

## **BDA LAB REPORT**

**Name: Suraj Perdoor**

**USN: 1BM18CS114**

**LAB 1**

**Date: 25-03-2021**

**A.Question**

Practice the queries

**B.Commands**

Connected to Test Cluster at 127.0.0.1:9042.

[cqlsh 5.0.1 | Cassandra 4.0 | CQL spec 3.4.5 | Native protocol v4]

Use HELP for help.

cqlsh> CREATE KEYSPACE Students WITH REPLICATION =

{'class':'SimpleStrategy','replication\_factor':1};

cqlsh> DESCRIBE KEYSPACES;

```
students    system_auth      system_schema  system_views  
system     system_distributed system_traces  system_virtual_schema
```

cqlsh> USE Students;

cqlsh:students> CREATE TABLE Students\_Info (Roll\_No int PRIMARY KEY, StudName text, DateOfJoining timestamp, last\_exam\_Percent double);

cqlsh:students> DESCRIBE TABLES;

students\_info

cqlsh:students> DESCRIBE TABLE Students\_Info;

CREATE TABLE students.students\_info (

roll\_no int PRIMARY KEY,

dateofjoining timestamp,

last\_exam\_percent double,

studname text

) WITH additional\_write\_policy = '99p'

AND bloom\_filter\_fp\_chance = 0.01

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

AND cdc = false

AND comment = ''

AND compaction = {'class':

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

AND compression = {'chunk\_length\_in\_kb': '16', 'class':

'org.apache.cassandra.io.compress.LZ4Compressor'}

```

AND crc_check_chance = 1.0
AND default_time_to_live = 0
AND extensions = {}
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair = 'BLOCKING'
AND speculative_retry = '99p';
cqlsh:students> BEGIN BATCH
    ... INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) VALUES (1,'Asha','2012-03-12',79.9)
    ... APPLY BATCH
    ...
cqlsh:students> INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) VALUES (2,'Krian','2012-0
3-12',89.9)
    ...
cqlsh:students> INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) VALUES (3,'Tarun','2012-0
3-12',78.9);
cqlsh:students> INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) VALUES (4,'Samarth','2012
-03-12',90.9);
cqlsh:students> INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) VALUES (5,'Smitha','2012-
03-12',67.9);
cqlsh:students> INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) VALUES (6,'Rohan','2012-0
3-12',56.9);
cqlsh:students> SELECT * FROM Student_Info;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table student_info
does not exist"
cqlsh:students> SELECT * FROM Students_Info;

```

roll_no	dateofjoining	last_exam_percent	studname
5	2012-03-11 18:30:00.000000+0000	67.9	Smitha
1	2012-03-11 18:30:00.000000+0000	79.9	Asha
2	2012-03-11 18:30:00.000000+0000	89.9	Krian
4	2012-03-11 18:30:00.000000+0000	90.9	Samarth
6	2012-03-11 18:30:00.000000+0000	56.9	Rohan
3	2012-03-11 18:30:00.000000+0000	78.9	Tarun

(6 rows)

```
cqlsh:students> SELECT * FROM Students_Info WHERE Roll_No IN (1,2,3);
```

roll_no	dateofjoining	last_exam_percent	studname
1	2012-03-11 18:30:00.000000+0000	79.9	Asha
2	2012-03-11 18:30:00.000000+0000	89.9	Krian
3	2012-03-11 18:30:00.000000+0000	78.9	Tarun

(3 rows)

```
cqlsh:students> SELECT * FROM Students_Info WHERE studname='Asha';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this
query as it might involve data filtering and thus may have unpredictable performance. If you
want to execute this query despite the performance unpredictability, use ALLOW
FILTERING"
```

```
cqlsh:students> CREATE INDEX ON Students_Info(StudName);
cqlsh:students> SELECT * FROM Students_Info WHERE Studname='Asha';
```

roll_no	dateofjoining	last_exam_percent	studname
1	2012-03-11 18:30:00.000000+0000	79.9	Asha

(1 rows)

```
cqlsh:students> select Roll_No, StudName from students_info LIMIT 2;
```

roll_no	studname
5	Smitha
1	Asha

(2 rows)

```
cqlsh:students> select Roll_No as "USN" from Students_Info;
```

```
USN
-----
5
1
2
4
6
3
```

(6 rows)

```
cqlsh:students> UPDATE students_info SET studname='David Sheen' WHERE Roll_No=2;
cqlsh:students> UPDATE students_info SET Roll_No=7 WHERE Roll_No=3;
InvalidRequest: Error from server: code=2200 [Invalid query] message="PRIMARY KEY
part roll_no found in SET part"
```

```
cqlsh:students> DELETE last_exam_percent FROM students_info where roll_no=2;
cqlsh:students> select * from students_info;
```

roll_no	dateofjoining	last_exam_percent	studname
5	2012-03-11 18:30:00.000000+0000	67.9	Smitha
1	2012-03-11 18:30:00.000000+0000	79.9	Asha
2	2012-03-11 18:30:00.000000+0000	null	David Sheen
4	2012-03-11 18:30:00.000000+0000	90.9	Samarth
6	2012-03-11 18:30:00.000000+0000	56.9	Rohan
3	2012-03-11 18:30:00.000000+0000	78.9	Tarun

(6 rows)

```
cqlsh:students> delete from students_info where roll_no=2;
cqlsh:students> select * from students_info;
```

roll_no	dateofjoining	last_exam_percent	studname
5	2012-03-11 18:30:00.000000+0000	67.9	Smitha
1	2012-03-11 18:30:00.000000+0000	79.9	Asha
4	2012-03-11 18:30:00.000000+0000	90.9	Samarth
6	2012-03-11 18:30:00.000000+0000	56.9	Rohan
3	2012-03-11 18:30:00.000000+0000	78.9	Tarun

(5 rows)

```
cqlsh:students> alter table students_info add hobbies set<text>;
cqlsh:students> alter table students_info add language list<text>;
cqlsh:students> update students_info set hobbies=hobbies+{'Chess,Table Tennis'} where
RollNo=1;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Undefined column
name rollno in table students.students_info"
cqlsh:students> update students_info set hobbies=hobbies+{'Chess,Table Tennis'} where
Roll_No=1;
cqlsh:students> update students_info set language=language+['Hindi,English'] where
Roll_No=1;
cqlsh:students> select * from students_info;
```

roll_no	dateofjoining	hobbies	language	last_exam_percent	studname
5	2012-03-11 18:30:00.000000+0000	null	null	67.9	Smitha
1	2012-03-11 18:30:00.000000+0000	{'Chess,Table Tennis'}	['Hindi,English']	79.9	Asha

4   2012-03-11 18:30:00.000000+0000	null	null	90.9
Samarth			
6   2012-03-11 18:30:00.000000+0000	null	null	56.9
Rohan			
3   2012-03-11 18:30:00.000000+0000	null	null	78.9
Tarun			

(5 rows)

```
cqlsh:students> alter table students_info add cval counter;
cqlsh:students> update students_info set cval=cval+1 where Roll_No>2;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Only EQ and IN relation are supported on the partition key (unless you use the token() function) for UPDATE statements"
cqlsh:students> update students_info set cval=cval+1 where Roll_No=4;
cqlsh:students> select * from students_info;
```

roll_no   cval   dateofjoining	hobbies	language		
last_exam_percent   studname				
-----+-----+-----+-----+-----+				
+-----				
5   null   2012-03-11 18:30:00.000000+0000	null	null	67.9	
Smitha				
1   null   2012-03-11 18:30:00.000000+0000   {'Chess,Table Tennis'}   ['Hindi,English']				
79.9   Asha				
4   1   2012-03-11 18:30:00.000000+0000	null	null	90.9	
Samarth				
6   null   2012-03-11 18:30:00.000000+0000	null	null	56.9	
Rohan				
3   null   2012-03-11 18:30:00.000000+0000	null	null	78.9	
Tarun				

(5 rows)

```
cqlsh:students> INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) VALUES (8,'Rola','2012-03-12',56.9) USING TTL 30;
cqlsh:students> SELECT TTL(last_exam_percent) from students_info where roll_no=1;
```

ttl(last\_exam\_percent)

-----  
null

(1 rows)

```
cqlsh:students> COPY Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) to 'week1.csv';
Using 7 child processes
```

Starting copy of students.students\_info with columns [roll\_no, studname, dateofjoining, last\_exam\_percent].

```
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row/s
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
Processed: 5 rows; Rate: 24 rows/s; Avg. rate: 11 rows/s
```

5 rows exported to 1 files in 0.502 seconds.

```
cqlsh:students> COPY Students_Info(Roll_No, StudName, DateOfJoining,
last_exam_Percent) to STDOUT;
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
3,Tarun,2012-03-11 18:30:00.000+0000,78.9
1,Asha,2012-03-11 18:30:00.000+0000,79.9
4,Samarth,2012-03-11 18:30:00.000+0000,90.9
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
6,Rohan,2012-03-11 18:30:00.000+0000,56.9
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
5,Smitha,2012-03-11 18:30:00.000+0000,67.9
```

## C. Screenshot

```
bmcgibnscc-Precision-T1700:~$ cd cassandra/apache-cassandra-3.11.0
bmcgibnscc-Precision-T1700:~/cassandra/apache-cassandra-3.11.0$ cd bin
bmcgibnscc-Precision-T1700:~/cassandra/apache-cassandra-3.11.0/bin$ ./cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
[cqlsh> CREATE KEYSPACE students WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
syntaxexception: line 1:28 mismatched input 'WITH' expecting K_WITM (CREATE KEYSPACE students [WITH])
[cqlsh> CREATE KEYSPACE students WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
AlreadyExists: Keyspace 'students' already exists
[cqlsh> CREATE KEYSPACE students1 WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
[cqlsh> DESCRIBE KEYSPACES;
students      system_auth      system_distributed      system_traces
system        system          system                students1
system_schema system          system                system
system_table  system          system                system
[cqlsh> SELECT * FROM system.schema_keyspaces;
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured table schema_keyspaces"
[cqlsh> SELECT * FROM system.schema_KEYSPACES;
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured table schema_KEYSPACES"
[cqlsh> SELECT * FROM system.schema_keyspaces;
SyntaxException: line 1:20 no viable alternative at Input ','
  (SELECT * FROM [system],...)
[cqlsh> SELECT * FROM system.schema.keyspaces;
keyspace_name      | durable_writes | replication
-----+-----+-----+
system_auth      |      True      | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_schema     |      True      | {'class': 'org.apache.cassandra.locator.LocalStrategy'}
system_table      |      True      | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_distributed |      True      | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system           |      True      | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_traces    |      True      | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}
students         |      True      | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
(7 rows)
[cqlsh> USE students1;
... USE students1;
Improper USE command.
[cqlsh> USE students1;
[cqlsh> students1> create table Student_Info(RollNo int PRIMARY KEY,StudName text,DateOfJoining timestamp,LastExamPercent double);
[cqlsh> students1> DESCRIBE TABLES;
student_info
[cqlsh> students1> DESCRIBE TABLE student_info;
CREATE TABLE students1.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    lastexampercent double,
    studname text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = ('keys': 'ALL', 'rows_per_partition': 'NONE')
    AND comment = ''
    AND compaction = ('class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4')
    AND compression = ('chunk_length_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor')
```

```

student_info
cqlsh:students> DESCRIBE TABLE student_info;
CREATE TABLE students.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    lastexampercent double,
    studname text
) WITH bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND compaction = {
    'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy',
    'max_threshold': '32',
    'min_threshold': '4'
}
AND compression = {'chunk_length_in_kb': '64',
    'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND crc_check_chance = 1.0
AND dclocal_read_repair_chance = 0.1
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.0
AND speculative_retry = '99PERCENTILE';

cqlsh:students> BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'ABCD EFGH','2011-04-26',85.2)
... ...
... BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'ABCD EFGH','2011-04-26',85.2)BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'PQRS PQRS','2021-04-26',64.2)
... ...
... BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'ABCD EFGH','2011-04-26',85.2)BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'PQRS PQRS','2021-04-26',64.2)APPLY BATCH;
syntaxException: line 3:10 mismatched input 'BEGIN' expecting APPLY (...,(1,'ABCD EFGH','2011-04-26',85.2){BEGIN}...)
cqlsh:students> BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'ABCD EFGH','2011-04-26',85.2)BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'PQRS PQRS','2021-04-26',64.2)APPLY BATCH;
syntaxException: line 3:10 mismatched input 'BEGIN' expecting APPLY (...,(1,'ABCD EFGH','2011-04-26',85.2){BEGIN}...)
cqlsh:students> BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'ABCD EFGH','2011-04-26',85.2)BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'PQRS PQRS','2021-04-26',64.2)APPLY BATCH;
syntaxException: line 3:10 mismatched input 'BEGIN' expecting APPLY (...,(1,'ABCD EFGH','2011-04-26',85.2){BEGIN}...)
cqlsh:students> BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'ABCD EFGH','2011-04-26',85.2)BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'PQRS PQRS','2021-04-26',64.2)APPLY BATCH;
syntaxException: line 3:10 mismatched input 'BEGIN' expecting APPLY (...,(1,'ABCD EFGH','2011-04-26',85.2){BEGIN}...)
cqlsh:students> BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'ABCD EFGH','2011-04-26',85.2)BEGIN BATCH INSERT INTO student_info(RollNo,StudName, DateofJoining,LastExamPercent)VALUES(1,'PQRS PQRS','2021-04-26',64.2)APPLY BATCH;
syntaxException: line 3:10 mismatched input 'BEGIN' expecting APPLY (...,(1,'ABCD EFGH','2011-04-26',85.2){BEGIN}...)
cqlsh:students> SELECT * FROM student_info;
rollno | dateofjoining | lastexampercent | studname
-----+-----+-----+-----+
1 | 2021-04-25 18:30:00.000000+0000 | 85.2 | PQRS PQRS

(1 rows)
cqlsh:students> UPDATE Student_Info SET StudName = 'David Sheen' WHERE RollNo = 1;
cqlsh:students> SELECT From Student_Info WHERE RollNo in (1,2,3);
syntaxException: line 1:7 no viable alternative at input 'FROM' (SELECT [FROM]...)
cqlsh:students> SELECT From Student_Info WHERE RollNo in (1);
syntaxException: line 1:7 no viable alternative at input 'FROM' (SELECT [FROM]...)
cqlsh:students> UPDATE Student_Info SET StudName = 'Stephen Fox' WHERE RollNo = 1;
syntaxException: line 1:7 no viable alternative at input 'FROM' (SELECT [FROM]...)
cqlsh:students> SELECT FROM Student_Info WHERE RollNo in (1);
syntaxException: line 1:7 no viable alternative at input 'FROM' (SELECT [FROM]...)
cqlsh:students> SELECT FROM Student_Info WHERE RollNo in (1,2,3);
syntaxException: line 1:7 no viable alternative at input 'FROM' (SELECT [FROM]...)
cqlsh:students> SELECT * FROM Student_Info WHERE RollNo in (1,2,3);

rollno | dateofjoining | lastexampercent | studname
-----+-----+-----+-----+
1 | 2021-04-25 18:30:00.000000+0000 | 85.2 | David Sheen
2 | 2021-04-27 18:30:00.000000+0000 | 85.2 | ABCD EFGH
3 | 2016-05-27 18:30:00.000000+0000 | 84.2 | IJKL

(3 rows)
cqlsh:students> CREATE INDEX ON Student_Info(studname);
cqlsh:students> SELECT * FROM student_info WHERE StudName='Stephen Fox';

rollno | dateofjoining | lastexampercent | studname
-----+-----+-----+-----+
1 | 2021-04-25 18:30:00.000000+0000 | 85.2 | Stephen Fox
2 | 2021-04-27 18:30:00.000000+0000 | 85.2 | ABCD EFGH
3 | 2016-05-27 18:30:00.000000+0000 | 84.2 | IJKL

(0 rows)
cqlsh:students> CREATE INDEX ON Student_Info(LastExamPercent);
cqlsh:students> SELECT RollNo, LastExamPercent FROM Student_info LIMIT 2;

rollno | lastexampercent
-----+-----+
1 | 85.2
2 | 85.2

(2 rows)
cqlsh:students> SELECT RollNo, StudName AS "Name" FROM Student_Info;
Invalid syntax at line 1, char 28
  SELECT RollNo, StudName AS "Name" FROM Student_Info;
          ^
cqlsh:students> SELECT RollNo, StudName AS 'Name' FROM Student_Info;
syntaxException: line 1:27 no viable alternative at input 'Name' (SELECT RollNo, StudName AS ['Name'])
cqlsh:students> SELECT RollNo, StudName AS "Name" FROM Student_Info;

rollno | Name
-----+-----+
1 | David Sheen
2 | ABCD EFGH
3 | IJKL

(3 rows)
cqlsh:students> UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 WHERE RollNo=2;
Invalid syntax at line 1, char 34
  UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 WHERE RollNo=2;
          ^
cqlsh:students> UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 WHERE RollNo=2;
syntaxException: line 1:42 no viable alternative at input ',' (UPDATE Student_Info SET StudName="Samaira",...)
cqlsh:students> UPDATE Student_Info SET StudName="IJKL", LastExamPercent=85.7 W[i]+ Stopped ./cqsh
bnscg@bnscg-Precision-T1700:/cassandra/apache-cassandra-3.11.0/bin$ ./cqsh
Connected to Test Cluster at 127.0.0.1:9042.

```

```

.....
1 | David Sheen
2 | ABCD EFGH
3 | IJKL

(3 rows)
cqlsh:students> UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 WHERE RollNo=2;
Invalid syntax at line 1, char 34
    UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 WHERE RollNo=2;
          ^
cqlsh:students> UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 WHERE RollNo=2;
SyntaxException: line 1:19 no viable alternative at input ',' (UPDATE Student_Info SET StudName=[ "Samair'a" ,...])
cqlsh:students> UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 W[1]; Stopped
/home/ganesha/Downloads/cassandra-3.11.0/bin$ ./cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> USE students
...
... USE students;
Improper USE command.
cqlsh> USE students;
cqlsh:students> UPDATE Student_Info SET StudName="Samaira", LastExamPercent=85.7 WHERE RollNo=2;
SyntaxException: line 1:19 no viable alternative at input ',' (UPDATE Student_Info SET StudName=[ "Samair'a" ,...])
cqlsh:students> UPDATE Student_Info SET StudName="Rutu", LastExamPercent=85.7 WHERE RollNo=2;
SyntaxException: line 1:19 no viable alternative at input ',' (UPDATE Student_Info SET StudName=[ "Rutu" ,...])
cqlsh:students> SELECT RollNo, Studname, LastExamPercent FROM StudentInfo WHERE RollNo=2;
InvalidRequest: Error from server: code=200 [Invalid query] message="unconfigured table studentinfo"
cqlsh:students> SELECT RollNo, Studname, LastExamPercent FROM Student_Info WHERE RollNo=2;
rollno | studname | lastexampercent
-----+-----+-----
2 | Rutu | 85.7

(1 rows)
cqlsh:students> DELETE LastExamPercent FROM Student_Info WHERE RollNo=2;
cqlsh:students> SELECT * FROM StudentInfo WHERE RollNo=2;
InvalidRequest: Error from server: code=200 [Invalid query] message="unconfigured table studentinfo"
cqlsh:students> SELECT * FROM Student_Info WHERE RollNo=2;
rollno | dateofjoining | lastexampercent | studname
-----+-----+-----+-----
2 | 2011-04-27 18:30:00.000000+0000 | null | Rutu

(1 rows)
cqlsh:students> DELETE FROM Student_Info WHERE RollNo=2;
cqlsh:students> SELECT * FROM StudentInfo WHERE RollNo=2;
InvalidRequest: Error from server: code=200 [Invalid query] message="unconfigured table studentinfo"
cqlsh:students> SELECT * FROM Student_Info WHERE RollNo=2;
rollno | dateofjoining | lastexampercent | studname
-----+-----+-----+-----

(0 rows)
cqlsh:students> []

```

## **LAB 2**

**Date: 01-04-2021**

### **A. Question**

A] 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 Employees.

### **B. Commands**

```
cqlsh> CREATE KEYSPACE Employee WITH  
REPLICATION={'class':'SimpleStrategy','replication_factor':1};  
cqlsh> DESCRIBE KEYSPACES
```

```
employee system_auth      system_schema system_views  
system   system_distributed system_traces system_virtual_schema
```

```
cqlsh> USE Employee
```

```
...;
```

```
cqlsh:employee> CREATE TABLE Employee_Info (Emp_id int PRIMARY KEY,  
Emp_Name text,Designation text,
```

```
... Date_Of_Joining timestamp, salary double, Dept_name text);
```

```
cqlsh:employee> DESCRIBE TABLES;
```

```
employee_info
```

```
cqlsh:employee> BEGIN BATCH
```

```
... INSERT INTO Employee_Info(Emp_id, Emp_Name, Designation,  
Date_Of_Joining, salary, Dept_name)
```

```
... VALUES (120,'Asha','Manager','2021-04-01',30000.0,'CSE')
```

```
... INSERT INTO Employee_Info(Emp_id, Emp_Name, Designation,  
Date_Of_Joining, salary, Dept_name)
```

```
... VALUES (121,'Kiran','Emp','2019-04-20',20000.0,'CSE')
```

```
... INSERT INTO Employee_Info(Emp_id, Emp_Name, Designation,  
Date_Of_Joining, salary, Dept_name)
```

```
... VALUES (122,'Tarun','Emp','2019-05-01',22000.0,'CSE')
```

```

... INSERT INTO Employee_Info(Emp_id, Emp_Name, Designation,
Date_Of_Joining, salary, Dept_name)
... VALUES (123,'Samarth','Emp','2020-08-01',22500.0,'CSE')
... INSERT INTO Employee_Info(Emp_id, Emp_Name, Designation,
Date_Of_Joining, salary, Dept_name)
... VALUES (124,'Rohan','Emp','2019-06-01',21000.0,'CSE')
... APPLY BATCH;

```

```
cqlsh:employee> select * from Employee_Info
... ;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
120	2021-04-01 07:00:00.000000+0000	CSE	Manager	Asha	30000
123	2020-08-01 07:00:00.000000+0000	CSE	Emp	Samarth	22500
122	2019-05-01 07:00:00.000000+0000	CSE	Emp	Tarun	22000
121	2019-04-20 07:00:00.000000+0000	CSE	Emp	Kiran	20000
124	2019-06-01 07:00:00.000000+0000	CSE	Emp	Rohan	21000

(5 rows)

```
cqlsh:employee> UPDATE Employee_Info SET Emp_Name='David', Dept_name='ECE'
WHERE Emp_id=121;
cqlsh:employee> select * from Employee_Info
... ;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
120	2021-04-01 07:00:00.000000+0000	CSE	Manager	Asha	30000
123	2020-08-01 07:00:00.000000+0000	CSE	Emp	Samarth	22500
122	2019-05-01 07:00:00.000000+0000	CSE	Emp	Tarun	22000
121	2019-04-20 07:00:00.000000+0000	ECE	Emp	David	20000
124	2019-06-01 07:00:00.000000+0000	CSE	Emp	Rohan	21000

(5 rows)

```
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects text;
cqlsh:employee> select * from Employee_Info;
```

emp_id	salary	date_of_joining	dept_name	designation	emp_name	projects
120	30000	2021-04-01 07:00:00.000000+0000	CSE	Manager	Asha	null
123	22500	2020-08-01 07:00:00.000000+0000	CSE	Emp	Samarth	null
122	22000	2019-05-01 07:00:00.000000+0000	CSE	Emp	Tarun	null
121	20000	2019-04-20 07:00:00.000000+0000	CSE	Emp	Kiran	null
124	21000	2019-06-01 07:00:00.000000+0000	CSE	Emp	Rohan	null

(5 rows)

```
cqlsh:employee> UPDATE Employee_Info SET Projects='Research'WHERE Emp_id=120  
and salary=30000.0;
```

```
cqlsh:employee> select * from Employee_Info;
```

emp_id	salary	date_of_joining	dept_name	designation	emp_name	projects
120	30000	2021-04-01 07:00:00.000000+0000	CSE	Manager	Asha	Research
123	22500	2020-08-01 07:00:00.000000+0000	CSE	Emp	Samarth	null
122	22000	2019-05-01 07:00:00.000000+0000	CSE	Emp	Tarun	null
121	20000	2019-04-20 07:00:00.000000+0000	CSE	Emp	Kiran	null
124	21000	2019-06-01 07:00:00.000000+0000	CSE	Emp	Rohan	null

(5 rows)

```
cqlsh:employee>
```

```
cqlsh:employee> UPDATE Employee_Info SET Projects='Data Migration'WHERE  
Emp_id=123 and salary=22500.0;
```

```
cqlsh:employee> UPDATE Employee_Info SET Projects='Data analysis'WHERE  
Emp_id=122 and salary=22000.0;
```

```
cqlsh:employee> UPDATE Employee_Info SET Projects='Reporting'WHERE Emp_id=121  
and salary=22000.0;
```

```
cqlsh:employee> UPDATE Employee_Info SET Projects='Research'WHERE Emp_id=124  
and salary=21000.0;
```

```
cqlsh:employee> UPDATE Employee_Info SET Projects='Reporting'WHERE Emp_id=121  
and salary=20000.0;
```

```
cqlsh:employee> select * from Employee_Info;
```

emp_id	salary	date_of_joining	dept_name	designation	emp_name	projects
120	30000	2021-04-01 07:00:00.000000+0000	CSE	Manager	Asha	Research
123	22500	2020-08-01 07:00:00.000000+0000	CSE	Emp	Samarth	Data Migration
122	22000	2019-05-01 07:00:00.000000+0000	CSE	Emp	Tarun	Data analysis
121	20000	2019-04-20 07:00:00.000000+0000	CSE	Emp	Kiran	Reporting
124	21000	2019-06-01 07:00:00.000000+0000	CSE	Emp	Rohan	Research

(5 rows)

```
cqlsh:employee> INSERT INTO Employee_Info(Emp_id, Emp_Name, Designation,
Date_Of_Joining, salary, Dept_name) VALUES (125,'Anya','Emp','2021-04-
01',25000.0,'CSE') using ttl 15;
cqlsh:employee> select ttl(Emp_Name) from Employee_Info Where Emp_id=125;

ttl(emp_name)
-----
6
```

### C. Screenshot

```
cqlsh> CREATE KEYSPACE employee WITH REPLICATION={ 'class' : 'SimpleStrategy', 'replication_factor' : 1};
cqlsh> USE employee;
```

```
cqlsh:employee> create table employee_info(emp_id int PRIMARY KEY, emp_name text, designation text, date_of_joining timestamp, salary double, dept_name text);
```

```
cqlsh:employee> BEGIN BATCH INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joi
STING')
... INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joining,salary,o
... INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joining,salary,o
... APPLY BATCH;
cqlsh:employee> SELECT * FROM employee_info;

emp_id | date_of_joining           | dept_name | designation | emp_name | salary
-----+-----+-----+-----+-----+-----+
 111 | 2020-06-21 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | SRIRAM | 25000
 121 | 2020-03-29 18:30:00.000000+0000 |          HR |      MANAGER | SHIVA | 35000
 100 | 2020-09-10 18:30:00.000000+0000 |      TESTING |      MANAGER | TANYA | 30000
(3 rows)
```

```
cqlsh:employee> UPDATE employee_info SET emp_name = 'SHAAN' WHERE emp_id = 121;
cqlsh:employee> SELECT * FROM employee_info;
```

```
emp_id | date_of_joining           | dept_name | designation | emp_name | salary
-----+-----+-----+-----+-----+-----+
 111 | 2020-06-21 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | SRIRAM | 25000
 121 | 2020-03-29 18:30:00.000000+0000 |          HR |      MANAGER | SHAAN | 35000
 100 | 2020-09-10 18:30:00.000000+0000 |      TESTING |      MANAGER | TANYA | 30000
(3 rows)
```

```
cqlsh:employee> ALTER TABLE employee_info ADD projects_done set<text>;
cqlsh:employee> ALTER TABLE employee_info ADD projects_done set<text>;
cqlsh:employee> UPDATE employee_info SET projects_done = {'chat app'} WHERE emp_id = 111;
cqlsh:employee> UPDATE employee_info SET projects_done = {'campusx,flashcard'} WHERE emp_id = 121;
cqlsh:employee> UPDATE employee_info SET projects_done = {'canteen app'} WHERE emp_id = 100;
cqlsh:employee> SELECT * FROM employee_info
    ...

```

emp_id	date_of_joining	dept_name	designation	emp_name	projects_done
111	2020-06-21 18:30:00.000000+0000	DEVELOPING	ASSOCIATE	SIRRAM	{'chat app'}
121	2020-03-29 18:30:00.000000+0000	HR	MANAGER	SHAAN	{'campusx,flashcard'}
100	2020-09-10 18:30:00.000000+0000	TESTING	MANAGER	TANYA	{'canteen app'}

```
(3 rows)

cqlsh:employee> INSERT INTO employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name) VALUES(110,'SAM','ASSOCIATE','2020-01-11',33000,TTL 15);
cqlsh:employee> SELECT TTL(emp_name) from employee_info WHERE emp_id = 110;

```

ttl(emp_name)
13

```
(1 rows)

cqlsh:employee> SELECT * FROM employee_info;

```

emp_id	date_of_joining	dept_name	designation	emp_name	projects_done	salary
110	2020-01-10 18:30:00.000000+0000	TESTING	ASSOCIATE	SAM	null	33000
111	2020-06-21 18:30:00.000000+0000	DEVELOPING	ASSOCIATE	SIRRAM	{'chat app'}	25000
121	2020-03-29 18:30:00.000000+0000	HR	MANAGER	SHAAN	{'campusx,flashcard'}	35000
100	2020-09-10 18:30:00.000000+0000	TESTING	MANAGER	TANYA	{'canteen app'}	30000

```
(4 rows)

cqlsh:employee> SELECT * FROM employee_info
    ...

```

emp_id	date_of_joining	dept_name	designation	emp_name	projects_done	salary
111	2020-06-21 18:30:00.000000+0000	DEVELOPING	ASSOCIATE	SIRRAM	{'chat app'}	25000
121	2020-03-29 18:30:00.000000+0000	HR	MANAGER	SHAAN	{'campusx,flashcard'}	35000
100	2020-09-10 18:30:00.000000+0000	TESTING	MANAGER	TANYA	{'canteen app'}	30000

```
(3 rows)
```

## A. Question

B] 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

## **B.Commands**

```
cqlsh> describe keyspaces;
```

```
employee system_auth      system_schema system_views  
system   system_distributed system_traces system_virtual_schema
```

```
cqlsh> CREATE KEYSPACE Library WITH  
REPLICATION={'class':'SimpleStrategy','replication_factor':1};  
cqlsh> describe keyspaces;
```

```
employee system      system_distributed system_traces system_virtual_schema  
library  system_auth system_schema      system_views
```

```
cqlsh:library> CREATE TABLE Library_Info (student_id int, student_Name  
text,book_name text,book_id int,Date_of_issue timestamp,primary key(student_id));  
cqlsh:library> alter table Library_Info add counter_value counter;  
cqlsh:library> describe tables;
```

```
library_info
```

```
cqlsh:library> BEGIN BATCH  
INSERT INTO Library_Info(student_id, student_Name,  
book_name, book_id, Date_of_issue) VALUES (120,'snehita','BDA',1000,'2021-04-01')  
INSERT INTO Library_Info(student_id, student_Name, book_name, book_id, Date_of_issue)  
VALUES (121,'asha','OOMD',1010,'2021-04-01')  
INSERT INTO Library_Info(student_id, student_Name, book_name, book_id, Date_of_issue)  
VALUES (122,'sakshi','BDA',1000,'2021-04-01')  
INSERT INTO Library_Info(student_id, student_Name, book_name, book_id, Date_of_issue)  
VALUES (123,'kiran','ML',1020,'2021-04-01')  
APPLY BATCH;
```

```
cqlsh:library> select * from Library_Info;
```

student_id	book_id	book_name	counter_value	date_of_issue	student_name
120	1000	BDA	null	2021-04-01 07:00:00.000000+0000	snehita
123	1020	ML	null	2021-04-01 07:00:00.000000+0000	kiran
122	1000	BDA	null	2021-04-01 07:00:00.000000+0000	sakshi
121	1010	OOMD	null	2021-04-01 07:00:00.000000+0000	asha

(4 rows)

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where  
student_id=120;  
cqlsh:library> select * from Library_Info;
```

student_id	book_id	book_name	counter_value	date_of_issue	student_name
120	1000	BDA	1	2021-04-01 07:00:00.000000+0000	snehita
123	1020	ML	1	2021-04-01 07:00:00.000000+0000	kiran
122	1000	BDA	1	2021-04-01 07:00:00.000000+0000	sakshi
121	1010	OOMD	1	2021-04-01 07:00:00.000000+0000	asha

120	1000	BDA	1	2021-04-01 07:00:00.000000+0000	snehita
123	1020	ML	null	2021-04-01 07:00:00.000000+0000	kiran
122	1000	BDA	null	2021-04-01 07:00:00.000000+0000	sakshi
121	1010	OOMD	null	2021-04-01 07:00:00.000000+0000	asha

(4 rows)

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where student_id=120;
```

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where student_id=121;
```

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where student_id=122;
```

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where student_id=123;
```

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where student_id=123;
```

```
cqlsh:library> select * from Library_Info;
```

student_id	book_id	book_name	counter_value	date_of_issue	student_name
120	1000	BDA	2	2021-04-01 07:00:00.000000+0000	snehita
123	1020	ML	2	2021-04-01 07:00:00.000000+0000	kiran
122	1000	BDA	1	2021-04-01 07:00:00.000000+0000	sakshi
121	1010	OOMD	1	2021-04-01 07:00:00.000000+0000	asha

(4 rows)

```
cqlsh:library> select student_id from Library_Info where book_name='BDA' and counter_value=2 allow filtering;
```

```
student_id
```

```
120
```

(1 rows)

```
cqlsh:library> copy
Library_Info(student_id,student_Name,book_name,book_name,book_id,counter_value) to
'd:\week2.csv';
Using 1 child processes
```

Starting copy of library.library\_info with columns [student\_id, student\_name, book\_name, book\_name, book\_id, counter\_value].

```
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
```

```
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
Processed: 4 rows; Rate:    46 rows/s; Avg. rate:    46 rows/s
4 rows exported to 1 files in 0.090 seconds.
cqlsh:library> copy
week2(student_id,student_Name,book_name,book_name,book_id,counter_value) from
'd:\week2.csv';
Column family week2 not found
cqlsh:library> copy
Library_Info(student_id,student_Name,book_name,book_name,book_id,counter_value) from
'd:\week2.csv';
Using 1 child processes
```

Starting copy of library.library\_info with columns [student\_id, student\_name, book\_name, book\_name, book\_id, counter\_value].

Processed: 4 rows; Rate: 7 rows/s; Avg. rate: 10 rows/s
4 rows imported from 1 files in 0.393 seconds (0 skipped).

```
cqlsh:library> copy
Library_Info(student_id,student_Name,book_name,book_name,book_id,counter_value) to
stdout;
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
122,sakshi,BDA,BDA,1000,1
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
120, snehita,BDA,BDA,1000,2
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
121,asha,OOMD,OOMD,1010,1
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
123,kiran,ML,ML,1020,2
```

## C. ScreenShots

```
|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, PR
|ud_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 = '2021-09-20' 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 = '2021-09-20' 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 = 'OOND' and date_of_issue = '2021-09-20' 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	OOND	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 = 112 and stud_name = 'SHAAN' and book_name = 'BDA' and date_of_issue = '2021-09-20' and book_id = 300;
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	OOND	2020-03-31 18:30:00.000000+0000	400	1
112	SHAAN	BDA	2020-09-20 18:30:00.000000+0000	300	2

(3 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_info2(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) FROM 'e:\libraryInfo.csv';
Using 3 child processes
```

Starting copy of library.library\_info2 with columns [stud\_id, stud\_name, book\_name, book\_id, date\_of\_issue, counter\_value].

```
cqlsh:library> create table library_info2(stud_id int, counter_value Counter, stud_name text,book_name text, date_of_issue timestamp, book_id int, P);
cqlsh:library> COPY library_info2(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) FROM 'e:\libraryInfo.csv';
Using 3 child processes
```

Starting copy of library.library\_info2 with columns [stud\_id, stud\_name, book\_name, book\_id, date\_of\_issue, counter\_value].

```
cqlsh:library> SELECT * FROM library_info2;
```

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	OOND	2020-03-31 18:30:00.000000+0000	400	1
112	SHAAN	BDA	2020-09-20 18:30:00.000000+0000	300	2

(3 rows)

## **LAB 3**

**Date: 15-04-2021**

### **A.Question**

1. College Database Collections
1. Student 2. Faculty 3. COE 4. Library 5. Admission 6. College Festival - Minimum 6 attributes (4 values with different set of attribute -value pair) - A set of 4 documents

### **B.Commands**

```
>use college;
>db.createCollection("Student");
>db.createCollection("Faculty");
>db.createCollection("COE");
>db.createCollection("Library");
>db.createCollection("Admission");
db.createCollection("CollegeFestival");
>db.Student.insert({_id:100,Name:"Asha",Dept:"CSE",Designation:21,Dept:"CSE",Mobile:"9980419038"});
>db.Student.insert({_id:2,StudName:"Asha",USN:"1BM18CS090",Age:21,Dept:"CSE",Mobile:"9980419038"});
>db.Student.insert({_id:1,StudName:"Tine",USN:"1BM18CS103",Age:21,Dept:"CSE",Mobile:"9980419038"});
>db.Student.insert({_id:1,StudName:"Aditya",USN:"1BM18CS103",Age:21,Dept:"CSE",Mobile:"9980419038"});

>db.Faculty.insert({_id:100,name:"Asha",Dept:"CSE",Designation:"Professor",salary:20000,Projects:"Research"});
>db.Faculty.insert({_id:110,name:"Rahul",Dept:"CSE",Designation:"Professor",salary:22000,Projects:"Research"});
>db.Faculty.insert({_id:120,name:"Neeti",Dept:"ISE",Designation:"Professor",salary:20000,Projects:"IOT"});
>db.Faculty.insert({_id:130,name:"James",Dept:"CSE",Designation:"HOD",salary:30000,Projects:"Research"});

>db.COE.insert({Name: "Vijay", Dept:"CSE", salary:45000,Designation:"Officer"});
>db.COE.insert({Name: "Prasad", Dept:"ISE", salary:40000,Designation:"Officer"});
>db.COE.insert({Name: "Rita", Dept:"ECE", salary:44000,Designation:"Officer"});
>db.COE.insert({Name: "Anand", Dept:"IEM", salary:40000,Designation:"Officer"});

>db.Library.insert({_BookID: "B123",BookName:"BDA", Dept:"CSE",pageNo:340});
>db.Library.insert({_BookID: "B456",BookName:"ML", Dept:"CSE",pageNo:340});
>db.Library.insert({_BookID: "B789",BookName:"CNS", Dept:"CSE",pageNo:340});
>db.Library.insert({_BookID: "B001",BookName:"Electronics", Dept:"ECE",pageNo:340});

>db.Admission.insert({_ID: "1BM18CS103", StudName:"snehita", Dept:"CSE", fee:120000,date:18-6-2018, type:"Management"});
>db.Admission.insert({_ID: "1BM18CS090", StudName:"Asha", Dept:"CSE", fee:80000,date:18-6-2018, type:"CET"});
```

```

>db.Admission.insert({_ID: "1BM18CS089", StudName:"Tina", Dept:"CSE", fee:125000, date:18-6-2018, type:"Management"});
>db.Admission.insert({_ID: "1BM18CS130", StudName:"Aditya", Dept:"CSE", fee:128000, date:18-6-2018, type:"Management"});

>db.CollegeFestival.insert({Name:"Utsav", Dept:"College",date:18-5-2021, type:"Cultural" , Head:"Principal"});
>db.CollegeFestival.insert({Name:"PhaseShift", Dept:"College",date:18-6-2021, type:"Cultural" , Head:"Principal"});

>db.CollegeFestival.insert({Name:"Protocol", Dept:"CSE",date:18-8-2021, type:"Technical" , Head:"CSEHOD"});
>db.CollegeFestival.insert({Name:"Women's Day",Dept:"College",date:08-03-2021, type:"Cultural" , Head:"Principal"});

```

## C.Screenshots

Key	Value	Type
> (1) ObjectId("607807538490646ff326839c")	{ 6 fields }	Object
< (2) ObjectId("607807538490646ff326839d")	{ 6 fields }	Object
_id	ObjectId("607807538490646ff326839d")	ObjectId
event_name	hackathon	String
event_location	pes	String
partner_name	cloudera	String
start_time	11	String
end_time	12	String
> (3) ObjectId("607807928490646ff326839e")	{ 6 fields }	Object
> (4) ObjectId("607807928490646ff326839f")	{ 6 fields }	Object
	ObjectId("6077fe3b8490646ff326838a"), ObjectId("6077fe3b8490646ff326838b")	
}		
> db.Faculty.insertMany([{"firstName": "Bheemsha", "lastName": "D", "Department": "ME", "Salary": "75000", "YOE": "15", "teaches": "Thermodynamics"}, {"firstName": "Giridhar", "lastName": "Sharan", "Department": "ME", "Salary": "75000", "YOE": "12", "teaches": "Solid Mechanics"}, {"firstName": "Rangavittal", "lastName": "H K", "Department": "ME", "Salary": "85000", "YOE": "17", "teaches": "Sharana"}, {"firstName": "Basavaraj", "lastName": "ME", "Salary": "85000", "YOE": "19", "teaches": "Engineering materials"}]);		
{		
"acknowledged" : true,		
"insertedIds" : [		
ObjectId("6078000f8490646ff326838c"),		
ObjectId("6078000f8490646ff326838d"),		
ObjectId("6078000f8490646ff326838e"),		
ObjectId("6078000f8490646ff326838f")		
]		
}		
> db.COE.insertMany([{"roomNo": "401", "capacity": "60", "building": "Platinum jubilee", "exam_id": "101", "time": "9:00 - 10:15", "faculty": "Bheemsha"}, {"roomNo": "402", "capacity": "60", "building": "Law Block", "exam_id": "102", "time": "11:00-12:15", "faculty": "Giridhar"}, {"roomNo": "204", "capacity": "60", "building": "Architecture", "exam_id": "105", "time": "5:00-6:15", "faculty": "Rangavittal"}]);		
{		
"acknowledged" : true,		
"insertedIds" : [		
ObjectId("607801ad8490646ff3268390"),		
ObjectId("607801ad8490646ff3268391"),		
ObjectId("607801ad8490646ff3268392"),		
ObjectId("607801ad8490646ff3268393")		
]		
}		
> db.Library.insertMany([{"USN": "1bm18cs112", "dueDate": "04/04/2021", "bookISBN": "xyz", "price": "400", "name": "SumanthKV", "title": "OOPS"}, {"USN": "1bm18cs134", "price": "300", "name": "ArunTS", "title": "Cloud Computing"}, {"USN": "ibm18ec034", "dueDate": "06/05/2021", "bookISBN": "qwerty", "price": "556", "name": "Kiran", "titl		

```

> db.Library.find({"dueDate": "05/05/2021"})
> db.Student.find({"lastName": "KV"}, { $set: { "Place": "Sagar Shimoga" } })
Error: error: {
  "ok" : 0,
  "errmsg" : "FieldPath field names may not start with '$'.",
  "code" : 16410,
  "codeName" : "Location16410"
}
> db.Student.find({"lastName": "KV"}, { $set: { "Place": "Sagar Shimoga" } })
Error: error: {
  "ok" : 0,
  "errmsg" : "FieldPath field names may not start with '$'.",
  "code" : 16410,
  "codeName" : "Location16410"
}
> db.Student.updateOne({ "lastName": "KV"}, { $set: { "Place": "Sagar Shimoga" } })
{
  "acknowledged" : true,
  "matchedCount" : 1,
  "modifiedCount" : 1
}
> db.Student.deleteOne({ "firstName": "Taru" })
{
  "acknowledged" : true,
  "deletedCount" : 1
}

```

db.getCollection('Student').find({})

Student 0.001 sec.

Key	Value	Type
> (1) ObjectId("6077fd778490646ff3268388")	{ 7 fields }	Object
> (2) ObjectId("6077fe3b8490646ff3268389")	{ 7 fields }	Object
> (3) ObjectId("6077fe3b8490646ff326838a")	{ 7 fields }	Object

Key	Value
> (1) ObjectId("607801ad8490646ff3268390")	{ 7 fields }
< (2) ObjectId("607801ad8490646ff3268391")	{ 7 fields }
_id	ObjectId("607801ad8490646ff3268391")
roomNo	502
capacity	55
building	ME block
exam_id	102
time	11:00-12:15
faculty	Giridhar
> (3) ObjectId("607801ad8490646ff3268392")	{ 7 fields }
> (4) ObjectId("607801ad8490646ff3268393")	{ 7 fields }

db.getCollection('CollegeFestival').find({})

CollegeFestival 0.001 sec.

Key	Value
> (1) ObjectId("607807538490646ff326839c")	{ 6 fields }
< (2) ObjectId("607807538490646ff326839d")	{ 6 fields }
_id	ObjectId("607807538490646ff326839d")
event_name	hackathon
event_location	pes
partner_name	cloudera
start_time	11
end_time	12
> (3) ObjectId("607807928490646ff326839e")	{ 6 fields }
> (4) ObjectId("607807928490646ff326839f")	{ 6 fields }

```
db.getCollection('Faculty').find({ })
```

Faculty 0.001 sec.

Key	Value
> (1) ObjectId("6078000f8490646ff326838c")	{ 7 fields }
(2) ObjectId("6078000f8490646ff326838d")	{ 7 fields }
_id	ObjectId("6078000f8490646ff326838d")
firstName	Giridhar
lastName	G
Department	ME
Salary	75000
YOE	12
teaches	Solid Mechanics
> (3) ObjectId("6078000f8490646ff326838e")	{ 7 fields }
> (4) ObjectId("6078000f8490646ff326838f")	{ 7 fields }

Student 0.001 sec.

Key	Value
> (1) ObjectId("6077fd778490646ff3268388")	{ 7 fields }
(2) ObjectId("6077fe3b8490646ff3268389")	{ 7 fields }
_id	ObjectId("6077fe3b8490646ff3268389")
firstName	Asha
lastName	S
USN	1BM18CS003
department	CSE
Place	Shimoga
DOB	02/04/2000
> (3) ObjectId("6077fe3b8490646ff326838a")	{ 7 fields }