_id = 400;
cqlsh:library> SELECT * FROM library_info;
stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
111 | SAM | ML | 2020-10-10 18:30:00.000000+0000 | 200 | 1
113 | AYMAN | OOMD | 2020-03-31 18:30:00.000000+0000 | 400 | 1
112 | SHAAN | BDA | 2020-09-20 18:30:00.000000+0000 | 300 | 1
(3 rows)
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id = 114 and stud_name = 'KIRAN' and book_name = 'JAVA' an
d date_of_issue = '2022-05-01' and book_id = 123;
cqlsh:library> SELECT * FROM library_info;
stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
114 | KIRAN | JAVA | 2022-04-30 18:30:00.000000+0000 | 123 | 1
111 | SAM | ML | 2020-10-10 18:30:00.000000+0000 | 200 | 1
113 | AYMAN | OOMD | 2020-03-31 18:30:00.000000+0000 | 400 | 1


Activities
Terminal
May 9 15:39
bmscecse@bmscecse-HP-Pro-3330-MT: ~
(4 rows)
cqlsh:library> UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id
... = 112 and stud_name = 'SHAAN' and book_name = 'BDA' and date_of_issue =
... '2020-09-21'and book_id = 300;
cqlsh:library> SELECT * FROM library_info;
stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
114 | KIRAN | JAVA | 2022-04-30 18:30:00.000000+0000 | 123 | 1
111 | SAM | ML | 2020-10-10 18:30:00.000000+0000 | 200 | 1
113 | AYMAN | OOMD | 2020-03-31 18:30:00.000000+0000 | 400 | 1
112 | SHAAN | BDA | 2020-09-20 18:30:00.000000+0000 | 300 | 2
(4 rows)
cqlsh:library> SELECT * FROM library_info WHERE stud_id = 112;
stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
112 | SHAAN | BDA | 2020-09-20 18:30:00.000000+0000 | 300 | 2
(1 rows)
cqlsh:library> COPY
... Library_Info(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_val
... ue) TO '/home/bmscecse/libraryInfo.csv';
Improper COPY command.
cqlsh:library> COPY library_info(stud_id, stud_name, book_name, book_id, date_of_issue, counter_value) TO '/home/bmscecse/libraryInfo.csv';
Using 3 child processes
Starting copy of library.library_info with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 4 rows; Rate: 40 rows/s; Avg. rate: 40 rows/s
4 rows exported to 1 files in 0.113 seconds.
cqlsh:library> create table library_test(stud_id int, counter_value Counter, stud_name text, book_name text, date_of_issue timestamp, book_id
int, PRIMARY KEY(stud_id, stud_name, book+name, date_of_issue, book_id));
SyntaxException: line 1:168 mismatched input '+' expecting ')' (... KEY(stud_id, stud_name, book[+]...)
cqlsh:library> create table library_test(stud_id int, counter_value Counter, stud_name text, book_name text, date_of_issue timestamp, book_id
int, PRIMARY KEY(stud_id, stud_name, book_name, date_of_issue, book_id));
cqlsh:library> COPY library_test(stud_id, stud_name, book_name, book_id, date_of_issue, counter_value) FROM 'home/bmscecse/libraryInfo.csv';
Using 3 child processes


Activities
Terminal
May 9 15:40
bmscecse@bmscecse-HP-Pro-3330-MT: ~
(4 rows)
cqlsh:library> SELECT * FROM library_info WHERE stud_id = 112;
stud_id | stud_name | book_name | date_of_issue | book_id | counter_value
112 | SHAAN | BDA | 2020-09-20 18:30:00.000000+0000 | 300 | 2
(1 rows)
cqlsh:library> COPY
... Library_Info(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_val
... ue) TO '/home/bmscecse/libraryInfo.csv';
Improper COPY command.
cqlsh:library> COPY library_info(stud_id, stud_name, book_name, book_id, date_of_issue, counter_value) TO '/home/bmscecse/libraryInfo.csv';
Using 3 child processes
Starting copy of library.library_info with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 4 rows; Rate: 40 rows/s; Avg. rate: 40 rows/s
4 rows exported to 1 files in 0.113 seconds.
cqlsh:library> create table library_test(stud_id int, counter_value Counter, stud_name text, book_name text, date_of_issue timestamp, book_id int, PRIMARY KEY(stud_id, stud_name, book+name, date_of_issue, book_id));
SyntaxException: line 1:168 mismatched input '+' expecting ')' (... KEY(stud_id, stud_name, book[+]...)
cqlsh:library> create table library_test(stud_id int, counter_value Counter, stud_name text, book_name text, date_of_issue timestamp, book_id int, PRIMARY KEY(stud_id, stud_name, book_name, date_of_issue, book_id));
cqlsh:library> COPY library_test(stud_id, stud_name, book_name, book_id, date_of_issue, counter_value) FROM 'home/bmscecse/libraryInfo.csv';
Using 3 child processes
Starting copy of library.library_test with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Failed to import 0 rows: OSError - Can't open 'home/bmscecse/libraryInfo.csv' for reading: no matching file found, given up after 1 attempts
Processed: 0 rows; Rate: 0 rows/s; Avg. rate: 0 rows/s
0 rows imported from 0 files in 0.231 seconds (0 skipped).
cqlsh:library> COPY library_test(stud_id, stud_name, book_name, book_id, date_of_issue, counter_value) FROM '/home/bmscecse/libraryInfo.csv';
Using 3 child processes
Starting copy of library.library_test with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 4 rows; Rate: 6 rows/s; Avg. rate: 10 rows/s
4 rows imported from 1 files in 0.420 seconds (0 skipped).
cqlsh:library>

Mongo DB Program 1



```
> db.Student.find({StudName:/M/}).pretty();
{ "_id" : 1, "StudName" : "Kiran M K", "sem" : 6, "Hobbies" : "Podcasts" }
> db.Student.find({StudName:/M/i}).pretty();
{ "_id" : 1, "StudName" : "Kiran M K", "sem" : 6, "Hobbies" : "Podcasts" }
{
        "_id" : 9,
        "StudName" : "Trivikram hegde",
        "sem" : 6,
        "Hobbies" : "Chess"
}
{
        "_id" : 10,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Skating"
}
> db.Student.find({StudName:/[mM]/}).pretty();
{ "_id" : 1, "StudName" : "Kiran M K", "sem" : 6, "Hobbies" : "Podcasts" }
{
        "_id" : 9,
        "StudName" : "Trivikram hegde",
        "sem" : 6,
        "Hobbies" : "Chess"
}
{
        "_id" : 10,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Skating"
```

```
> db.Student.find({Hobbies:{$eq:'Chess'}}).pretty();
{
        "_id" : 3,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Chess"
}
> db.Student.find({Hobbies :{ $in: ['Political analysis','Skating']}}).pretty ();
{
        "_id" : 2,
        "StudName" : "Trivikram Hegde",
        "sem" : 6,
        "Hobbies" : "Political analysis"
}
{ "_id" : 4, "StudName" : "Laran", "sem" : 6, "Hobbies" : "Skating" }
> db.Student.find({StudName:/^P/}).pretty();
{
        "_id" : 3,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Chess"
}
> db.Student.find({StudName:/e/}).pretty();
{
        "_id" : 2,
        "StudName" : "Trivikram Hegde",
        "sem" : 6,
        "Hobbies" : "Political analysis"
}
{
        "_id" : 3,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Chess"
}
```

```
> db.Student.find({_id:1},{StudName:1,sem:1,_id:0});
{ "StudName" : "Kiran", "sem" : 6 }
> db.Student.find({Hobbies:{$ne:'Chess'}}).pretty();
{ "_id" : 1, "StudName" : "Kiran", "sem" : 6, "Hobbies" : "Podcasts" }
{
        "_id" : 2,
        "StudName" : "Trivikram Hegde",
        "sem" : 6,
        "Hobbies" : "Political analysis"
}
{
        "_id" : 4,
        "StudName" : "Laran",
        "sem" : 6,
        "Hobbies" : "Skating",
        "Location" : "Bengaluru"
}
{ "_id" : ObjectId("62543111e29a7583f87aa599"), "StudName" : "Kiran" }
> db.Student.update({_id:3},{$set:{Location:null}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.count();
5
> db.Student.find({sem:6}).limit(3).pretty();
{ "_id" : 1, "StudName" : "Kiran", "sem" : 6, "Hobbies" : "Podcasts" }
{
        "_id" : 2,
        "StudName" : "Trivikram Hegde",
        "sem" : 6,
        "Hobbies" : "Political analysis"
}
{
        "_id" : 3,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Chess",
        "Location" : null
}
```

```
> db.Student.count();
4
> db.Student.find().sort({StudName:-1}).pretty();
{
       "_id" : 2,
       "StudName" : "Trivikram Hegde",
       "sem" : 6,
       "Hobbies" : "Political analysis"
}
{
       "_id" : 3,
       "StudName" : "Padmanabha Shetty",
       "sem" : 6,
       "Hobbies" : "Chess"
}
{ "_id" : 4, "StudName" : "Laran", "sem" : 6, "Hobbies" : "Skating" }
{ "_id" : 1, "StudName" : "Kiran", "sem" : 6, "Hobbies" : "Podcasts" }
> db.Student.save({StudName:"Kiran"})
WriteResult({ "nInserted" : 1 })
> db.Student.find();
{ "_id" : 1, "StudName" : "Kiran", "sem" : 6, "Hobbies" : "Podcasts" }
{ "_id" : 2, "StudName" : "Trivikram Hegde", "sem" : 6, "Hobbies" : "Political analysis" }
{ "_id" : 3, "StudName" : "Padmanabha Shetty", "sem" : 6, "Hobbies" : "Chess" }
{ "_id" : 4, "StudName" : "Laran", "sem" : 6, "Hobbies" : "Skating" }
{ "_id" : ObjectId("62543111e29a7583f87aa599"), "StudName" : "Kiran" }
> db.Student.update({_id:4},{$set:{Location:"Bengaluru"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find();
{ "_id" : 1, "StudName" : "Kiran", "sem" : 6, "Hobbies" : "Podcasts" }
{ "_id" : 2, "StudName" : "Trivikram Hegde", "sem" : 6, "Hobbies" : "Political analysis" }
{ "_id" : 3, "StudName" : "Padmanabha Shetty", "sem" : 6, "Hobbies" : "Chess" }
{ "_id" : 4, "StudName" : "Laran", "sem" : 6, "Hobbies" : "Skating", "Location" : "Bengaluru" }
{ "_id" : ObjectId("62543111e29a7583f87aa599"), "StudName" : "Kiran" }
```

```
> db.Student.find().sort({StudName:1}).pretty();
{ "_id" : 1, "StudName" : "Kiran", "sem" : 6, "Hobbies" : "Podcasts" }
{ "_id" : ObjectId("62543111e29a7583f87aa599"), "StudName" : "Kiran" }
{
        "_id" : 4,
        "StudName" : "Laran",
        "sem" : 6,
        "Hobbies" : "Skating",
        "Location" : "Bengaluru"
}
{
        "_id" : 3,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Chess",
        "Location" : null
}
{
        "_id" : 2,
        "StudName" : "Trivikram Hegde",
        "sem" : 6,
        "Hobbies" : "Political analysis"
}
> db.Student.find().skip(2).pretty();
{
        "_id" : 3,
        "StudName" : "Padmanabha Shetty",
        "sem" : 6,
        "Hobbies" : "Chess",
        "Location" : null
}
{
        "_id" : 4,
        "StudName" : "Laran",
        "sem" : 6,
        "Hobbies" : "Skating",
        "Location" : "Bengaluru"
```

```
> db.createCollection("food");
{ "ok" : 1 }
> db.food.insert( { _id:1, fruits:['grapes','mango','apple'] } )
WriteResult({ "nInserted" : 1 })
> db.food.insert( { _id:2, fruits:['grapes','mango','cherry'] } )
WriteResult({ "nInserted" : 1 })
> db.food.insert( { _id:3, fruits:['banana','mango'] } )
WriteResult({ "nInserted" : 1 })
> db.food.find( {fruits: ['grapes','mango','apple'] } ).pretty();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
> db.food.find( {'fruits.1':'grapes'} )
> ;
> db.food.find({'fruits.1':'grapes'});
> db.food.find( {"fruits": {$size:2}} );
{ "_id" : 3, "fruits" : [ "banana", "mango" ] }
> db.food.find({_id:1},{"fruits":{$slice:2}})
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }
> db.food.find({fruits:{$all:[“mango”,”grapes”]}})
uncaught exception: SyntaxError: illegal character :
@(shell):1:28
> db.food.find({fruits:{$all:["mango", "grapes"]}});
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
> db.food.update({_id:3},{$set:{'fruits.1':'apple'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.food.find();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
> db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.food.find();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
```

```
> db.food.insertOne( { _id: 5, fruits: [ "mango", "apple", "orange" ] } );
{ "acknowledged" : true, "insertedId" : 5 }
> db.food.updateOne( { _id: 5 }, { $pop: { fruits: -1 } } );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.food.find();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
{ "_id" : 5, "fruits" : [ "apple", "orange" ] }
```

```
> db.createCollection("Customers");
{ "ok" : 1 }
> db.Customers.insert({_id:1, custID:1, AccBal: 15000, AccType: "Savings"});
WriteResult({ "nInserted" : 1 })
> db.Customers.insert({_id:2, custID:1, AccBal: 16000, AccType: "Savings"});
WriteResult({ "nInserted" : 1 })
> db.Customers.insert({_id:3, custID:2, AccBal: 18000, AccType: "Savings"});
WriteResult({ "nInserted" : 1 })
> db.Customers.insert({_id:4, custID:2, AccBal: 20000, AccType: "Savings"});
WriteResult({ "nInserted" : 1 })
> db.Customers.aggregate( {$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } } );
{ "_id" : 2, "TotAccBal" : 38000 }
{ "_id" : 1, "TotAccBal" : 31000 }
> db.Customers.insert({custID:5, AccBal:10000000, AccType: "Current"});
WriteResult({ "nInserted" : 1 })
> db.Customers.aggregate( {$match:{AcctType:"Savings"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } } );
> db.Customers.aggregate( {$match:{AccType:"Savings"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } } );
{ "_id" : 2, "TotAccBal" : 38000 }
{ "_id" : 1, "TotAccBal" : 31000 }
> db.Customers.find();
{ "_id" : 1, "custID" : 1, "AccBal" : 15000, "AccType" : "Savings" }
{ "_id" : 2, "custID" : 1, "AccBal" : 16000, "AccType" : "Savings" }
{ "_id" : 3, "custID" : 2, "AccBal" : 18000, "AccType" : "Savings" }
{ "_id" : 4, "custID" : 2, "AccBal" : 20000, "AccType" : "Savings" }
{ "_id" : ObjectId("625448f0ee5e75021943909a"), "custID" : 5, "AccBal" : 10000000, "AccType" : "Current" }
> db.Customers.insert({custID: 3, AccBal:1000, AccType: "Savings"});
WriteResult({ "nInserted" : 1 })
> db.Customers.find();
{ "_id" : 1, "custID" : 1, "AccBal" : 15000, "AccType" : "Savings" }
{ "_id" : 2, "custID" : 1, "AccBal" : 16000, "AccType" : "Savings" }
{ "_id" : 3, "custID" : 2, "AccBal" : 18000, "AccType" : "Savings" }
{ "_id" : 4, "custID" : 2, "AccBal" : 20000, "AccType" : "Savings" }
{ "_id" : ObjectId("625448f0ee5e75021943909a"), "custID" : 5, "AccBal" : 10000000, "AccType" : "Current" }
{ "_id" : ObjectId("62544974ee5e75021943909b"), "custID" : 3, "AccBal" : 1000, "AccType" : "Savings" }
```

```
> db.food.updateOne({_id:1},{$addToSet:{fruits:"banana"}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.food.find();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple", "banana" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
{ "_id" : 5, "fruits" : [ "apple", "orange" ] }
> db.food.updateOne({_id:1}, {$pullAll:{fruits:["mango", "apple"]}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.food.find();
{ "_id" : 1, "fruits" : [ "grapes", "banana" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
{ "_id" : 5, "fruits" : [ "apple", "orange" ] }
> db.food.updateMany({},{$pull:{fruits:{$in:["mango", "grapes"]}}});
{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 2 }
> db.food.find();
{ "_id" : 1, "fruits" : [ "banana" ] }
{ "_id" : 2, "fruits" : [ "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
{ "_id" : 5, "fruits" : [ "apple", "orange" ] }
```

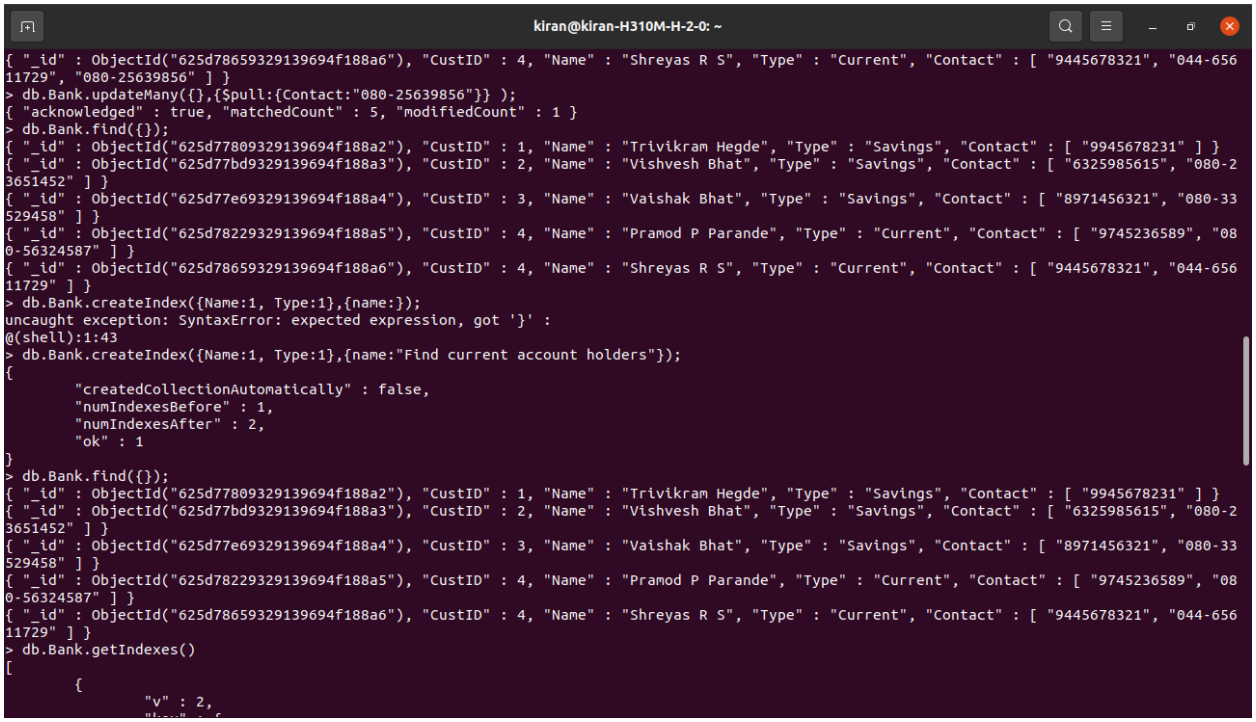
Mongo DB Program 2

```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ sudo mongoexport --host localhost --db 1bm19cs073_kiranmk --collection Student --type=csv  --out /home/bms
ce/Desktop/output.txt --fields "_id","Name","USN","Sem","Dept","CGPA","Hobbies"
2022-04-18T15:13:05.991+0530    connected to: mongodb://localhost/
2022-04-18T15:13:05.996+0530    exported 6 records
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ls
024  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
bmscecse@bmscecse-HP-Pro-3330-MT:~$ cd Desktop
bmscecse@bmscecse-HP-Pro-3330-MT:~/Desktop$ ls
bmscecse@bmscecse-HP-Pro-3330-MT:~/Desktop$ mongoexport --host localhost --db 1bm19cs073_kiranmk --collection Student --csv  --out /home/bmsce
/Desktop/output.txt --fields "_id","Name","USN","Sem","Dept","CGPA","Hobbies"
2022-04-18T15:14:16.323+0530    csv flag is deprecated; please use --type=csv instead
2022-04-18T15:14:16.325+0530    connected to: mongodb://localhost/
2022-04-18T15:14:16.325+0530    error opening output stream: mkdir /home/bmsce/Desktop: permission denied
bmscecse@bmscecse-HP-Pro-3330-MT:~/Desktop$ sudo mongoexport --host localhost --db 1bm19cs073_kiranmk --collection Student --csv  --out /home/
bmsce/Desktop/output.txt --fields "_id","Name","USN","Sem","Dept","CGPA","Hobbies"
2022-04-18T15:14:22.162+0530    csv flag is deprecated; please use --type=csv instead
2022-04-18T15:14:22.164+0530    connected to: mongodb://localhost/
2022-04-18T15:14:22.169+0530    exported 6 records
bmscecse@bmscecse-HP-Pro-3330-MT:~/Desktop$ pwd
/home/bmscecse/Desktop
bmscecse@bmscecse-HP-Pro-3330-MT:~/Desktop$ sudo mongoexport --host localhost --db 1bm19cs073_kiranmk --collection Student --type=csv  --out /
home/bmscecse/Desktop/output.txt --fields "_id","Name","USN","Sem","Dept","CGPA","Hobbies"
2022-04-18T15:15:10.012+0530    connected to: mongodb://localhost/
2022-04-18T15:15:10.017+0530    exported 6 records
bmscecse@bmscecse-HP-Pro-3330-MT:~/Desktop$ cd ..
bmscecse@bmscecse-HP-Pro-3330-MT:~$ ls
024  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
bmscecse@bmscecse-HP-Pro-3330-MT:~$ cd ..
bmscecse@bmscecse-HP-Pro-3330-MT:/home$ ls
bmsce  bmscecse  lost+found
bmscecse@bmscecse-HP-Pro-3330-MT:/home$ cd bmsce
bash: cd: bmsce: Permission denied
bmscecse@bmscecse-HP-Pro-3330-MT:/home$ sudo cd bmsce
sudo: cd: command not found
bmscecse@bmscecse-HP-Pro-3330-MT:/home$ login bmsce
login: Cannot possibly work without effective root
bmscecse@bmscecse-HP-Pro-3330-MT:/home$ cd bmscecse
bmscecse@bmscecse-HP-Pro-3330-MT:~$
```

```
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use 1bm19cs073_kiranmk;
switched to db 1bm19cs073_kiranmk
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({_id:1, Name:"Kodandaraman C V", USN:"1BM19CS078", Sem: 6, Dept: "CSE", CGPA: 9.4, Hobbies:"Politics"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:2, Name:"Shrinivas Shastry P", USN:"1BM19CS077", Sem: 6, Dept: "CSE", CGPA: 8.3, Hobbies:"YouTuber"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:3, Name:"H N Subramanyam", USN:"1BM19IS025", Sem: 6, Dept: "ISE", CGPA: 9.6, Hobbies:"History of India"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:4, Name:"J Ashwani Kumar", USN:"1BM19IS050", Sem: 6, Dept: "ISE", CGPA: 9.8, Hobbies:"Quantum Physics"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:5, Name:"Sonar H Maruti", USN:"1BM19CS075", Sem: 6, Dept: "CSE", CGPA: 9.2, Hobbies:"Organic Chemistry"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:5, Name:"H K Venkatesh Bhat", USN:"1BM20CS081", Sem: 4, Dept: "CSE", CGPA: 9.8, Hobbies:"Multivariable Vector Calculu
s"});
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 11000,
                "errmsg" : "E11000 duplicate key error collection: 1bm19cs073_kiranmk.Student index: _id_ dup key: { _id: 5.0 }"
        }
})
> db.Student.insert({_id:6, Name:"H K Venkatesh Bhat", USN:"1BM20CS081", Sem: 4, Dept: "CSE", CGPA: 9.8, Hobbies:"Multivariable Vector Calculu
s"});
WriteResult({ "nInserted" : 1 })
> db.Customers.aggregate ( {$match:{AcctType:"S"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } }, {$match:{TotAccBal:{$gt:1200}
}});
uncaught exception: SyntaxError: illegal character :
@(shell):1:43
> db.Student.aggregate({$match:{Dept:"CSE"}}, {$group:{Sem:[4,6], AvgCGPA:{$avg:"$CGPA"}}}, {$match:{AvgCGPA:{$gt:7.5}}});
uncaught exception: Error: command failed: {
        "ok" : 0,
```

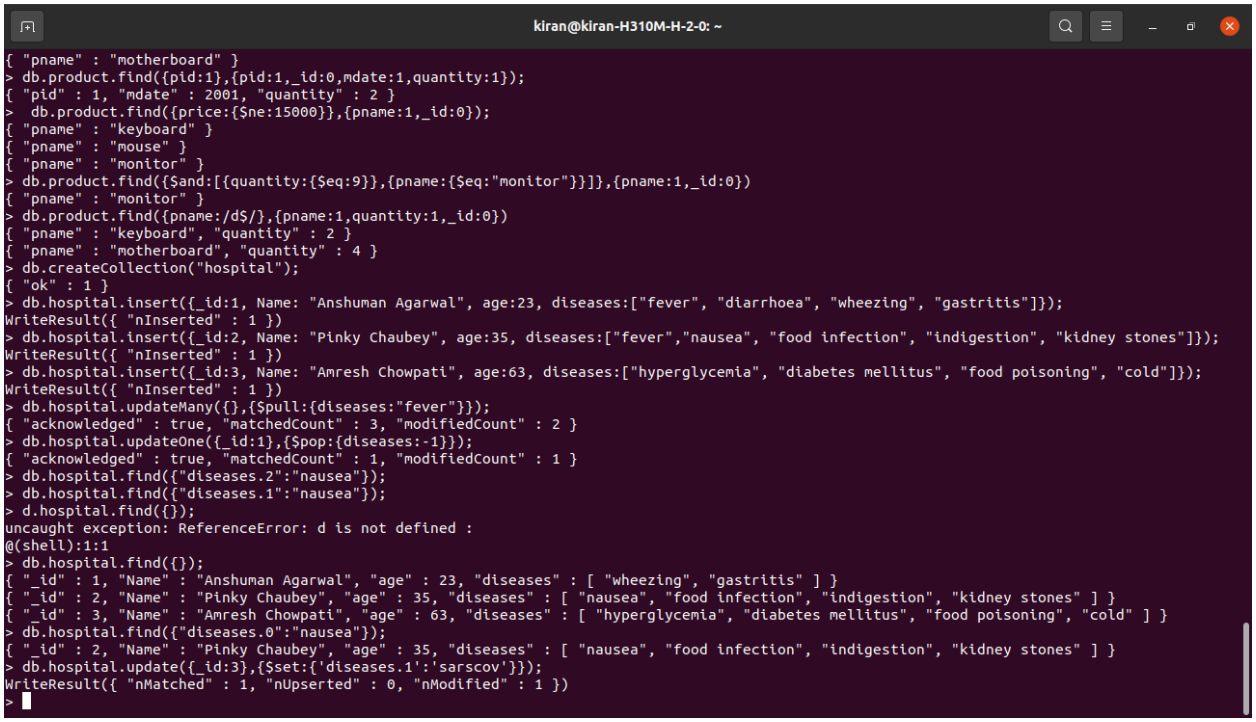
```
bmscecse@bmscecse-HP-Pro-3330-MT: ~

        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use 1bm19cs073_kiranmk;
switched to db 1bm19cs073_kiranmk
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({_id:1, Name:"Kodandaraman C V", USN:"1BM19CS078", Sem: 6, Dept: "CSE", CGPA: 9.4, Hobbies:"Politics"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:2, Name:"Shrinivas Shastry P", USN:"1BM19CS077", Sem: 6, Dept: "CSE", CGPA: 8.3, Hobbies:"YouTuber"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:3, Name:"H N Subramanyam", USN:"1BM19IS025", Sem: 6, Dept: "ISE", CGPA: 9.6, Hobbies:"History of India"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:4, Name:"J Ashwani Kumar", USN:"1BM19IS050", Sem: 6, Dept: "ISE", CGPA: 9.8, Hobbies:"Quantum Physics"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:5, Name:"Sonar H Maruti", USN:"1BM19CS075", Sem: 6, Dept: "CSE", CGPA: 9.2, Hobbies:"Organic Chemistry"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:5, Name:"H K Venkatesh Bhat", USN:"1BM20CS081", Sem: 4, Dept: "CSE", CGPA: 9.8, Hobbies:"Multivariable Vector Calculu
s"});
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 11000,
                "errmsg" : "E11000 duplicate key error collection: 1bm19cs073_kiranmk.Student index: _id_ dup key: { _id: 5.0 }"
        }
})
> db.Student.insert({_id:6, Name:"H K Venkatesh Bhat", USN:"1BM20CS081", Sem: 4, Dept: "CSE", CGPA: 9.8, Hobbies:"Multivariable Vector Calculu
s"});
WriteResult({ "nInserted" : 1 })
> db.Customers.aggregate ( {$match:{AcctType:"S"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } }, {$match:{TotAccBal:{$gt:1200}
}});
uncaught exception: SyntaxError: illegal character :
@(shell):1:43
> db.Student.aggregate({$match:{Dept:"CSE"}}, {$group:{Sem:[4,6], AvgCGPA:{$avg:"$CGPA"}}}, {$match:{AvgCGPA:{$gt:7.5}}});
uncaught exception: Error: command failed: {
        "ok" : 0,
```

```
kiran@kiran-H310M-H-2-0: ~

> db.createCollection("Bank");
{ "ok" : 1 }
> db.insert({CustID:1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]});
uncaught exception: TypeError: db.insert is not a function :
@(shell):1:1
> db.Bank.insert({CustID:1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:2, Name:"Vishvesh Bhat", Type:"Savings", Contact:["6325985615", "080-23651452"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:3, Name:"Vaishak Bhat", Type:"Savings", Contact:["8971456321", "080-33529458"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:4, Name:"Pramod P Parande", Type:"Current", Contact:["9745236589", "080-56324587"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:4, Name:"Shreyas R S", Type:"Current", Contact:["9445678321","044-65611729", "080-25639856"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231", "080
-22364587" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-2
3651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33
529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "08
0-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-656
11729", "080-25639856" ] }
> db.Bank.updateMany({CustID:1},{$pop:{Contact:1}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-2
3651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33
529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "08
0-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-656
11729", "080-25639856" ] }
```


kiran@kiran-H310M-H-2-0: ~
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : ["9445678321", "044-656
11729", "080-25639856"] }
> db.Bank.updateMany({},{$pull:{Contact:"080-25639856"}});
{ "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 1 }
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : ["9945678231"] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : ["6325985615", "080-2
3651452"] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : ["8971456321", "080-33
529458"] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : ["9745236589", "08
0-56324587"] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : ["9445678321", "044-656
11729"] }
> db.Bank.createIndex({Name:1, Type:1},{name:});
uncaught exception: SyntaxError: expected expression, got '}' :
@(shell):1:43
> db.Bank.createIndex({Name:1, Type:1},{name:"Find current account holders"});
{
"createdCollectionAutomatically" : false,
"numIndexesBefore" : 1,
"numIndexesAfter" : 2,
"ok" : 1
}
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : ["9945678231"] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : ["6325985615", "080-2
3651452"] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : ["8971456321", "080-33
529458"] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : ["9745236589", "08
0-56324587"] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : ["9445678321", "044-656
11729"] }
> db.Bank.getIndexes()
[
{
"v" : 2,

```
kiran@kiran-H310M-H-2-0: ~
@(shell):1:20
> db.Bank.update({_id:625d78659329139694f188a6}, {$set: {CustID:5}},{upsert:true});
uncaught exception: SyntaxError: identifier starts immediately after numeric literal :
@(shell):1:20
> db.Bank.update({_id:"625d78659329139694f188a6"}, {$set: {CustID:5}},{upsert:true});
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 1,
        "nModified" : 0,
        "_id" : "625d78659329139694f188a6"
})
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-2
3651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33
529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "08
0-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-656
11729" ] }
{ "_id" : "625d78659329139694f188a6", "CustID" : 5 }
> db.Bank.update({_id:"625d78659329139694f188a6", CustID:5}, {$set: {Name:"Sumantha K S", Type:"Savings", Contact:["9856321478","011-65897458"
]}},{upsert:true});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-2
3651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33
529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "08
0-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-656
11729" ] }
{ "_id" : "625d78659329139694f188a6", "CustID" : 5, "Contact" : [ "9856321478", "011-65897458" ], "Name" : "Sumantha K S", "Type" : "Savings"
}
>
```

```
kiran@kiran-H310M-H-2-0: ~
}
> db.createCollection("faculty");
{ "ok" : 1 }
> db.faculty.insert({_id:1,name:"Dr. Balaraman Ravindran",designation:"Professor",department:"CSE",age:45,salary:100000,specialization:['pytho
n','mysql','sklearn', 'tensorflow']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:2,name:"Dr. Mahadev Ghorki",designation:"Assistant Professor",department:"CSE",age:35,salary:80000,specialization:['p
ython','numpy','sklearn', 'tensorflow', 'java']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:3,name:"Dr. Praveen Borade",designation:"Associate Professor",department:"ME",age:40,salary:75000,specialization:['au
tocad', 'aerodynamics', 'thermal physics']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:4,name:"Dr. Madhav Nayak",designation:"Assistant Professor",department:"ME",age:37,salary:95000,specialization:['auto
cad', 'flight-dynamics', 'Finite Element Analysis']});
WriteResult({ "nInserted" : 1 })
> db.faculty.aggregate ( {$match:{department:"ME"}}, {$group : {_id : "$designation", AverageSal :{$avg:"$salary"} } },  {$match:{AverageSal:{
$gt:50000}}});
{ "_id" : "Associate Professor", "AverageSal" : 75000 }
{ "_id" : "Assistant Professor", "AverageSal" : 95000 }
> db.createCollection("product");
{ "ok" : 1 }
> db.product.insert({pid:1,pname:"keyboard",mdate:2001,price:1800,quantity:2});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:2,pname:"mouse",mdate:2005,price:1500,quantity:5});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:3,pname:"monitor",mdate:2015,price:10000,quantity:9});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:4,pname:"motherboard",mdate:2021,price:15000,quantity:4});
WriteResult({ "nInserted" : 1 })
> db.product.find({},{pname:1,_id:0})
{ "pname" : "keyboard" }
{ "pname" : "mouse" }
{ "pname" : "monitor" }
{ "pname" : "motherboard" }
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
>  db.product.find({price:{$ne:15000}},{pname:1,_id:0});
{ "pname" : "keyboard" }
```


kiran@kiran-H310M-H-2-0: ~
{ "pname" : "motherboard" }
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
> db.product.find({price:{$ne:15000}},{pname:1,_id:0});
{ "pname" : "keyboard" }
{ "pname" : "mouse" }
{ "pname" : "monitor" }
> db.product.find({$and:[{quantity:{$eq:9}},{pname:{$eq:"monitor"}}]},{pname:1,_id:0})
{ "pname" : "monitor" }
> db.product.find({pname:/d$/},{pname:1,quantity:1,_id:0})
{ "pname" : "keyboard", "quantity" : 2 }
{ "pname" : "motherboard", "quantity" : 4 }
> db.createCollection("hospital");
{ "ok" : 1 }
> db.hospital.insert({_id:1, Name: "Anshuman Agarwal", age:23, diseases:["fever", "diarrhoea", "wheezing", "gastritis"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.insert({_id:2, Name: "Pinky Chaubey", age:35, diseases:["fever","nausea", "food infection", "indigestion", "kidney stones"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.insert({_id:3, Name: "Amresh Chowpati", age:63, diseases:["hyperglycemia", "diabetes mellitus", "food poisoning", "cold"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.updateMany({},{$pull:{diseases:"fever"}});
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 2 }
> db.hospital.updateOne({_id:1},{$pop:{diseases:-1}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.hospital.find({"diseases.2":"nausea"});
> db.hospital.find({"diseases.1":"nausea"});
> d.hospital.find({});
uncaught exception: ReferenceError: d is not defined :
@(shell):1:1
> db.hospital.find({});
{ "_id" : 1, "Name" : "Anshuman Agarwal", "age" : 23, "diseases" : ["wheezing", "gastritis"] }
{ "_id" : 2, "Name" : "Pinky Chaubey", "age" : 35, "diseases" : ["nausea", "food infection", "indigestion", "kidney stones"] }
{ "_id" : 3, "Name" : "Amresh Chowpati", "age" : 63, "diseases" : ["hyperglycemia", "diabetes mellitus", "food poisoning", "cold"] }
> db.hospital.find({"diseases.0":"nausea"});
{ "_id" : 2, "Name" : "Pinky Chaubey", "age" : 35, "diseases" : ["nausea", "food infection", "indigestion", "kidney stones"] }
> db.hospital.update({_id:3},{$set:{'diseases.1':'sarscov'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>