VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS
(20CS6PEBDA)

Submitted by

Chaitanya Gadgil (1BM19CS223)

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 **Chaitanya Gadgil(1BM19CS223),** 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 Choudhury
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

`

Index Sheet

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 – LAB 1

```
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing"});
WriteResult({ "nInserted" : 1 })
> db.Student.update({_id:1},{$set:{hobbies:"cricket"}},{upsert:true})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find()
{ "_id" : 1, "StudName" : "MichelleJacintha", "Grade" : "VII", "Hobbies" : "InternetSurfing", "hobbies" : "cricket" }
> db.Student.insert({id:1,name:"xyz",grade:"VIII",hobbies:"chess"})
WriteResult({ "nInserted" : 1 })
> db.Student.find({name:/xyz/}).pretty()
{
    "_id" : ObjectId("6256987834dadfe4d50f9d70"),
    "id" : 1,
    "name" : "xyz",
    "grade" : "VIII",
    "hobbies" : "chess"
}
> db.Student.find().sort({name:1}).pretty()
{
    "_id" : 1,
    "StudName" : "MichelleJacintha",
    "Grade" : "VII",
    "Hobbies" : "InternetSurfing",
    "hobbies" : "cricket"
}
{
    "_id" : ObjectId("6256987834dadfe4d50f9d70"),
    "id" : 1,
    "name" : "xyz",
    "grade" : "VIII",
    "hobbies" : "chess"
}
> db.Student.find().skip(1).pretty()
{
    "_id" : ObjectId("6256987834dadfe4d50f9d70"),
    "id" : 1,
    "name" : "xyz",
    "grade" : "VIII",
    "hobbies" : "chess"
}
> db.createCollection("food")
{ "ok" : 1 }
> db.food.insert({_id:1,fruits:['grapes','mango']})
WriteResult({ "nInserted" : 1 })
> db.food.insert({_id:2,fruits:['grapes','mango','cherry']})
WriteResult({ "nInserted" : 1 })
> db.food.insert({_id:3,fruits:['banana','cherry']})
WriteResult({ "nInserted" : 1 })
> db.food.find({fruits:['grapes','mango']})
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }
> db.food.find({'fruits':{$size:2}})
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }
{ "_id" : 3, "fruits" : [ "banana", "cherry" ] }
> db.food.find({_id:2},{'fruits':{$slice:2}})
{ "_id" : 2, "fruits" : [ "grapes", "mango" ] }
> db.food.find({fruits:{$all:['grapes','mango']}})
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
> db.food.update({_id:3},{$set:{'fruits.1':'apple'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.food.find()
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
> db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.createCollection("Customers")
{ "ok" : 1 }
> db.Customers.insert({custId:1,acctBal:1000,acctType:"current"})
WriteResult({ "nInserted" : 1 })
> db.Customers.insert({custId:2,acctBal:2000,acctType:"current"})
WriteResult({ "nInserted" : 1 })
> db.Customers.insert({custId:3,acctBal:3000,acctType:"savings"})
WriteResult({ "nInserted" : 1 })
> db.Customers.aggregate({$group:{_id:"$custId",toAcctBal:{$sum:"$acctBal"}}})
{ "_id" : 3, "toAcctBal" : 3000 }
{ "_id" : 1, "toAcctBal" : 1000 }
{ "_id" : 2, "toAcctBal" : 2000 }
> db.Customers.aggregate({$match:{acctType:"current"}},{$group:{_id:"$custId",toAcctBal:{$sum:"$acctBal"}}})
{ "_id" : 2, "toAcctBal" : 2000 }
{ "_id" : 1, "toAcctBal" : 1000 }
> db.Customers.aggregate({$match:{acctType:"current"}},{$group:{_id:"$custId",toAcctBal:{$sum:"$acctBal"}}},
{$match:{toAcctBal:{$gt:500}}})
{ "_id" : 2, "toAcctBal" : 2000 }
{ "_id" : 1, "toAcctBal" : 1000 }
```

MongoDB – LAB 2

```
> db.createCollection("Bank");
> {"ok": 1 }
> db.insert({CustID:1, Name:"Trivikran Hegde, Type:"Savings", Contact:["9945678231",
"080-22364587"}});
> db.Bank.Insert({CustID:1, Nane: "Trtvtkran Hegde, Type:"Savings",
Contact:["9945678231", "060-22364587"]}); writeResult([ 'nInserted": 1})
> db.Bank. Insert({CustID:2, Name: "Vishvesh Bhat", Type:"Savings,
Contact:["6325985615", "000-23651452"]}); WriteResult("ninserted": 1 })
> db.Bank. Insert({CustID:3, Name: "Vaishak that", Type:"Savings",
Contact:["8971456321", "000-13529458"]}); 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 5, Type:"Current",
Contact:["9445678321","044-65611729", "080-25639856"]});
> WriteResult({nInserted": 1})
> db.Bank.find({});
{_id: ObjectId("62587780932913969471882"), "CustID" : 1, "Name": "Trivikran Hegde",
"Type": "Savings", "Contact" : [ "9945078231", "080-223645871"]}
{_id: ObjectId("625d77bd9329139694f188a3"), "CustID": 2, "Name": "Vishvesh Bhat",
"Type": "Savings", "Contact" : [ "6325985615", "080-23651452"]}
{_id: ObjectId("625d77e693291396941884"), "CustID" : 3, "Name": "Vatshak Bhat",
"Type": "Savings", "Contact" : ["8971456321", "080-335294581]
{_id: ObjectId("6254782293291398 94f188a5"), "CustID" : 4, "Name": "Praned P Parande",
"Type" : "Current", "Contact" : [ "9745236589", "080-56324587"]}
{_id: ObjectId("625d78659329139894f188a6"), "CustID" : 4, "Name": "shreyas R 5", "Type"
: "Current", "Contact" : [ "9445678321", "044-65011729"]}
> db.Bank.updateMany({CustID:1},{$pop: {Contact:1}});
{acknowledged": true, "natchedCount": 1, "nodifiedCount": 1}
> db.Bank.find();
{_id: ObjectId("625d7709329139694f188a2"), "CustIo": 1, "Name": "Trivikran Hegde",
"Type": "Savings", "Contact":"9945678231" }
{_id: ObjectId>("625d77bd9329139694f18a3"), "CustID": 2, "Name": "Vishvesh Bhat",
"Type": "Savings", "Contact" : [ "Savings", "Contact" : [ "6325985615", "010-23651452"]}
{_id: ObjectId("625d77e69329139894 71884"), "CustID" : 3, "Name": "Vaishak Bhat",
"Type": "Savings", "Contact":["8971456321", "080-3529458"]}
{_id ObjectId("625d78229329139694 7188a5"), "CustID" : 4, "Name": "Pramod P Parande",
"Type" : "Current", "Contact" : [ "9745236589", "080-56324587"]}
{_id: ObjectId("625d78659329139694 7188a6"), "CustID": 4, "Name": "Shreyas R S",
"Type" : "Current", "Contact" : [ "9445678321", "044-65611729"]}
>db.Bank.updateMany({CustID: 1}), {$pull: {Contact:"000-25639856"}} };
acknowledged": true, "natchedCount": 5, "modifiedCount": 1 }
>db.Bank.find({});
{_id: ObjectId("625d77809329139694f18882"), "CustID" : 1, "Name": "Trivikram Hegde",
"Type": "Savings", "Contact" ["9945678231" ]},
{_id: ObjectId("625877bd9329139694f1983), "CustID": 2, "Name": "Vishvesh Bhat",
"Type": "Savings", "Contact": ["6325985615", "080-23651452"]}
{_id: ObjectId("625677e69329139694718804"), "CustID": 3, "Name": "Vaishak Bhat",
"Type": "Savings", "Contact":["8971456321", "080-33529458"]}
{_id ObjectId("625d78229329139694 71885), "CustID": 4, "Name": "Pranod Parande",
"Type" : "Current", "Contact" : ["9745236589", "080-563245871"]}
{_id: ObjectId("625678659329139694188a6"), "CustID": 4, "Name": "Shreyas RS", "Type" :
"Current", "Contact" : [ 9445678321", "044-65011729"]}
```

```
>db.Bank.createIndex({Name:1, Type:1}, {name:});
uncaught exception: SyntaxError: expected expression, got '}'
(shell)11:43
db.Bank.createIndex({Name:1, Type:1}, {name:"Find current account holders"});
{
"createdCollectionAutomatically":false,
"nunIndexesBefore": 1,
TounIndexesAfter": 2,
"ok": 1
}
>db.Bank.find({});
{_id: ObjectId("62567708932913969410882"), "CustID": 1, "Name": "Trivikram Hegde"
"Type": "Savings", "Contact":"9945678231"]}
{_id: ObjectId("625477bd932913969410883"), "CustID": 2, "Name": "Vishvesh Bhat",
"Type": "Savings", "Contact" : ["6325985615", "080-23651452"]
{_id: ObjectId("625d77e59329139694718884"), "CustID": 3, "Name": "Vatshak Bhat",
"Type": "Savings", "Contact: [ "8971456321", "080-33529458"]}
{_id: ObjectId("6254782293291396944f188a5″), "CustID" : 4, "Name": "Pramod P Parande",
"Type" : "Current", "Contact" : [ "9745236589", "080-56324587"]}
{_id: ObjectId("62587865932913969471880б"), "CustID": 4, "Name": "Shreyas RS", "Type"
: "Current", "Contact" : [ "9445678321", "044-65611729"]}
>db.Bank.getIndexes()
> db.Bank.update({id:625078659329139694F188a6), ($set: {CustID:53}, {upsert:true});
uncaught exception: SyntaxError: identifier starts innediately after > numeric literal:
(shell):1:20
> db.Bank.update({id:"62585932913941"}, {$set: {CustID:5}}, {upsert:true});
writeResult({
nhatched":0,
"nUpserted" 11,
"Modified": 0,
"id":"625d78659329139694f18826"
})
> db.Bank.find({});
{_id: ObjectId("625d7700932913909418882"), "CustID" : 1, "Name": "Trivikran Hegde",
"Type": "Savings", "Contact":"9945678231" ] }
{_id: ObjectId("625477bd9329139094F188a3"), "CustID": 2, "Name": "Vishvesh Bhat",
"Type": "Savings. "Contact" : ["0325981615", "O80-36514521"]}
{_id: ObjectId("625d77e693291396947188a4"), "CustID": 3, "Name": "Valshak Bhat",
"Type": "Savings", "Contact": [ "8971456321", "080-33529458"]}
{_id: ObjectId("825878229329139694188a5"), "CustID: 4, "Name": "Pramod P Parande",
"Type" : "Current", "Contact" : [ "9745236589", "080-56324587"]}
{_id: ObjectId("625d766593291390941886"), "CustID": 4, "Name": "Shreyas R S", "Type" :
"Current", "Contact" : [ "9445678323", "044-65611729"]}
> db.Bank.update({_id:"625078659329139694718 8a6", CustID:4}, {$set: {Name:"Sumantha
K 5, Type:"Savings", Contact:["9856325478","11-65897458"]},{upsert:true});
WriteResult("Matched" : 1, "nupserted": 6, "Modified":1})
> db.Bank.find({});
{_id: ObjectId("625d7780932913909418882"), "CustID": 1, "Name": "Trivikran Hegde",
"Type": "Savings", "Contact" : [ "9945678231"] }
{_id: ObjectId("625d77bd9329139694f188a3"), "CustID": 2, "Name": "Vishvesh Bhat",
"Type": "Savings, "Contact" : ["6325985615", "080-36514529"]}
{_id: ObjectId("825d77e6932913969418844"), "CustID" : 3, "Name": "Vaishak Bhat",
"Type": "Savings", "Contact" : [ "8971456321", "080-34529458"]}
{_id: ObjectId("625d78229329139094F188a5"), "CustID" : 4, "Name": "Pranod P Parande",
"Type" : "Current", "Contact" : [ "9745236589", "080-56324587"]}
{(id: ObjectId("625d78659329139694f188a6"), "CustID: 4, "Name": "Sumantha x 5",
```

"Type": "Savings", "Contact" : ["9445678321", "044-05611729"]}

Lab 3 - Cassandra

Perform the following DB operations using Cassandra.

1.Create a keyspace by name Employee

2. Create a column family by name

Employee-Info with attributes

Emp_Id Primary Key, Emp_Name,

Designation, Date_of_Joining,

Salary, Dept_Name

3. Insert the values into the table in batch

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

5. Sort the details of Employee records based on salary

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.

7. Update the altered table to add project names.

8.Create a TTL of 15 seconds to display the values of Employee

```
cqlsh> create keyspace mployee_space WITH REPLICATION = {'class' : 'SimpleStrategy','replication_factor':2};

CREATE TABLE employee_space.employee_info (emp_id int PRIMARY KEY,emp_name text,designation
text,date_of_joining timestamp,salary float,dept_name text);

cqlsh> begin batch INSERT INTO
employee_space.employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
VALUES(1,'abc','Manager','2022-01-24',100000,'Marketing');
   ... apply batch;
cqlsh> begin batch INSERT INTO
employee_space.employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
VALUES(2,'pqr','Accountant','2021-01-24',200000,'Accounts');
   ... INSERT INTO
employee_space.employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
VALUES(3,'xyz','Manager','2021-03-24',500000,'Marketing');
   ... INSERT INTO
employee_space.employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
VALUES(4,'ijk','Administrator','2021-05-24',500000,'Administration');
```

```
   ... INSERT INTO
employee_space.employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
VALUES(5,'lmn','Administrator','2009-05-24',2000000,'Administration');
   ... apply batch;

cqlsh> use employee_space;

cqlsh:employee_space> select * from employee_info;

 emp_id | date_of_joining                 | dept_name      | designation   | emp_name  | salary
--------+---------------------------------+----------------+---------------+-----------+--------
      5 | 2009-05-23 18:30:00.000000+0000 | Administration | Administrator |   lmn     |  2e+06
      1 | 2022-01-23 18:30:00.000000+0000 |      Marketing |       Manager |   abc     |  1e+05
      2 | 2021-01-23 18:30:00.000000+0000 |       Accounts |    Accountant |   pqr     |  2e+05
      4 | 2021-05-23 18:30:00.000000+0000 | Administration | Administrator |   ijk     |  5e+05
      3 | 2021-03-23 18:30:00.000000+0000 |      Marketing |       Manager |   xyz     |  5e+05

(5 rows)


cqlsh:employee_space> update employee_info set emp_name='efg' where emp_id=1;

cqlsh:employee_space> update employee_info set dept_name='Development' where emp_id=1;

cqlsh:employee_space> select * from employee_info;

 emp_id | date_of_joining                 | dept_name      | designation   | emp_name  | salary
--------+---------------------------------+----------------+---------------+-----------+--------
      5 | 2009-05-23 18:30:00.000000+0000 | Administration | Administrator |       lmn |  2e+06
      1 | 2022-01-23 18:30:00.000000+0000 |    Development |       Manager |       efg |  1e+05
      2 | 2021-01-23 18:30:00.000000+0000 |       Accounts |    Accountant |       pqr |  2e+05
      4 | 2021-05-23 18:30:00.000000+0000 | Administration | Administrator |       ijk |  5e+05
      3 | 2021-03-23 18:30:00.000000+0000 |      Marketing |       Manager |       xyz |  5e+05

(5 rows)

cqlsh:employee_space> alter table employee_info add projects set<text>;

cqlsh:employee_space> update employee_info set projects=projects+{'Web development','machine learning'} where
emp_id=2;

cqlsh:employee_space> select * from employee_info;

 emp_id | date_of_joining                 | dept_name      | designation   | emp_name  | projects                                 | salary
--------+---------------------------------+----------------+---------------+-----------+------------------------------------------
+--------
      5 | 2009-05-23 18:30:00.000000+0000 | Administration | Administrator |       lmn |                                     null |
2e+06
      1 | 2022-01-23 18:30:00.000000+0000 |    Development |       Manager |       efg |                                     null |
1e+05
      2 | 2021-01-23 18:30:00.000000+0000 |       Accounts |    Accountant |       pqr | {'Web development', 'machine
learning'} |  2e+05
      4 | 2021-05-23 18:30:00.000000+0000 | Administration | Administrator |       ijk |                                     null |
5e+05
      3 | 2021-03-23 18:30:00.000000+0000 |      Marketing |       Manager |       xyz |                                     null |  5e+05

(5 rows)
cqlsh:employee_space> update employee_info set projects=projects+{'Web development','machine
learning','cybersecurity'} where emp_id=5;

cqlsh:employee_space> select * from employee_info;

emp_id | date_of_joining                 | dept_name      | designation   | emp_name  | projects
| salary
```

```
--------+---------------------------------+----------------+---------------+-----------
+--------------------------------------------------------+--------
      5 | 2009-05-23 18:30:00.000000+0000 | Administration | Administrator |     lmn | {'Web development',
'cybersecurity', 'machine learning'} |  2e+06
      1 | 2022-01-23 18:30:00.000000+0000 |    Development |       Manager |     efg |
null |  1e+05
      2 | 2021-01-23 18:30:00.000000+0000 |       Accounts |    Accountant |     pqr |                  {'Web development',
'machine learning'} |  2e+05
      4 | 2021-05-23 18:30:00.000000+0000 | Administration | Administrator |     ijk |
null |  5e+05
      3 | 2021-03-23 18:30:00.000000+0000 |      Marketing |       Manager |     xyz |
null |  5e+05

(5 rows)

cqlsh:employee_space> INSERT INTO
employee_space.employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
VALUES(6,'mno','Manager','2022-01-24',100000,'Marketing') using ttl 15;

cqlsh:employee_space> select * from employee_info;

 emp_id | date_of_joining                 | dept_name      | designation   | emp_name  | projects
| salary
--------+---------------------------------+----------------+---------------+-----------
+--------------------------------------------------------+--------
      5 | 2009-05-23 18:30:00.000000+0000 | Administration | Administrator |     lmn | {'Web development',
'cybersecurity', 'machine learning'} |  2e+06
      1 | 2022-01-23 18:30:00.000000+0000 |    Development |       Manager |     efg |
null |  1e+05
      2 | 2021-01-23 18:30:00.000000+0000 |       Accounts |    Accountant |     pqr |                  {'Web development',
'machine learning'} |  2e+05
      4 | 2021-05-23 18:30:00.000000+0000 | Administration | Administrator |     ijk |
null |  5e+05
      6 | 2022-01-23 18:30:00.000000+0000 |      Marketing |       Manager |     mno |
null |  1e+05
      3 | 2021-03-23 18:30:00.000000+0000 |      Marketing |       Manager |     xyz |
null |  5e+05

(6 rows)

cqlsh:employee_space> select * from employee_info;

 emp_id | date_of_joining                 | dept_name      | designation   | emp_name  | projects
| salary
--------+---------------------------------+----------------+---------------+-----------
+--------------------------------------------------------+--------
      5 | 2009-05-23 18:30:00.000000+0000 | Administration | Administrator |     lmn | {'Web development',
'cybersecurity', 'machine learning'} |  2e+06
      1 | 2022-01-23 18:30:00.000000+0000 |    Development |       Manager |     efg |
null |  1e+05
      2 | 2021-01-23 18:30:00.000000+0000 |       Accounts |    Accountant | pqr |                  {'Web development',
'machine learning'} |  2e+05
      4 | 2021-05-23 18:30:00.000000+0000 | Administration | Administrator |     ijk |
null |  5e+05
      3 | 2021-03-23 18:30:00.000000+0000 |      Marketing |       Manager |    xyz |
null |  5e+05

(5 rows)
```

Lab 4 - Cassandra

Perform the following DB operations using Cassandra.

1. Create a keyspace by name Library

2. Create a column family by name Library-Info with attributes

Stud_Id Primary Key,

Counter_value of type Counter,

Stud_Name, Book-Name, Book-Id,

Date_of_issue

3. Insert the values into the table in batch

4. Display the details of the table created and increase the value of the counter

5. Write a query to show that a student with id 112 has taken a book “BDA” 2 times.

6. Export the created column to a csv file

7. Import a given csv dataset from local file system into Cassandra column family

```
cqlsh> create keyspace library_space WITH REPLICATION={'class':'SimpleStrategy','replication_factor':2};

cqlsh> use library_space;

cqlsh:library_space> create table library_info(stud_id int,counter_value counter,stud_name text,book_name text,book_id int,date_of_issue timestamp,PRIMARY KEY(stud_id,stud_name,book_name,book_id,date_of_issue));

cqlsh:library_space> update library_info set counter_value=counter_value+1 where stud_id=1 and stud_name='abc' and book_name='book1' and book_id=11 and date_of_issue='2022-01-30';
```

```
cqlsh:library_space> update library_info set counter_value=counter_value+1 where stud_id=2 and stud_name='def'
and book_name='book2' and book_id=12 and date_of_issue='2022-03-30';

cqlsh:library_space> update library_info set counter_value=counter_value+1 where stud_id=3 and stud_name='ghi'
and book_name='book3' and book_id=13 and date_of_issue='2022-05-30';

cqlsh:library_space> update library_info set counter_value=counter_value+1 where stud_id=4 and stud_name='jkl'
and book_name='book4' and book_id=14 and date_of_issue='2022-07-30';

cqlsh:library_space> update library_info set counter_value=counter_value+1 where stud_id=5 and stud_name='mno'
and book_name='book5' and book_id=15 and date_of_issue='2022-09-30';

cqlsh:library_space> select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
       5 |       mno |     book5 |      15 | 2022-09-29 18:30:00.000000+0000 |             1
       1 |       abc |     book1 |      11 | 2022-01-29 18:30:00.000000+0000 |             1
       2 |       def |     book2 |      12 | 2022-03-29 18:30:00.000000+0000 |             1
       4 |       jkl |     book4 |      14 | 2022-07-29 18:30:00.000000+0000 |             1
       3 |       ghi |     book3 |      13 | 2022-05-29 18:30:00.000000+0000 |             1

(5 rows)

cqlsh:library_space> update library_info set counter_value=counter_value+1 where stud_id=5 and stud_name='mno'
and book_name='book5' and book_id=15 and date_of_issue='2022-09-30';

cqlsh:library_space> select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
       5 |       mno |     book5 |      15 | 2022-09-29 18:30:00.000000+0000 |             2
       1 |       abc |     book1 |      11 | 2022-01-29 18:30:00.000000+0000 |             1
       2 |       def |     book2 |      12 | 2022-03-29 18:30:00.000000+0000 |             1
       4 |       jkl |     book4 |      14 | 2022-07-29 18:30:00.000000+0000 |             1
       3 |       ghi |     book3 |      13 | 2022-05-29 18:30:00.000000+0000 |             1

(5 rows)

cqlsh:library_space> copy library_info(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) to
'/home/bmscecse/Desktop/bda.csv';
```

```
Using 11 child processes

Starting copy of library_space.library_info with columns [stud_id, stud_name, book_name, book_id, date_of_issue,
counter_value].
Processed: 5 rows; Rate:      45 rows/s; Avg. rate:      45 rows/s
5 rows exported to 1 files in 0.121 seconds.

cqlsh:library_space> create table library_info_copy(stud_id int,counter_value counter,stud_name text,book_name
text,book_id int,date_of_issue timestamp,PRIMARY KEY(stud_id,stud_name,book_name,book_id,date_of_issue));

cqlsh:library_space> copy library_info_copy(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value)
from '/home/bmscecse/Desktop/new.csv';

Using 11 child processes

Starting copy of library_space.library_info_copy with columns [stud_id, stud_name, book_name, book_id,
date_of_issue, counter_value].
Processed: 5 rows; Rate:       8 rows/s; Avg. rate:      12 rows/s
5 rows imported from 1 files in 0.406 seconds (0 skipped).

cqlsh:library_space> select * from library_info where counter_value=2 allow filtering;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
       5 |       mno |     book5 |      15 | 2022-09-29 18:30:00.000000+0000 |             2
```