BDA LAB REPORT

Name – Shivam Raj

USN – 1BM17CS095

BATCH – B3

Q1.)

Perform the following DB operations using MongoDB.

A. Create a database “Student” with the following attributes Rollno, Age, ContactNo, Email-Id.
B. Insert appropriate values.
C. Write query to update Email-Id of a student with rollno 10.
D. Replace the student name from “ABC” to “FEM” of rollno 11.
E. Export the created table into local file system.
F. Drop the table.
G. Import a given csv dataset from local file system into mongodb collection.

CODE – 1

```
> use student;

> db;

> show dbs;


> db.createCollection("Stud");

> db.Stud.insert({_id:1, name : "Shivam", rollno:1, age:21, ContactNo:999, EmailId: "shiv@r.com"});
```

```
> db.Stud.insert({_id:2, name : "Shiv", rollno:10, age:21, ContactNo:9999, EmailId:
"aBCD@b.com"});

> db.Stud.insert({_id:3, name : "ABC", rollno:11, age:21, ContactNo:999999, EmailId:
"aCD@b.com"});

> db.Stud.insert({_id:4, name : "ADAABC", rollno:12, age:22, ContactNo:99999988, EmailId:
"DNEW@b.com"});


> db.Stud.find();

> db.Stud.update({rollno:10},{$set:{EmailId:"hello@b.com"}});

> db.Stud.find({rollno:10});

> db.Stud.replaceOne({"rollno":11},{"name": "FEM", "rollno":11, "age":21,
"ContactNo":9999, "EmailId": "aBCD@b.com"});

> exit


mongoexport -d student -c Stud -f name,rollno,age,ContactNo,EmailId --type=csv -o
F:\Mongo\Student.csv


> use student;

> db;

> db.Stud.drop();

> db.Stud.find();

> exit


mongoimport -d student -c Stud --type csv -f name,rollno,age,ContactNo,EmailId --file
F:\Mongo\Student.csv


> use student;

> db.Stud.find();
```

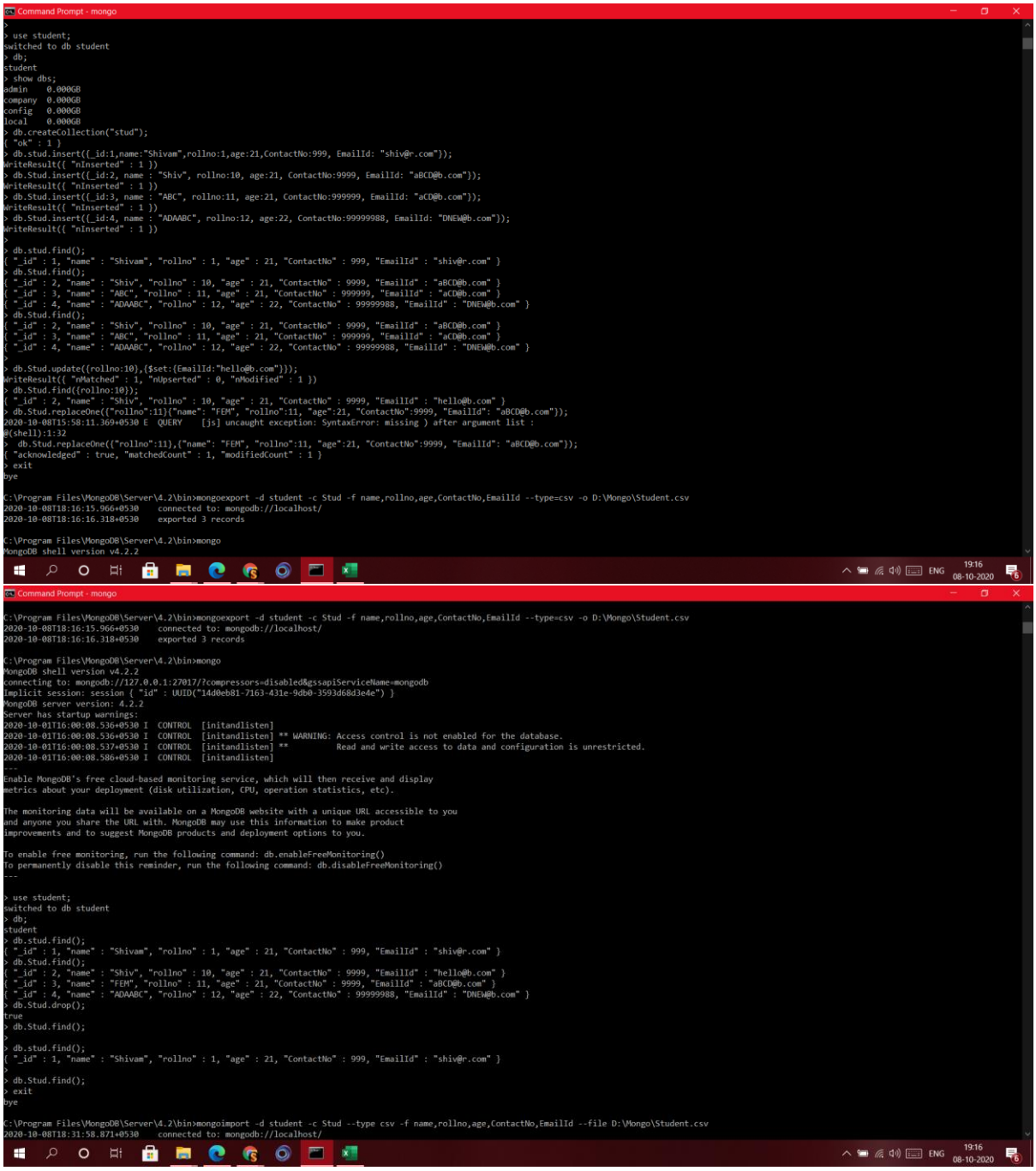
Write-up-1

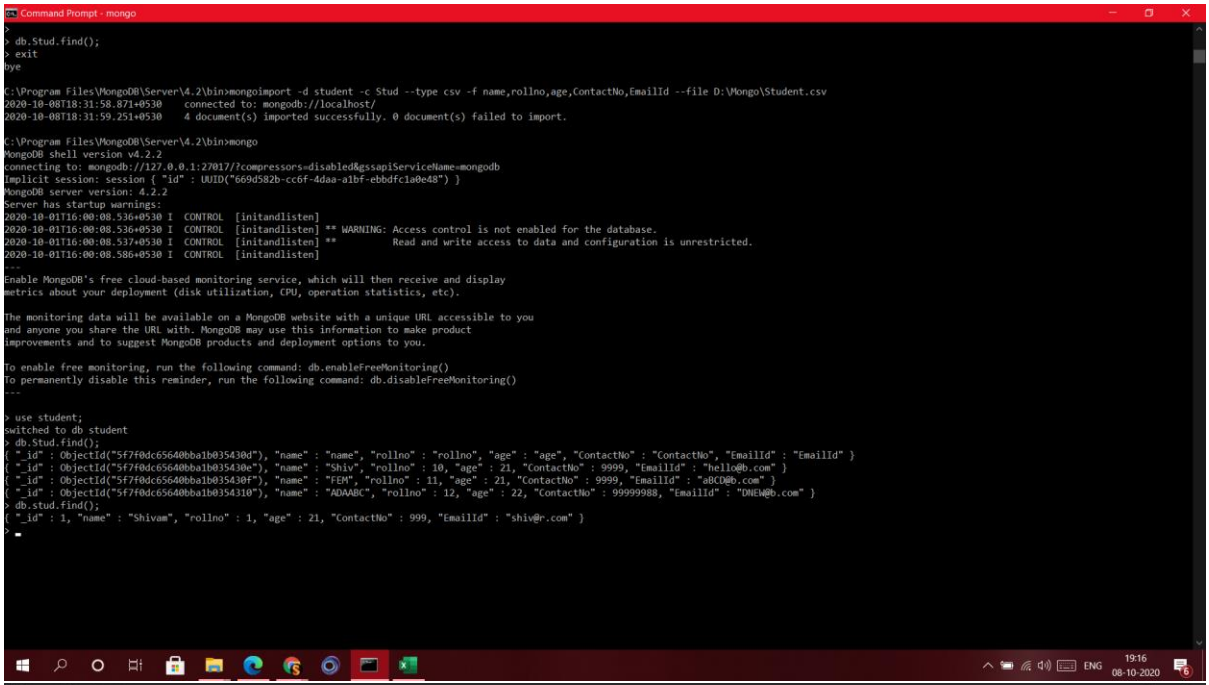
Shivam Raj BDA Lab
1BM17CS095 B3

Lab - 3 (I)

```
use Student;
db;
db.createCollection("stud");
db.stud.insert({_id:1, name:"Shivam",
                rollno:1, age:21, contactno:9999,
                EmailId:"Shivam@r.com"});

db.stud.update({rollno:10}, {$set:{EmailId:
                "hello@s.com"}});

db.stud.replaceOne({"rollno":11}, {"name":"FEM",
                "rollno":11, "age":21, "ContactNo":999,
                "EmailId":"abc@s.com"});

mongoexport -d student -c Stud -f name,rollno, age,
ContactNo, EmailId --type=csv -o D:\mongo\student.csv

db.stud.drop();
db.stud.find();

mongoimport -d student -c stud --type csv -f
name, rollno, age, ContactNo, EmailId --file D:\mongo\
                                        student.csv
db.stud.find();
```

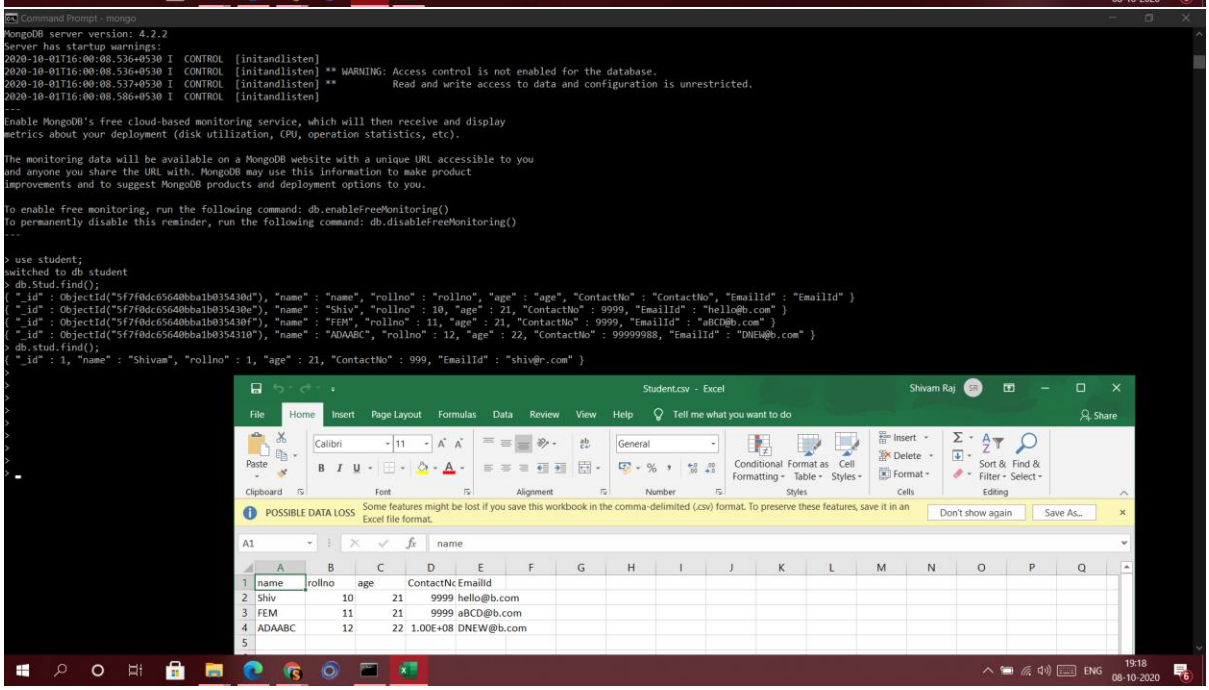
Screenshot – 1




Command Prompt - mongo
> db.Stud.find();
> exit
bye
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport -d student -c Stud --type csv -f name,rollno,age,ContactNo,EmailId --file D:\Mongo\Student.csv
2020-10-08T18:31:58.871+0530 connected to: mongodb://localhost/
2020-10-08T18:31:59.251+0530 4 document(s) imported successfully. 0 document(s) failed to import.
C:\Program Files\MongoDB\Server\4.2\bin>mongo
MongoDB shell version v4.2.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("669d582b-cc6f-4daa-a1bf-ebbdfc1a0e48") }
MongoDB server version: 4.2.2
Server has startup warnings:
2020-10-01T16:00:08.536+0530 I CONTROL [initandlisten]
2020-10-01T16:00:08.536+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-10-01T16:00:08.537+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-10-01T16:00:08.586+0530 I CONTROL [initandlisten]

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
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 student;
switched to db student
> db.Stud.find();
{ "_id" : ObjectId("5f7f0dc65640bba1b035430d"), "name" : "name", "rollno" : "rollno", "age" : "age", "ContactNo" : "ContactNo", "EmailId" : "EmailId" }
{ "_id" : ObjectId("5f7f0dc65640bba1b035430e"), "name" : "Shiv", "rollno" : 10, "age" : 21, "ContactNo" : 9999, "EmailId" : "hello@b.com" }
{ "_id" : ObjectId("5f7f0dc65640bba1b035430f"), "name" : "FEM", "rollno" : 11, "age" : 21, "ContactNo" : 9999, "EmailId" : "aBCD@b.com" }
{ "_id" : ObjectId("5f7f0dc65640bba1b0354310"), "name" : "ADAABC", "rollno" : 12, "age" : 22, "ContactNo" : 99999988, "EmailId" : "DNEW@b.com" }
> db.stud.find();
{ "_id" : 1, "name" : "Shivam", "rollno" : 1, "age" : 21, "ContactNo" : 999, "EmailId" : "shiv@r.com" }
ENG
19:16
08-10-2020


Command Prompt - mongo
Student.csv - Excel
Shivam Raj
POSSIBLE DATA LOSS
Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
name
rollno
age
ContactNc
EmailId
Shiv
10
21
9999
hello@b.com
FEM
11
21
9999
aBCD@b.com
ADAABC
12
22
1.00E+08
DNEW@b.com
ENG
19:18
08-10-2020

Q2.)

Perform the following DB operations using MongoDB.

A. Create a collection by name Customers with the following attributes. Cust_id, Acc_Bal, Acc_Type.
B. Insert at least 5 values into the table.
C. Write a query to display those records whose total account balance is greater than 1200 of account type ‘Z’ for each customer_id.
D. Determine Minimum and Maximum account balance for each customer_id.
E. Export the created collection into local file system
F. Drop the table.
G. Import a given csv dataset from local file system into mongodb collection.

CODE – 2

```
> use mynew;
> db;
> show dbs;
> db.createCollection("Customer");
> db.Customer.insert({cust_id:1,Acc_bal:1500,Acc_type:"Z"});
> db.Customer.insert({cust_id:2,Acc_bal:3000,Acc_type:"A"});
> db.Customer.insert({cust_id:3,Acc_bal:1200,Acc_type:"A"});
> db.Customer.insert({cust_id:2,Acc_bal:1200,Acc_type:"Z"});
> db.Customer.insert({cust_id:3,Acc_bal:1280,Acc_type:"Z"});

> db.Customer.find();
> db.Customer.find({Acc_bal:{$gt:1200}, Acc_type:"Z"});
> db.Customer.aggregate([
... ...    {
... ...        $group: {
```

```
... ...          _id: "$cust_id",
... ...          min_bal: {$min: "$Acc_bal"},
... ...          max_bal: {$max: "$Acc_bal"}
... ...       }
... ...    }
... ... ]);
>
> db.Customer.find();
> exit
```

```
mongoexport -d mynew -c Customer -f cust_id,Acc_bal,Acc_type --type=csv -o
F:\Mongo\Customer.csv
mongo
```

```
> use mynew;
> db.Customer.drop();
> db.Customer.find();
> exit
```

```
mongoimport -d mynew -c Customer --type csv -f cust_id,Acc_bal,Acc_type --file
F:\Mongo\Customer.csv
mongo
```

```
> use mynew;
> db.Customer.find();
```

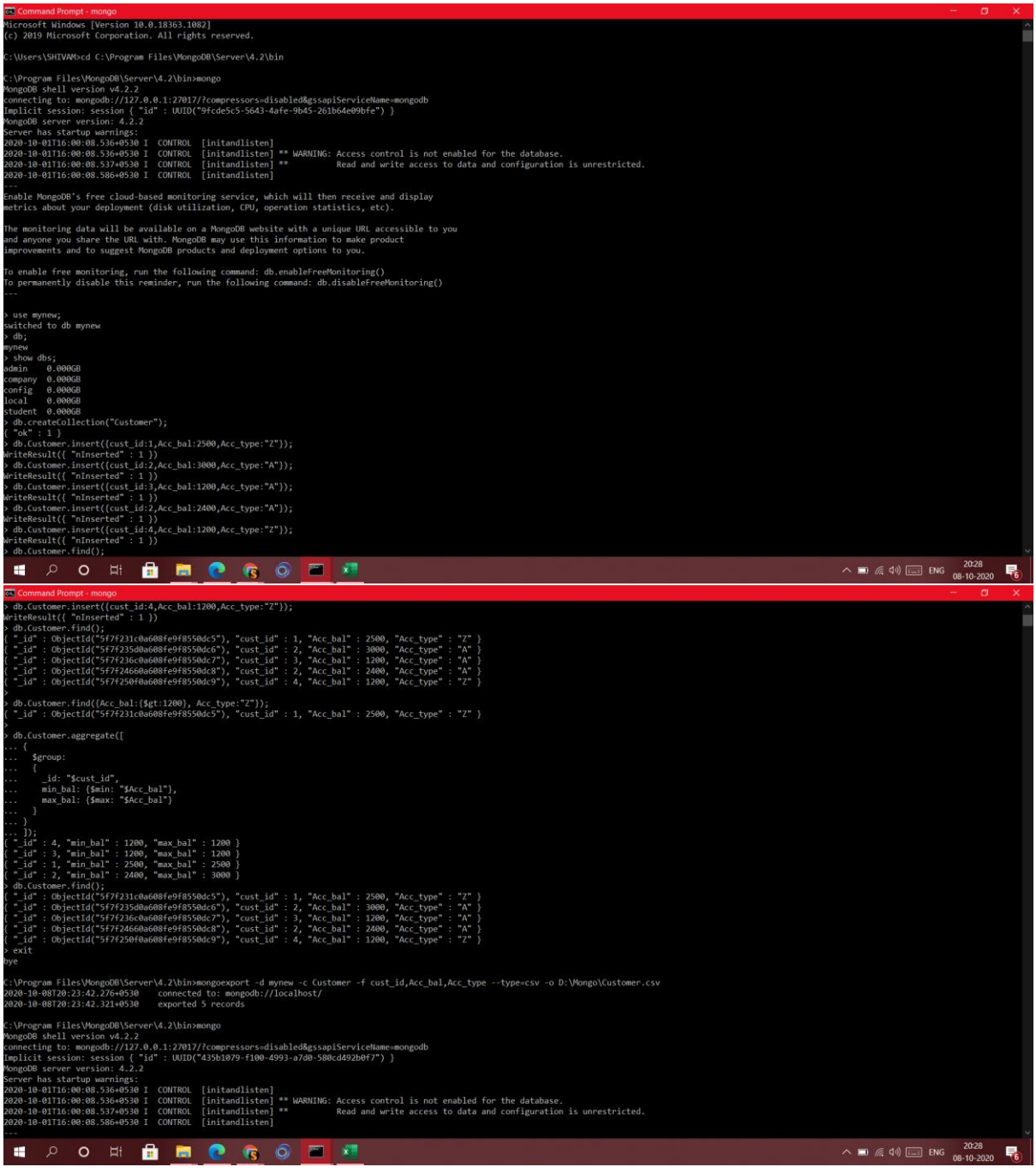
Write-up-2

Shivam Raj BDA Lab
1BM17CS095 B3

Lab-3 (II)

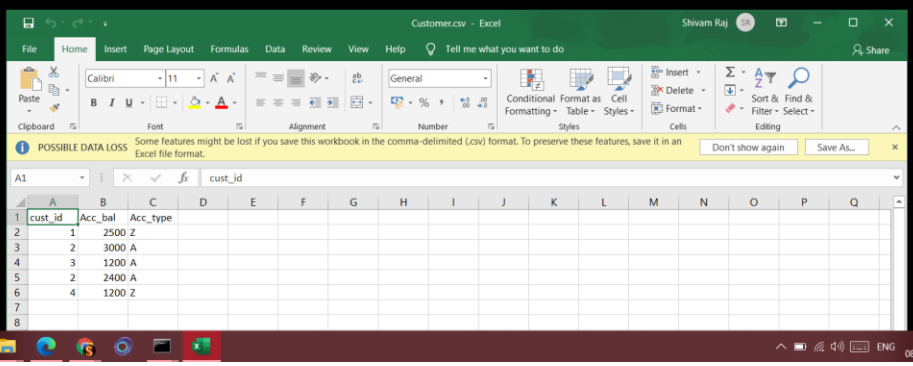
```
use my new;
db;
db.createCollection("Customer");
db.Customer.insert({cust_id:1, Acc_bal:1500,
                    Acc_type:"Z"});
db.Customer.find();
db.Customer.find({Acc_bal:{$gt:1200},
                  Acc_type:"Z"});

db.Customer.aggregate([
    {
        $group:{
            _id:"$cust_id",
            min_bal:{$min:"$Acc_bal"},
            max_bal:{$max:"$Acc_bal"}
        }
    }
]);
mongoimport -d mynew -c Customer -f cust_id,
Acc_bal,Acc_type --type csv -0 D:\mongo\customer.csv

db.customer.drop()
db.Customer.find()

mongoimport -d mynew -c Customer --type csv -f cust_id,
Acc_bal,Acc_type --file D:\mongo\Customer.csv

db.Customer.find();
```

Screenshot – 2



```
Command Prompt - mongo
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
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 mynew;
switched to db mynew
> db.Customer.drop();
true
> db.Customer.find();
>
> exit
bye

C:\Program Files\MongoDB\Server\4.2\bin>mongoimport -d mynew -c Customer --type csv -f cust_id,Acc_bal,Acc_type --file D:\Mongo\Customer.csv
2020-10-08T20:26:31.329+0530    connected to: mongodb://localhost/
2020-10-08T20:26:31.516+0530    6 document(s) imported successfully. 0 document(s) failed to import.

C:\Program Files\MongoDB\Server\4.2\bin>mongo
MongoDB shell version v4.2.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("87c4cb33-338c-4949-aace-b5012e1977c6") }
MongoDB server version: 4.2.2
Server has startup warnings:
2020-10-01T16:00:08.536+0530 I  CONTROL  [initandlisten]
2020-10-01T16:00:08.536+0530 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-10-01T16:00:08.537+0530 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-10-01T16:00:08.586+0530 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
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 mynew;
switched to db mynew
> db.Customer.find();
```

```
Command Prompt - mongo
2020-10-01T16:00:08.537+0530 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-10-01T16:00:08.586+0530 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
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 mynew;
switched to db mynew
> db.Customer.find();
{ "_id" : ObjectId("5f7f289fa8939e106c990cc6"), "cust_id" : 2, "Acc_bal" : 3000, "Acc_type" : "A" }
{ "_id" : ObjectId("5f7f289fa8939e106c990cc7"), "cust_id" : 1, "Acc_bal" : 2500, "Acc_type" : "Z" }
{ "_id" : ObjectId("5f7f289fa8939e106c990cc8"), "cust_id" : 2, "Acc_bal" : 2400, "Acc_type" : "A" }
{ "_id" : ObjectId("5f7f289fa8939e106c990cc9"), "cust_id" : 4, "Acc_bal" : 1200, "Acc_type" : "Z" }
{ "_id" : ObjectId("5f7f289fa8939e106c990cca"), "cust_id" : "cust_id", "Acc_bal" : "Acc_bal", "Acc_type" : "Acc_type" }
{ "_id" : ObjectId("5f7f289fa8939e106c990ccb"), "cust_id" : 3, "Acc_bal" : 1200, "Acc_type" : "A" }
>
```



Q3.)

Perform the following DB operations using Cassandra.

A. Create a keyspace by name Employee
B. Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name.
C. Insert the values into the table in batch
D. Update Employee name and Department of Emp-Id 121.
E. Sort the details of Employee records based on salary.
F. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
G. Update the altered table to add project names.
H. Create a TTL of 15 seconds to display the values of Employees.

CODE – 3

```
cqlsh> CREATE KEYSPACE Employees WITH replication =
{'class':'SimpleStrategy','replication_factor':3};

cqlsh> use employees;

cqlsh:employees> CREATE COLUMNFAMILY employee_info(Emp_id INT PRIMARY KEY,
Emp_name VARCHAR, Designation VARCHAR, Date_of_joining VARCHAR, Salary FLOAT,
Dept_name VARCHAR);


cqlsh:employees> BEGIN BATCH INSERT INTO employee_info (emp_id , emp_name ,
designation , date_of_joining , salary , dept_name ) values
(119,'Shivam','CEO','01/06/2020',10000, 'IT');

        ... INSERT INTO employee_info (emp_id , emp_name , designation , date_of_joining ,
salary , dept_name ) values (120,'Shivam Raj', 'CFO','01/01/2020',5000,'Finance');

        ... INSERT INTO employee_info (emp_id , emp_name , designation , date_of_joining ,
salary , dept_name ) values (121,'Shiv', 'HR head','01/04/2020',4000,'SALES');

        ... INSERT INTO employee_info (emp_id , emp_name , designation , date_of_joining ,
salary , dept_name ) values (122,'Raj','CTO','01/06/2020',10000, 'SALES'); APPLY BATCH;


cqlsh:employees> SELECT * FROM employee_info;
```

```
cqlsh:employees> UPDATE employee_info SET emp_name = 'Shiva',dept_name = 'IT' WHERE emp_id = 121;

cqlsh:employees> SELECT * FROM employee_info;

cqlsh:employees> ALTER TABLE employee_info ADD PROJECT SET<VARCHAR>;

cqlsh:employees> UPDATE employee_info SET project=project+{'Covid-19','BDA Analysis'} WHERE emp_id=122;

cqlsh:employees> INSERT INTO employee_info (emp_id , emp_name , designation , date_of_joining , salary , dept_name ) values (123,'Random Guy','Manager','01/06/2020',10000, 'IT') USING TTL 15;

cqlsh:employees> SELECT TTL(designation) FROM employee_info where emp_id=123;

cqlsh:employees> SELECT * FROM employee_info;

cqlsh:employees> SELECT TTL(designation) FROM employee_info where emp_id=123;
```

Write-up-3

Shivam Raj BDA Lab
1BM17CS095 B3
Lab - 6 (I)

> cqlsh

```
CREATE KEYSPACE Employees WITH replication = {'class':'SimpleStrategy','replication_factor':3};
use employees;
CREATE COLUMNFAMILY employee_info (Emp_id INT PRIMARY KEY, Emp_name VARCHAR, Designation VARCHAR, Date_of_joining VARCHAR, Salary FLOAT, Dept_name VARCHAR);

BEGIN BATCH INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) values (11,'Shivam','CEO','01/06/2020',10000,'IT');
...
... APPLY BATCH;
Select * from employee_info;
UPDATE employee_info SET emp_name = 'Shiva', dept_name = 'HR' WHERE emp_id = 121;
ALTER TABLE employee_info ADD PROJECT SET<VARCHAR>;
UPDATE employee_info SET project = project + {'COVID-19','BDA Analysis'} WHERE emp_id = 122;
Select * from employee_info
BEGIN BATCH INSERT INTO employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) values (123,'Random Guy','Manager','01/06/2020',10000,'IT') USING TTL 15; ... APPLY BATCH;
SELECT TTL(designation) FROM employee_info where emp_id=123;
```

Screenshot – 3

Q4.)

Perform the following DB operations using Cassandra.

A. Create a keyspace by name Library.
B. 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.
C. Insert the values into the table in batch.
D. Display the details of the table created and increase the value of the counter.
E. Write a query to show that a student with id 112 has taken a book "BDA" 2 times.
F. Export the created column to a csv file.
G. Import a given csv dataset from local file system into Cassandra column family

CODE – 4

```
cqlsh> CREATE KEYSPACE Library WITH replication =
{'class':'SimpleStrategy','replication_factor':3};

cqlsh> use Library;

cqlsh:library> CREATE COLUMNFAMILY library_info(stud_id int, counter_value counter,
stud_name VARCHAR, book_name VARCHAR, book_id INT, DOI VARCHAR, PRIMARY
KEY(stud_id,stud_name,book_name,book_id,doi));

cqlsh:library> UPDATE library_info set counter_value=counter_value+1 where stud_id=112
and stud_name='Shivam' and book_name='Vedas' and book_id=1 and doi='01/01/2020';

cqlsh:library> UPDATE library_info set counter_value=counter_value+1 where stud_id=112
and stud_name='Shivam' and book_name='ved' and book_id=1 and doi='01/01/2020';

cqlsh:library> UPDATE library_info set counter_value=counter_value+1 where stud_id=113
and stud_name='Raj' and book_name='Geeta' and book_id=2 and doi='01/05/2020';

cqlsh:library> SELECT * FROM library_info;

cqlsh:library> SELECT * FROM library_info WHERE stud_id=112;
```

```
cqlsh:library> SELECT book_name,counter_value FROM library_info WHERE stud_id=112;

cqlsh:library> COPY library_info (stud_id , counter_value , stud_name , book_name ,
book_id , doi ) TO 'library.csv' WITH HEADER=TRUE;

cqlsh:library> truncate library_info ;

cqlsh:library> SELECT * FROM library_info;

cqlsh:library> COPY library_info (stud_id , counter_value , stud_name , book_name ,
book_id , doi ) FROM 'library.csv' WITH HEADER=TRUE;

cqlsh:library> SELECT * FROM library_info;
```

Write-up-4

Shivam Raj
1BM17CS095

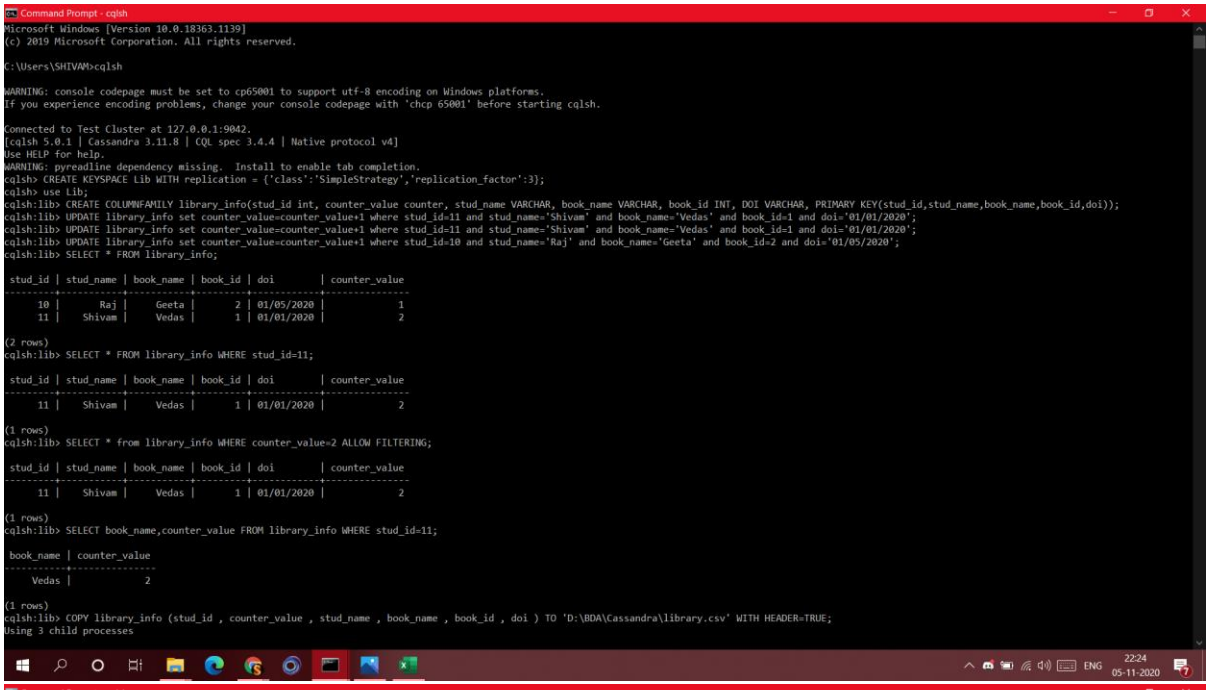
BDA Lab
B3

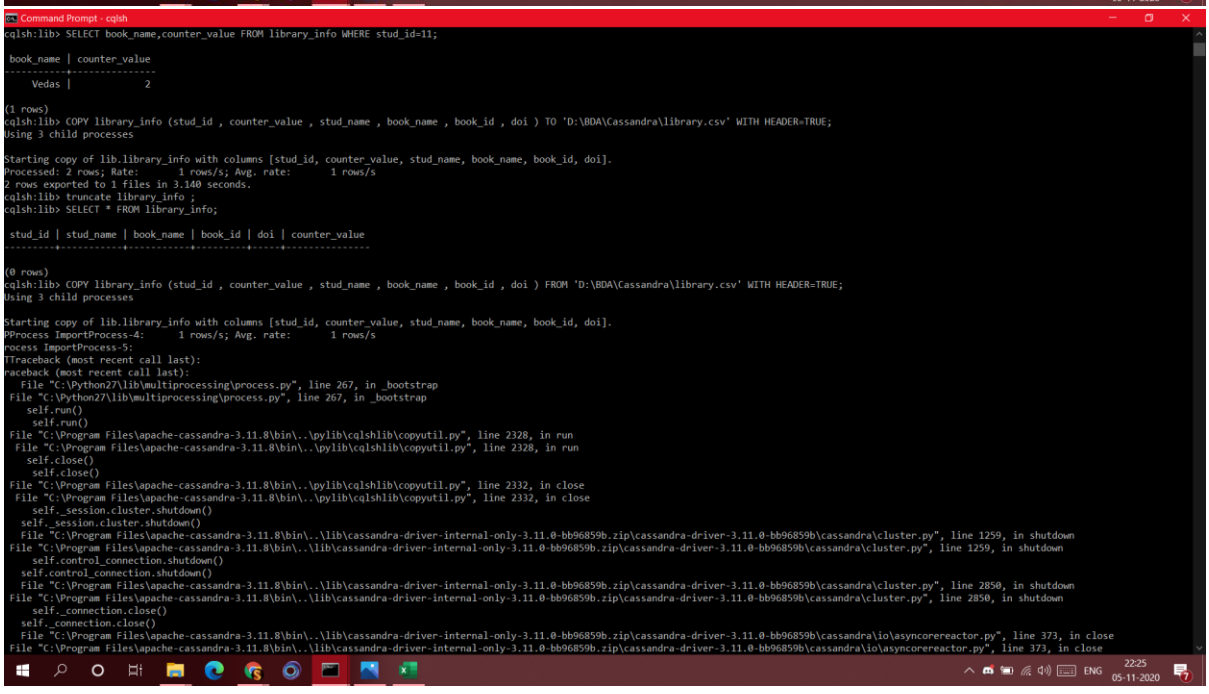
Lab - 6 (II)

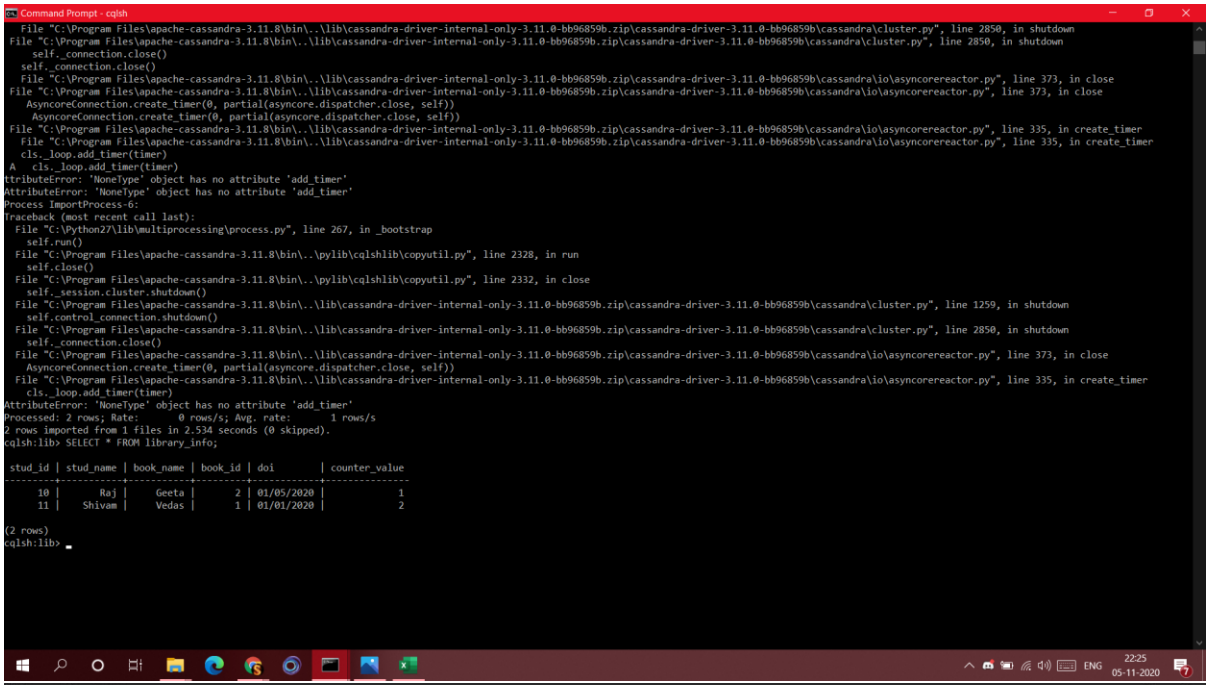
> cqlsh

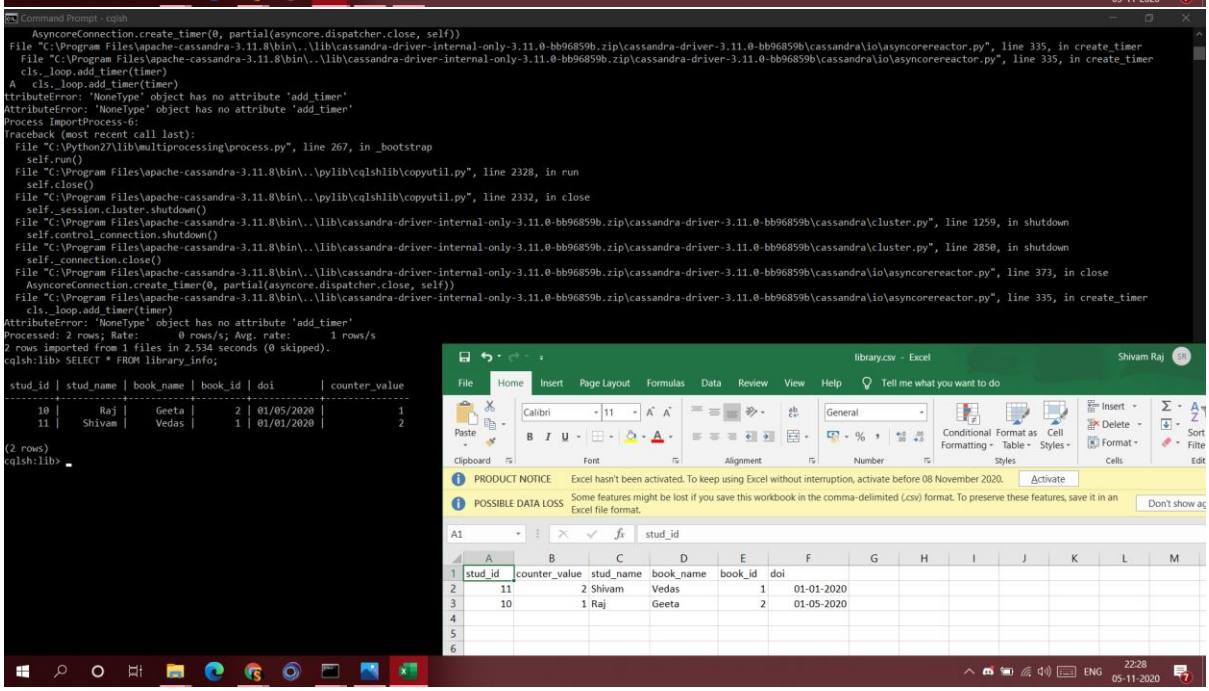
```
CREATE KEYSPACE Lib WITH replication =
{'class':'SimpleStrategy', 'replication_factor':3};
use Lib;
CREATE COLUMNFAMILY library_info (stud_id int,
counter_value counter, stud_name VARCHAR, book_name
VARCHAR, book_id INT, DOI VARCHAR, PRIMARY KEY
(stud_id, stud_name, book_name, book_id, doi));

UPDATE library_info set counter_value =
counter_value+1 where stud_id=11 and
stud_name='Shivam' and book_name=
'Vedas' and book_id=1 and doi='01/01/2020';

Select * from library_info where stud_id=11;

COPY library_info (stud_id, counter_value, stud_name,
book_id, doi) TO 'D:\BDA\Cassandra\library.csv'
WITH HEADER=TRUE;

COPY library_info(stud_id, counter_value, stud_name,
book_name, book_id, doi) FROM 'D:\BDA\
Cassandra\library.csv' WITH HEADER=TRUE;

Select * from library_info;
```

Screenshot – 4






Command Prompt - cqlsh
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 2850, in shutdown
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 2850, in shutdown
self._connection.close()
self._connection.close()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
cls._loop.add_timer(timer)
A cls._loop.add_timer(timer)
ttributeError: 'NoneType' object has no attribute 'add_timer'
AttributeError: 'NoneType' object has no attribute 'add_timer'
Process ImportProcess-6:
raceback (most recent call last):
File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
self.run()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
self.close()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2332, in close
self._session.cluster.shutdown()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 1259, in shutdown
self.control_connection.shutdown()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 2850, in shutdown
self._connection.close()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
cls._loop.add_timer(timer)
AttributeError: 'NoneType' object has no attribute 'add_timer'
Processed: 2 rows; Rate: 0 rows/s; Avg. rate: 1 rows/s
2 rows imported from 1 files in 2.534 seconds (0 skipped).
cqlsh:lib> SELECT * FROM library_info;
stud_id | stud_name | book_name | book_id | doi | counter_value
10 | Raj | Geeta | 2 | 01/05/2020 | 1
11 | Shivam | Vedas | 1 | 01/01/2020 | 2
(2 rows)
cqlsh:lib>
ENG
22:25
05-11-2020


Command Prompt - cqlsh
AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
cls._loop.add_timer(timer)
A cls._loop.add_timer(timer)
ttributeError: 'NoneType' object has no attribute 'add_timer'
AttributeError: 'NoneType' object has no attribute 'add_timer'
Process ImportProcess-6:
raceback (most recent call last):
File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
self.run()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2328, in run
self.close()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\pylib\cqlshlib\copyutil.py", line 2332, in close
self._session.cluster.shutdown()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 1259, in shutdown
self.control_connection.shutdown()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 2850, in shutdown
self._connection.close()
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
File "C:\Program Files\apache-cassandra-3.11.8\bin\..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
cls._loop.add_timer(timer)
AttributeError: 'NoneType' object has no attribute 'add_timer'
Processed: 2 rows; Rate: 0 rows/s; Avg. rate: 1 rows/s
2 rows imported from 1 files in 2.534 seconds (0 skipped).
cqlsh:lib> SELECT * FROM library_info;
stud_id | stud_name | book_name | book_id | doi | counter_value
10 | Raj | Geeta | 2 | 01/05/2020 | 1
11 | Shivam | Vedas | 1 | 01/01/2020 | 2
(2 rows)
cqlsh:lib>
library.csv - Excel
Shivam Raj
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do
PRODUCT NOTICE Excel hasn't been activated. To keep using Excel without interruption, activate before 08 November 2020. Activate
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
A1 stud_id
stud_id counter_value stud_name book_name book_id doi
11 2 Shivam Vedas 1 01-01-2020
10 1 Raj Geeta 2 01-05-2020
ENG
22:28
05-11-2020

Q5.)

Develop a MapReduce program to count the number of occurrences of words in a given file.

CODE – 5

cd /usr/local/hadoop/sbin

start-all.sh

jps

mkdir hadfiles

cd hadfiles

touch wordcountfile.txt

nano wordcountfile.txt

hadoop fs -mkdir/programs

hadoop fs -copyFromLocal/home/hadoop/hadfiles/wordcountfiles.txt/programs/prog1.txt

hadoop jar/home/hdoop/hadfiles/wordcount.jar
WordCount/programs/prog1.txt/programs/output

hadoop fs -ls/programs/output

hadoop fs -cat/programs/output2/part-r-00000

stop-all.sh

Screenshot – 5

1. Starting Hadoop Cluster

```
[/]$ cd /usr/local/hadoop/sbin
[/usr/local/hadoop/sbin]$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hdoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [DESKTOP-72UO25C]
2020-12-08 23:10:37,345 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java clas
ses where applicable
Starting resourcemanager
Starting nodemanagers
[/usr/local/hadoop/sbin]$ jps
14546 NameNode
15315 NodeManager
14948 SecondaryNameNode
15159 ResourceManager
15671 Jps
14698 DataNode
```

2. Creating file for counting words of

```
[~]$ mkdir hadfiles
[~]$ cd hadfiles
[~/hadfiles]$ touch wordcountfile.txt
[~/hadfiles]$ nano wordcountfile.txt
```

3. Moving file to Hadoop File System

```
[/usr/local/hadoop]$ hadoop fs -mkdir /programs
```

```
[/usr/local/hadoop]$ hadoop fs -copyFromLocal /home/hdoop/hadfiles/wordcountfile.txt /programs/prog1.txt
```

4. Running wordcount.jar, which consists of wordcount$Map, wordcount$Reduce, and wordcount classes

```
[~/hadfiles]$ hadoop jar /home/hdoop/hadfiles/wordcount.jar WordCount /programs/prog1.txt /programs/output/
```

5. Output received

```
~/hadfiles
  hadoop fs -ls /programs/output
Found 2 items
-rw-r--r--   1 hdoop supergroup          0 2020-12-08 23:36 /programs/output/_SUCCESS
-rw-r--r--   1 hdoop supergroup         69 2020-12-08 23:36 /programs/output/part-r-00000

~/hadfiles
  hadoop fs -cat /programs/output/part-r-00000
are     1
brother 1
family  1
hi      1
how     5
is      4
job     1
sister  1
you     1
your    4
```

6. Stopping Hadoop

```
~/hadfiles
  stop-all.sh
```

Q6.)

For the given file, create a Map Reduce program to

a) Find the average temperature for each year from NCDC data set.

CODE – 6

```
cd /usr/local/hadoop/sbin
start-all.sh
jps

hadoop fs -copyFromLocal/home/hadfiles/1901/programs/avgtemp.txt
hadoop fs-ls/programs
hadoop jar/home/hdoop/hadfiles/average.jar AverageDriver/programs/avgtemp.txt/progr
hadoop fs -ls/programs
hadoop fs -cat/programs/output2/part-r-00000
stop-all.sh
```

Screenshot – 6

1. Starting Hadoop Cluster

```
hdoop@DESKTOP-72UO25C /$ cd /usr/local/hadoop/sbin
hdoop@DESKTOP-72UO25C /usr/local/hadoop/sbin$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hdoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [DESKTOP-72UO25C]
Starting resourcemanager
Starting nodemanagers
hdoop@DESKTOP-72UO25C /usr/local/hadoop/sbin$ jps
4128 SecondaryNameNode
3906 DataNode
4499 NodeManager
4342 ResourceManager
4859 Jps
3725 NameNode
```

2. Copying Binary file to Hadoop File System as a txt file

```
hdoop@DESKTOP-72UO25C ~/hadfiles$ hadoop fs -copyFromLocal /home/hdoop/hadfiles/1901 /programs/avgtemp.txt
hdoop@DESKTOP-72UO25C ~/hadfiles$ hadoop fs -ls /programs
Found 3 items
-rw-r--r--   1 hdoop supergroup     888190 2020-12-14 15:04 /programs/avgtemp.txt
```

3. Running Average.jar

```
hdoop@DESKTOP-72UO25C ~/hadfiles$ hadoop jar /home/hdoop/hadfiles/average.jar AverageDriver /programs/avgtemp.txt /progr
ams/output2
```

4. Checking location of output file

```
hdoop@DESKTOP-72UO25C ~/hadfiles$ hadoop fs -ls /programs
Found 4 items
-rw-r--r--   1 hdoop supergroup     888190 2020-12-14 15:04 /programs/avgtemp.txt
drwxr-xr-x   - hdoop supergroup          0 2020-12-08 23:36 /programs/output
drwxr-xr-x   - hdoop supergroup          0 2020-12-14 15:07 /programs/output2
-rw-r--r--   1 hdoop supergroup         92 2020-12-08 23:20 /programs/prog1.txt
```

5. Result

```
hdoop@DESKTOP-72UO25C ~/hadfiles$ hadoop fs -cat /programs/output2/part-r-00000
1901    46
```

6. Stopping Hadoop

```
hdoop@DESKTOP-72UO25C ~/hadfiles$ stop-all.sh
```

Q7.)

Write Queries in Hive to do the following

A. Create an external table named with the following attributes -> Empl_ID ->Emp_Name -> Designation -> Salary
B. Load data into table from a given file.
C. Create a view to Generate a query to retrieve the employee details who earn a salary of more than Rs 30000.
D. Alter the table to add a column Dept_Id and Generate a query to retrieve the employee details in order by using Dept_Id.
E. Generate a query to retrieve the number of employees in each department whose salary is greater than 30000
F. Create another table Department with attributes -> Dept_Id ->Dept_name ->Emp_Id 7.Display the cumulative details of each employee along with department details

CODE – 7

A.

```
CREATE DATABASE IF NOT EXISTS lab9 COMMENT 'employee program' WITH DBPROPERTIES
('creator'=ROUNAK);
SHOW DATABASES;
DESCRIBE DATABASE lab9;
USE lab9;
CREATE EXTERNAL TABLE IF NOT EXISTS Employee(EmpID INT,EmpName
STRING,Designation STRING,Salary FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED
BY '\t';
```

B.

```
LOAD DATA LOCAL INPATH '/home/rounak/Desktop/employeeInput.txt' OVERWRITE INTO
```

```
TABLE Employee;
SELECT * FROM Employee;
```

C.

```
CREATE VIEW emp_30000 AS SELECT * FROM Employee WHERE Salary>30000;
SELECT * FROM emp_30000;
```

D.

```
ALTER TABLE Employee ADD COLUMNS(DeptID INT);
LOAD DATA LOCAL INPATH '/home/rounak/Desktop/employeeInputAltered.txt'
OVERWRITE INTO TABLE Employee;
SELECT * FROM Employee;
SELECT * FROM Employee ORDER BY DeptID;
```

E.

```
SELECT DeptID,count(*) FROM Employee WHERE Salary>=30000 GROUP BY DeptID;
```

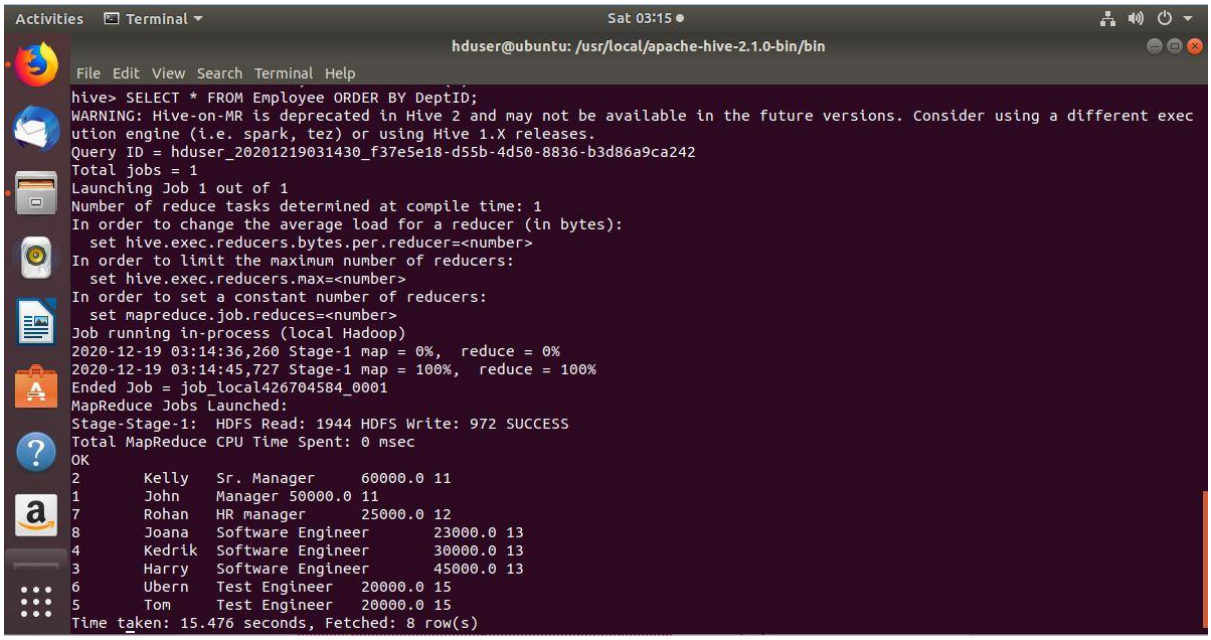
F.

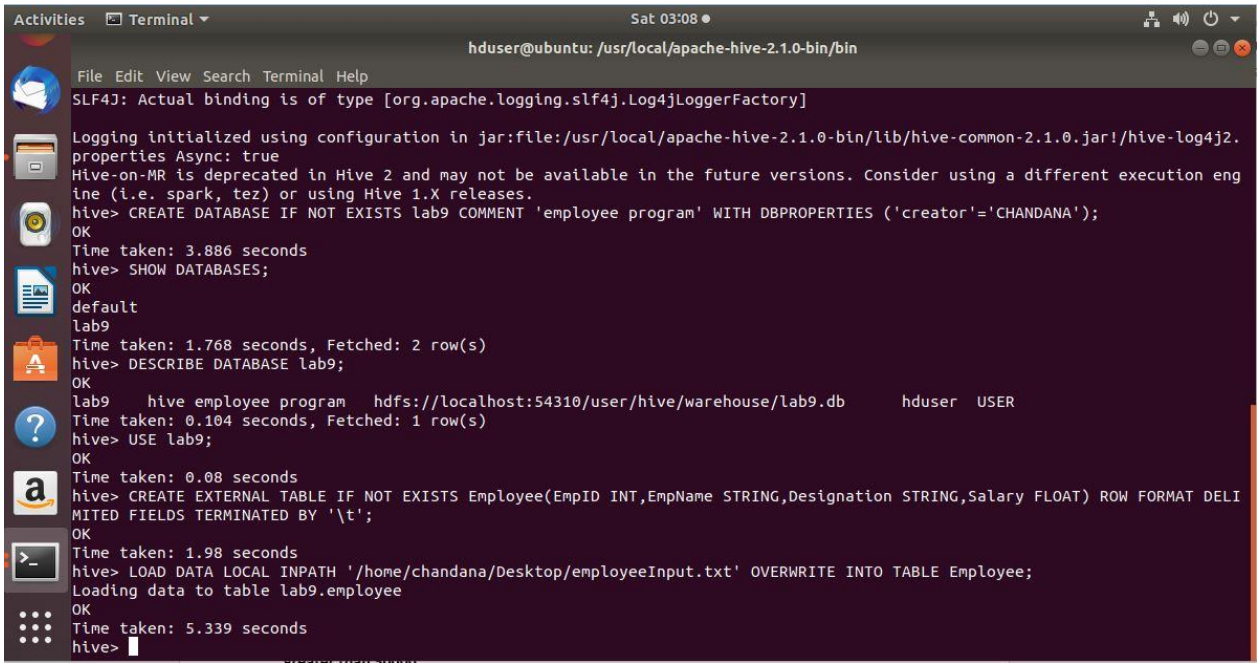
```
CREATE EXTERNAL TABLE IF NOT EXISTS Department(DeptId INT,DeptName STRING) ROW
FORMAT DELIMITED FIELDS TERMINATED BY '\t';
LOAD DATA LOCAL INPATH '/home/rounak/Desktop/DepartmentInput.txt' OVERWRITE
INTO TABLE Department;
SELECT * FROM Department;
```

G.

```
SELECT a.EmpID,a.EmpName,a.Designation,a.Salary,b.DeptName FROM Employee a
JOIN Department b ON a.DeptID=b.DeptId;
```

Screenshot – 7




Activities
Terminal
Sat 03:08
hduser@ubuntu: /usr/local/apache-hive-2.1.0-bin/bin
File Edit View Search Terminal Help
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Logging initialized using configuration in jar:file:/usr/local/apache-hive-2.1.0-bin/lib/hive-common-2.1.0.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> CREATE DATABASE IF NOT EXISTS lab9 COMMENT 'employee program' WITH DBPROPERTIES ('creator'='CHANDANA');
OK
Time taken: 3.886 seconds
hive> SHOW DATABASES;
OK
default
lab9
Time taken: 1.768 seconds, Fetched: 2 row(s)
hive> DESCRIBE DATABASE lab9;
OK
lab9 hive employee program hdfs://localhost:54310/user/hive/warehouse/lab9.db hduser USER
Time taken: 0.104 seconds, Fetched: 1 row(s)
hive> USE lab9;
OK
Time taken: 0.08 seconds
hive> CREATE EXTERNAL TABLE IF NOT EXISTS Employee(EmpID INT,EmpName STRING,Designation STRING,Salary FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
OK
Time taken: 1.98 seconds
hive> LOAD DATA LOCAL INPATH '/home/chandana/Desktop/employeeInput.txt' OVERWRITE INTO TABLE Employee;
Loading data to table lab9.employee
OK
Time taken: 5.339 seconds
hive>