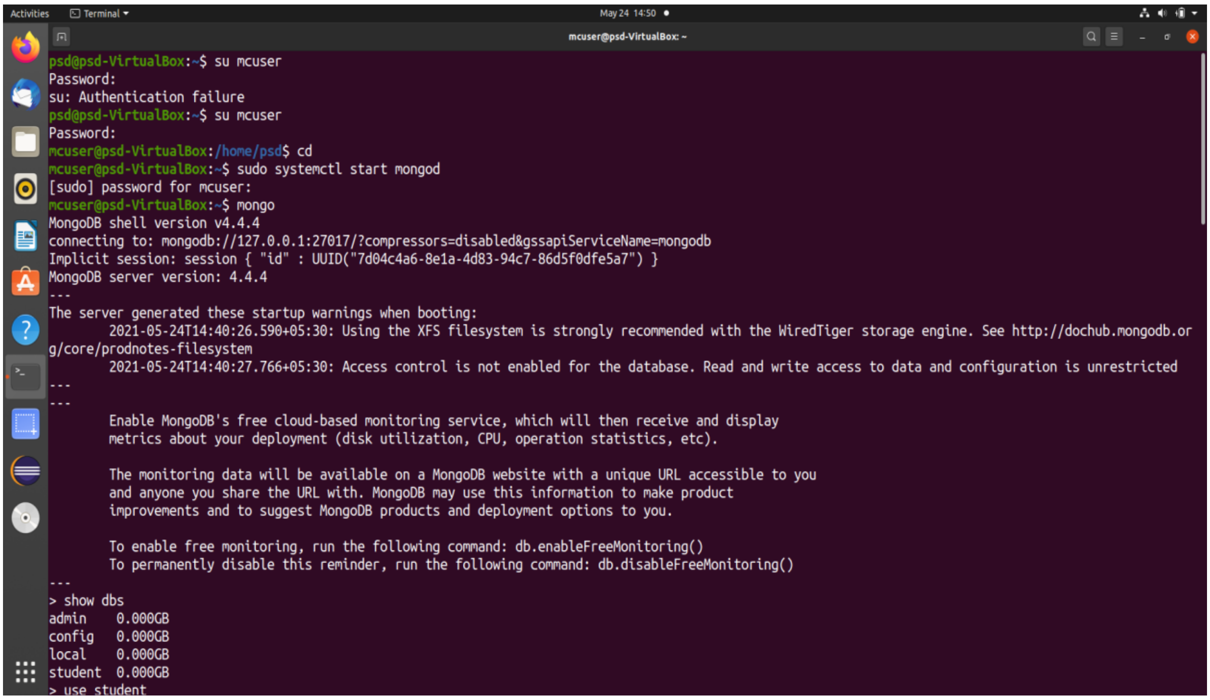
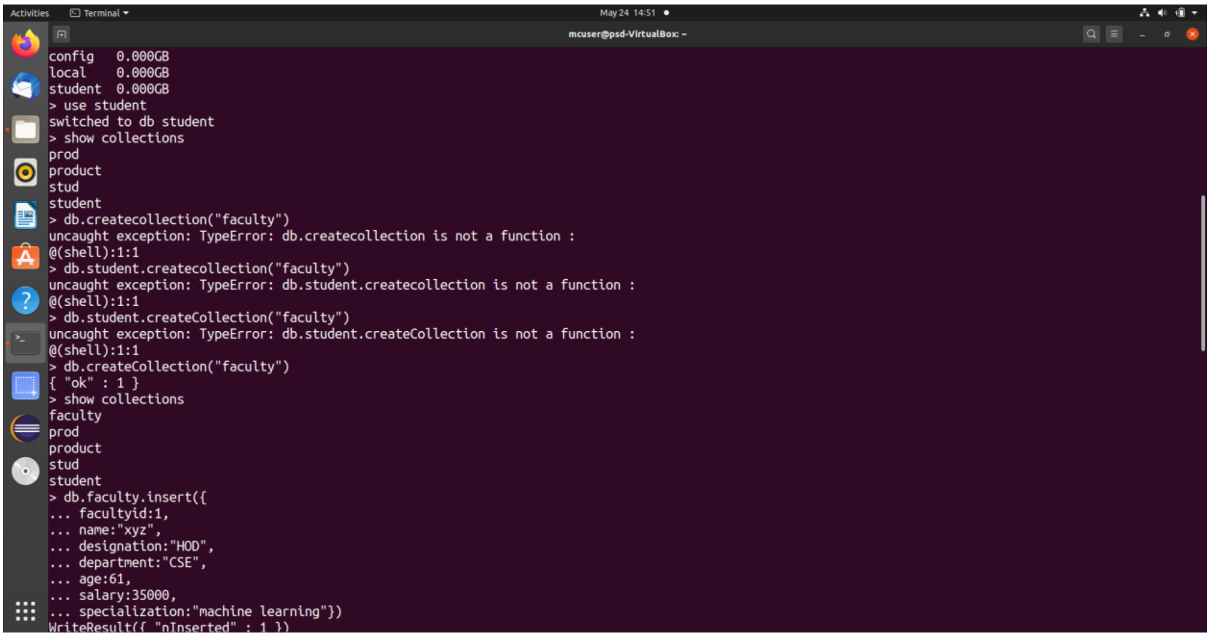
P Sai Deekshith
1BM18CS148 BDA LAB CIE-1

MongoDB Outputs

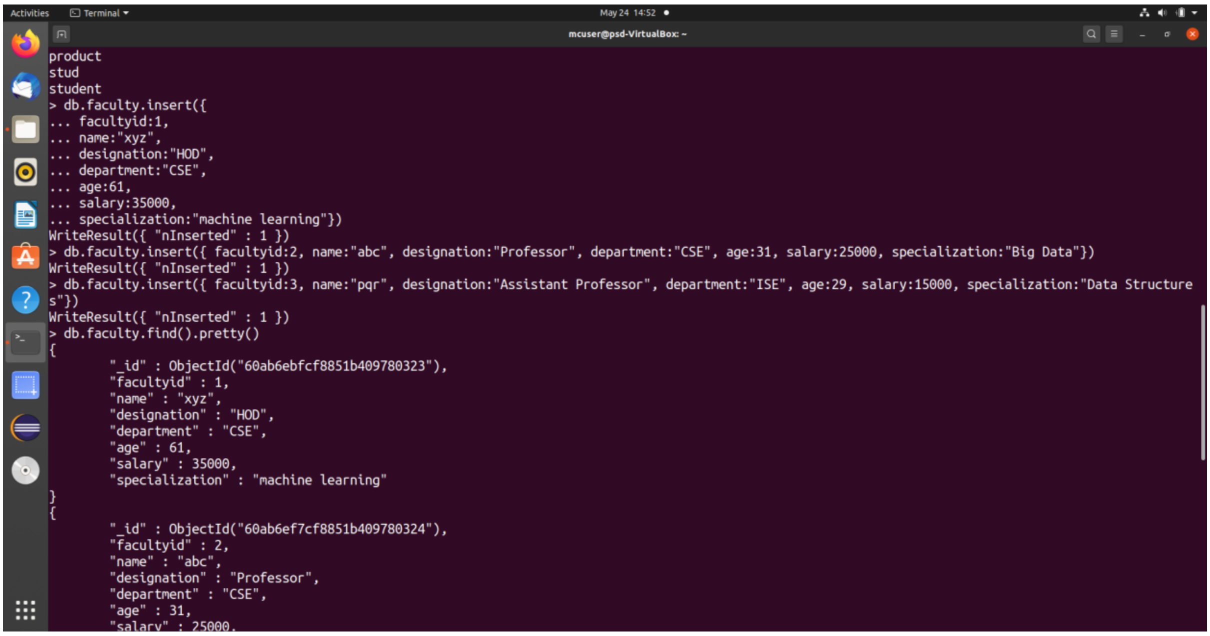


```
psd@psd-VirtualBox:~$ su mcuser
Password:
su: Authentication failure
psd@psd-VirtualBox:~$ su mcuser
Password:
mcuser@psd-VirtualBox:/home/psd$ cd
mcuser@psd-VirtualBox:~$ sudo systemctl start mongod
[sudo] password for mcuser:
mcuser@psd-VirtualBox:~$ mongo
MongoDB shell version v4.4.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7d04c4a6-8e1a-4d83-94c7-86d5f0dfe5a7") }
MongoDB server version: 4.4.4
---
The server generated these startup warnings when booting:
        2021-05-24T14:40:26.590+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.or
g/core/prodnotes-filesystem
        2021-05-24T14:40:27.766+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        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()
---
> show dbs
admin    0.000GB
config   0.000GB
local    0.000GB
student  0.000GB
> use student
```

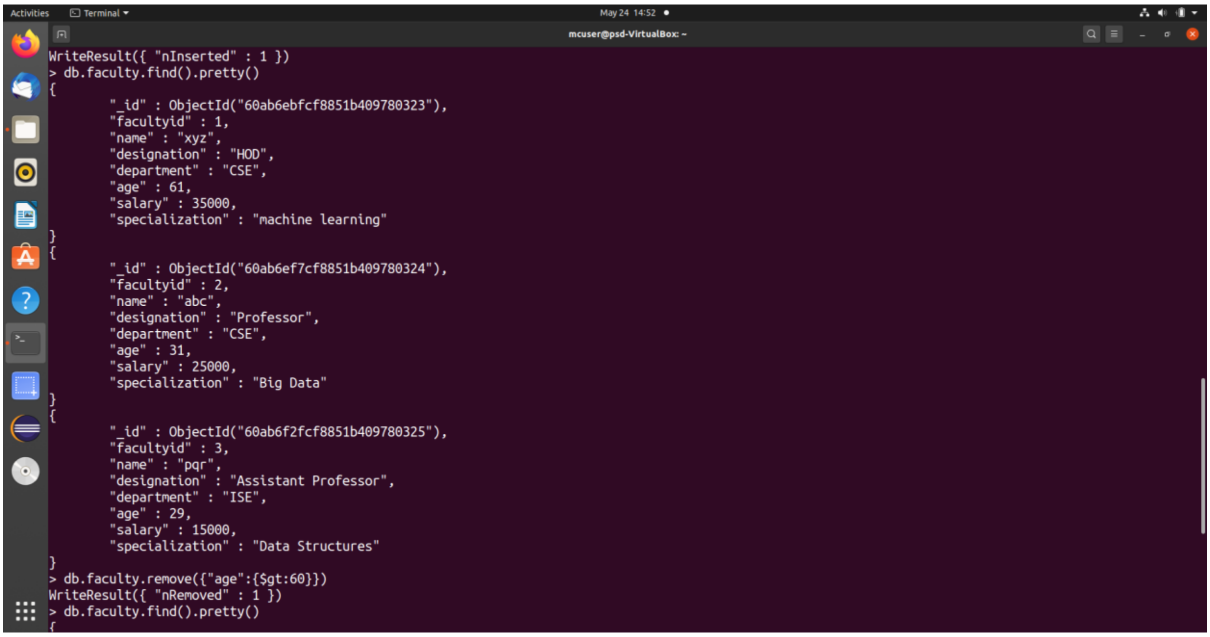


```
config   0.000GB
local    0.000GB
student  0.000GB
> use student
switched to db student
> show collections
prod
product
stud
student
> db.createcollection("faculty")
uncaught exception: TypeError: db.createcollection is not a function :
@(shell):1:1
> db.student.createcollection("faculty")
uncaught exception: TypeError: db.student.createcollection is not a function :
@(shell):1:1
> db.student.createCollection("faculty")
uncaught exception: TypeError: db.student.createCollection is not a function :
@(shell):1:1
> db.createCollection("faculty")
{ "ok" : 1 }
> show collections
faculty
prod
product
stud
student
> db.faculty.insert({
... facultyid:1,
... name:"xyz",
... designation:"HOD",
... department:"CSE",
... age:61,
... salary:35000,
... specialization:"machine learning"})
WriteResult({ "nInserted" : 1 })
```

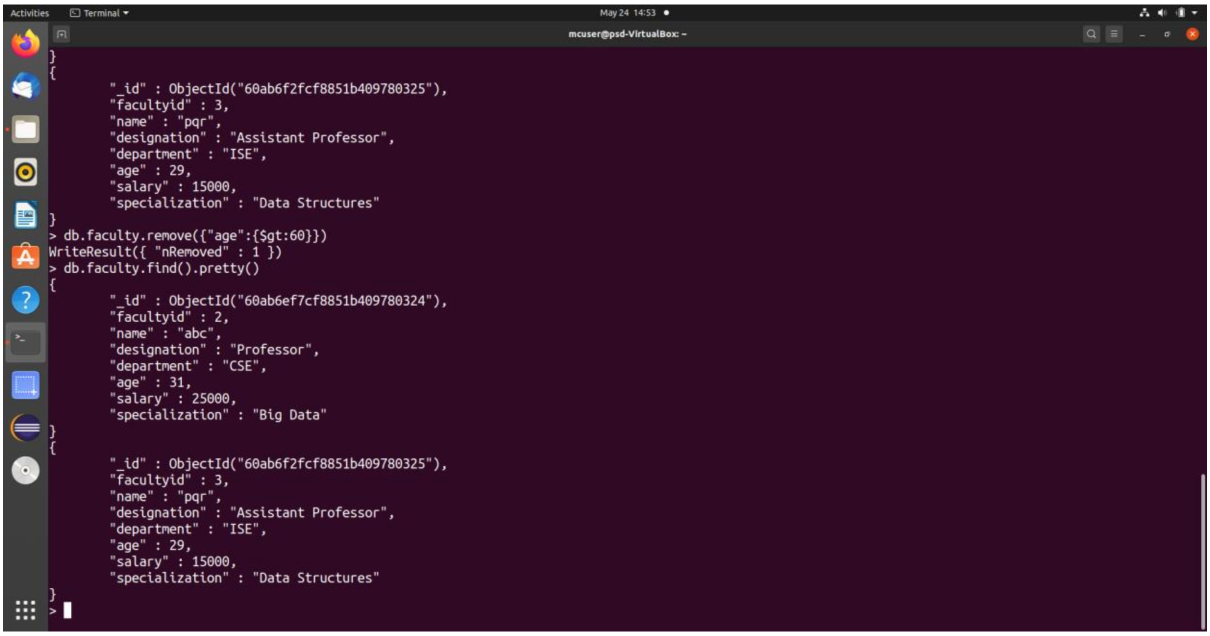
P Sai Deekshith
1BM18CS148 BDA LAB CIE-1



```
product
stud
student
> db.faculty.insert({
... facultyid:1,
... name:"xyz",
... designation:"HOD",
... department:"CSE",
... age:61,
... salary:35000,
... specialization:"machine learning"})
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({ facultyid:2, name:"abc", designation:"Professor", department:"CSE", age:31, salary:25000, specialization:"Big Data"})
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({ facultyid:3, name:"pqr", designation:"Assistant Professor", department:"ISE", age:29, salary:15000, specialization:"Data Structure
s"})
WriteResult({ "nInserted" : 1 })
> db.faculty.find().pretty()
{
        "_id" : ObjectId("60ab6ebfcf8851b409780323"),
        "facultyid" : 1,
        "name" : "xyz",
        "designation" : "HOD",
        "department" : "CSE",
        "age" : 61,
        "salary" : 35000,
        "specialization" : "machine learning"
}
{
        "_id" : ObjectId("60ab6ef7cf8851b409780324"),
        "facultyid" : 2,
        "name" : "abc",
        "designation" : "Professor",
        "department" : "CSE",
        "age" : 31,
        "salary" : 25000,
```



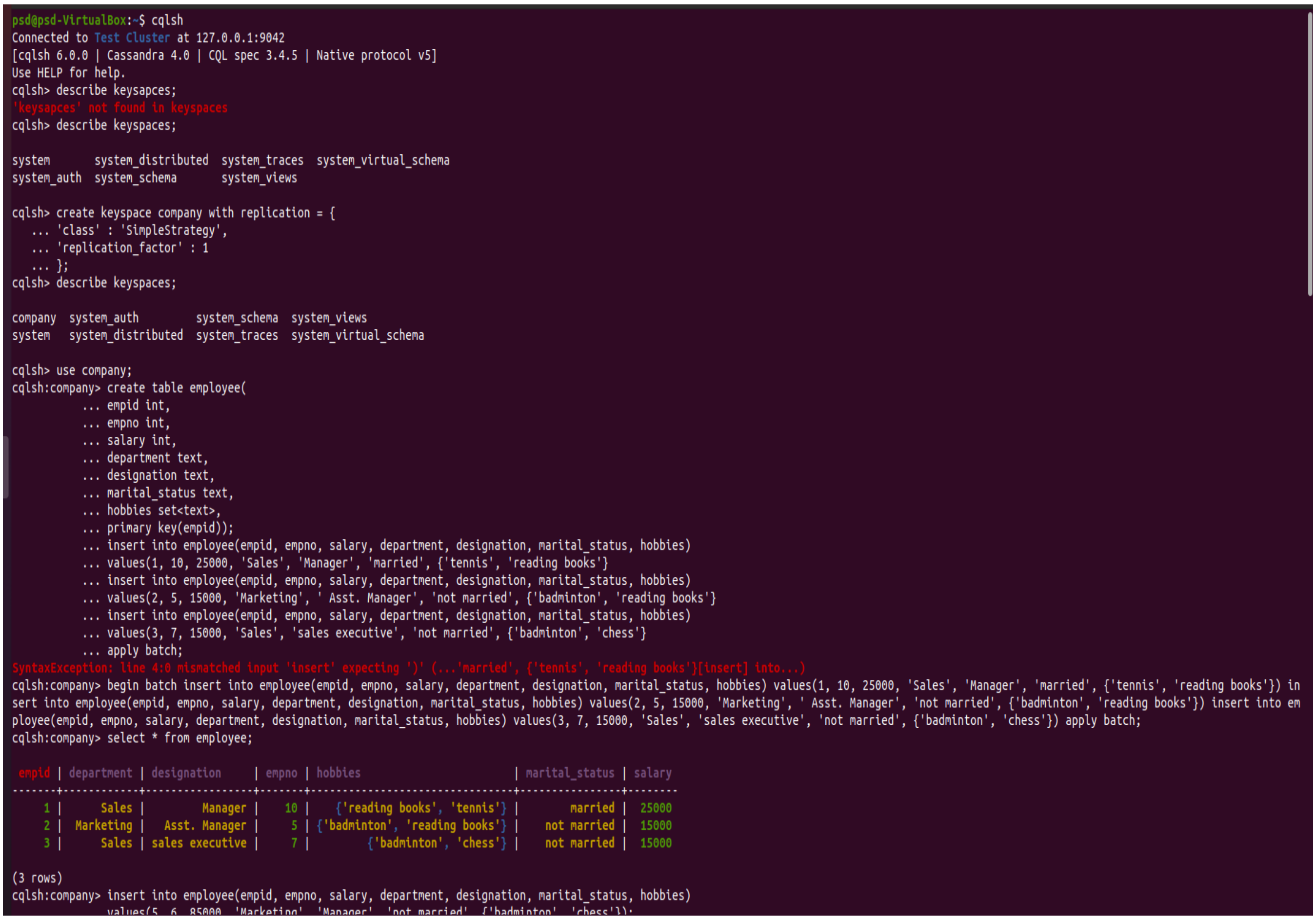
```
WriteResult({ "nInserted" : 1 })
> db.faculty.find().pretty()
{
        "_id" : ObjectId("60ab6ebfcf8851b409780323"),
        "facultyid" : 1,
        "name" : "xyz",
        "designation" : "HOD",
        "department" : "CSE",
        "age" : 61,
        "salary" : 35000,
        "specialization" : "machine learning"
}
{
        "_id" : ObjectId("60ab6ef7cf8851b409780324"),
        "facultyid" : 2,
        "name" : "abc",
        "designation" : "Professor",
        "department" : "CSE",
        "age" : 31,
        "salary" : 25000,
        "specialization" : "Big Data"
}
{
        "_id" : ObjectId("60ab6f2fcf8851b409780325"),
        "facultyid" : 3,
        "name" : "pqr",
        "designation" : "Assistant Professor",
        "department" : "ISE",
        "age" : 29,
        "salary" : 15000,
        "specialization" : "Data Structures"
}
> db.faculty.remove({"age":{$gt:60}})
WriteResult({ "nRemoved" : 1 })
> db.faculty.find().pretty()
{
```



```
}
{
        "_id" : ObjectId("60ab6f2fcf8851b409780325"),
        "facultyid" : 3,
        "name" : "pqr",
        "designation" : "Assistant Professor",
        "department" : "ISE",
        "age" : 29,
        "salary" : 15000,
        "specialization" : "Data Structures"
}
> db.faculty.remove({"age":{$gt:60}})
WriteResult({ "nRemoved" : 1 })
> db.faculty.find().pretty()
{
        "_id" : ObjectId("60ab6ef7cf8851b409780324"),
        "facultyid" : 2,
        "name" : "abc",
        "designation" : "Professor",
        "department" : "CSE",
        "age" : 31,
        "salary" : 25000,
        "specialization" : "Big Data"
}
{
        "_id" : ObjectId("60ab6f2fcf8851b409780325"),
        "facultyid" : 3,
        "name" : "pqr",
        "designation" : "Assistant Professor",
        "department" : "ISE",
        "age" : 29,
        "salary" : 15000,
        "specialization" : "Data Structures"
}
> 
```

P Sai Deekshith
1BM18CS148 BDA LAB CIE-1

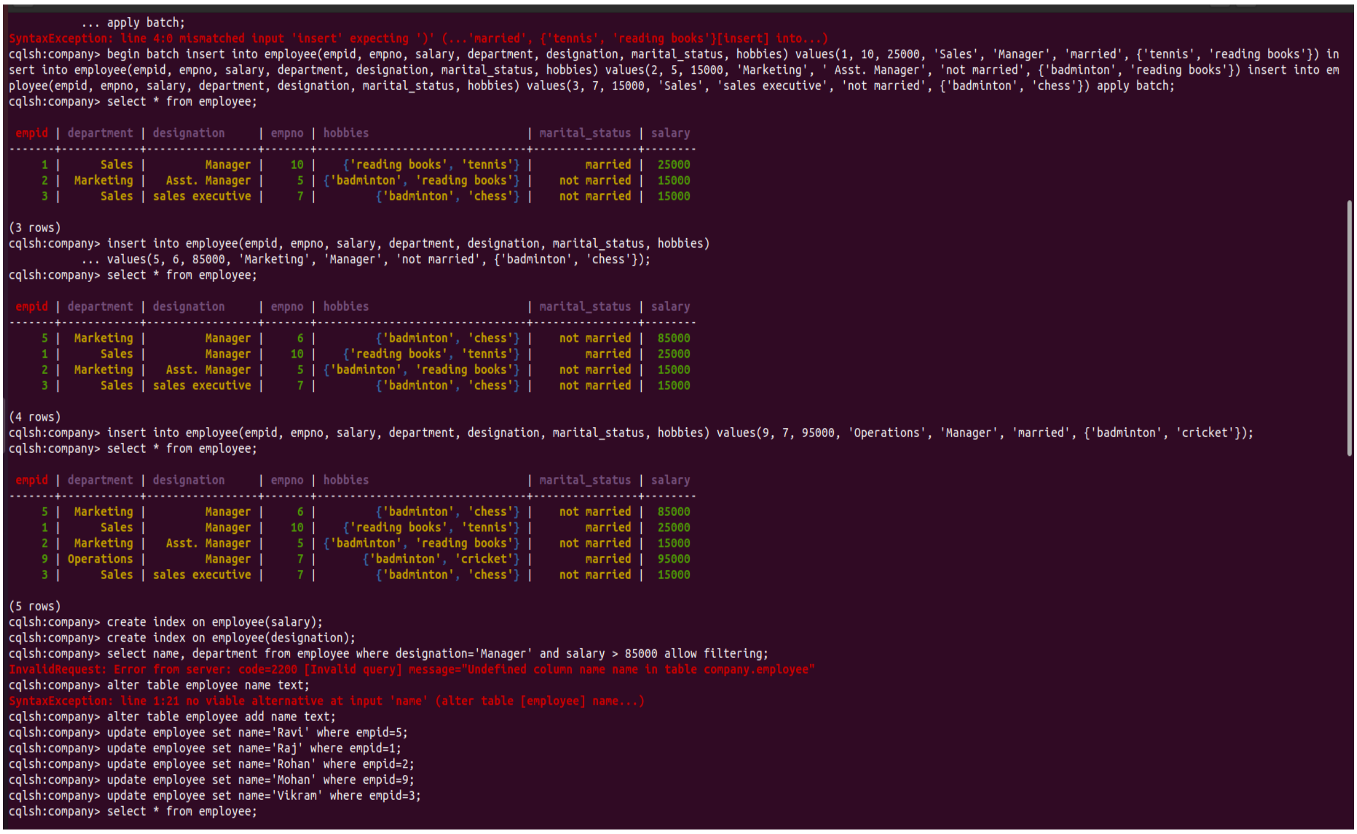
Cassandra Outputs



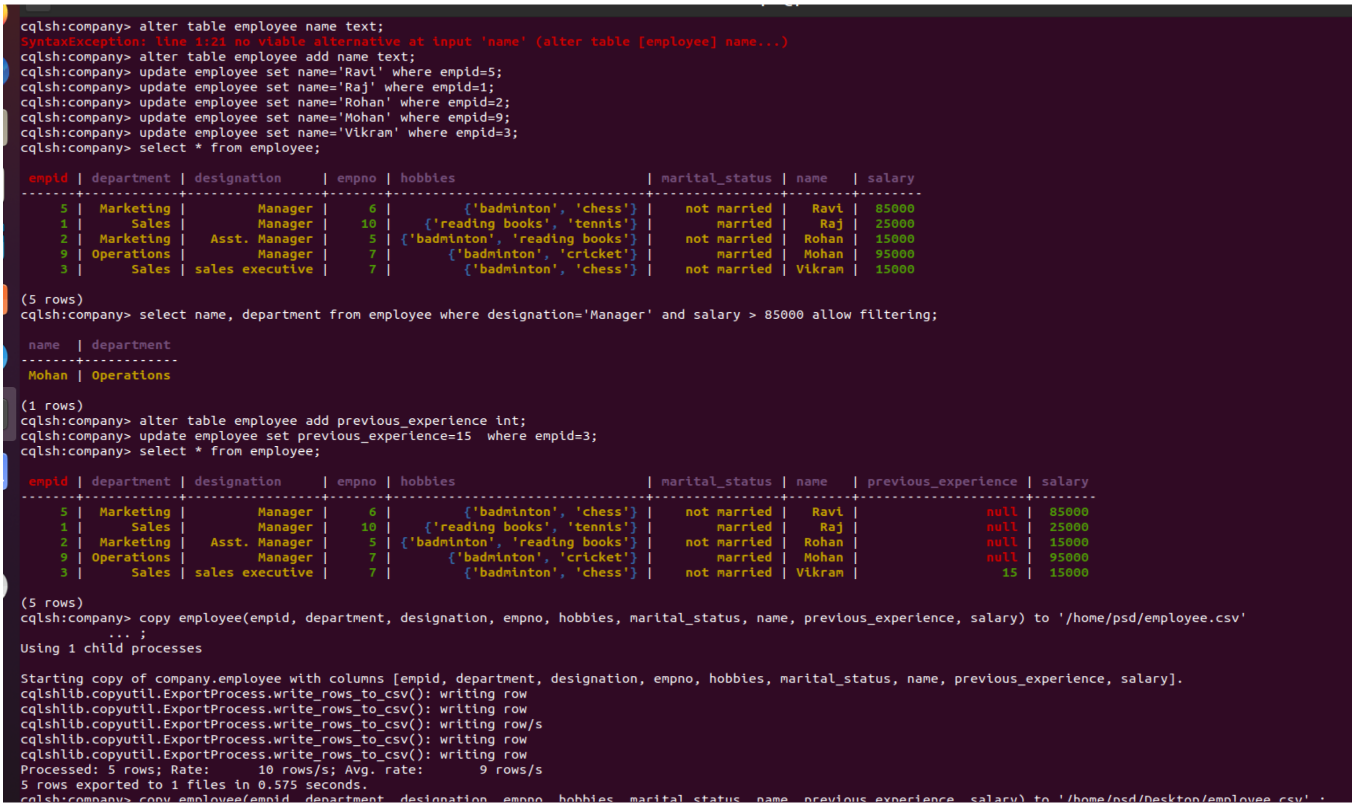
```
psd@psd-VirtualBox:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> describe keysapces;
'keysapces' not found in keyspaces
cqlsh> describe keyspaces;

system       system_distributed  system_traces  system_virtual_schema
system_auth  system_schema       system_views

cqlsh> create keyspace company with replication = {
   ... 'class' : 'SimpleStrategy',
   ... 'replication_factor' : 1
   ... };
cqlsh> describe keyspaces;

company  system_auth         system_schema  system_views
system   system_distributed  system_traces  system_virtual_schema

cqlsh> use company;
cqlsh:company> create table employee(
           ... empid int,
           ... empno int,
           ... salary int,
           ... department text,
           ... designation text,
           ... marital_status text,
           ... hobbies set<text>,
           ... primary key(empid));
           ... insert into employee(empid, empno, salary, department, designation, marital_status, hobbies)
           ... values(1, 10, 25000, 'Sales', 'Manager', 'married', {'tennis', 'reading books'}
           ... insert into employee(empid, empno, salary, department, designation, marital_status, hobbies)
           ... values(2, 5, 15000, 'Marketing', ' Asst. Manager', 'not married', {'badminton', 'reading books'}
           ... insert into employee(empid, empno, salary, department, designation, marital_status, hobbies)
           ... values(3, 7, 15000, 'Sales', 'sales executive', 'not married', {'badminton', 'chess'}
           ... apply batch;
SyntaxException: line 4:0 mismatched input 'insert' expecting ')' (...'married', {'tennis', 'reading books'}[insert] into...)
cqlsh:company> begin batch insert into employee(empid, empno, salary, department, designation, marital_status, hobbies) values(1, 10, 25000, 'Sales', 'Manager', 'married', {'tennis', 'reading books'}) in
sert into employee(empid, empno, salary, department, designation, marital_status, hobbies) values(2, 5, 15000, 'Marketing', ' Asst. Manager', 'not married', {'badminton', 'reading books'}) insert into em
ployee(empid, empno, salary, department, designation, marital_status, hobbies) values(3, 7, 15000, 'Sales', 'sales executive', 'not married', {'badminton', 'chess'}) apply batch;
cqlsh:company> select * from employee;

 empid | department | designation     | empno | hobbies                        | marital_status | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  15000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married |  15000

(3 rows)
cqlsh:company> insert into employee(empid, empno, salary, department, designation, marital_status, hobbies)
           values(5, 6, 85000, 'Marketing', 'Manager', 'not married', {'badminton', 'chess'});
```

P Sai Deekshith

1BM18CS148 BDA LAB CIE-1



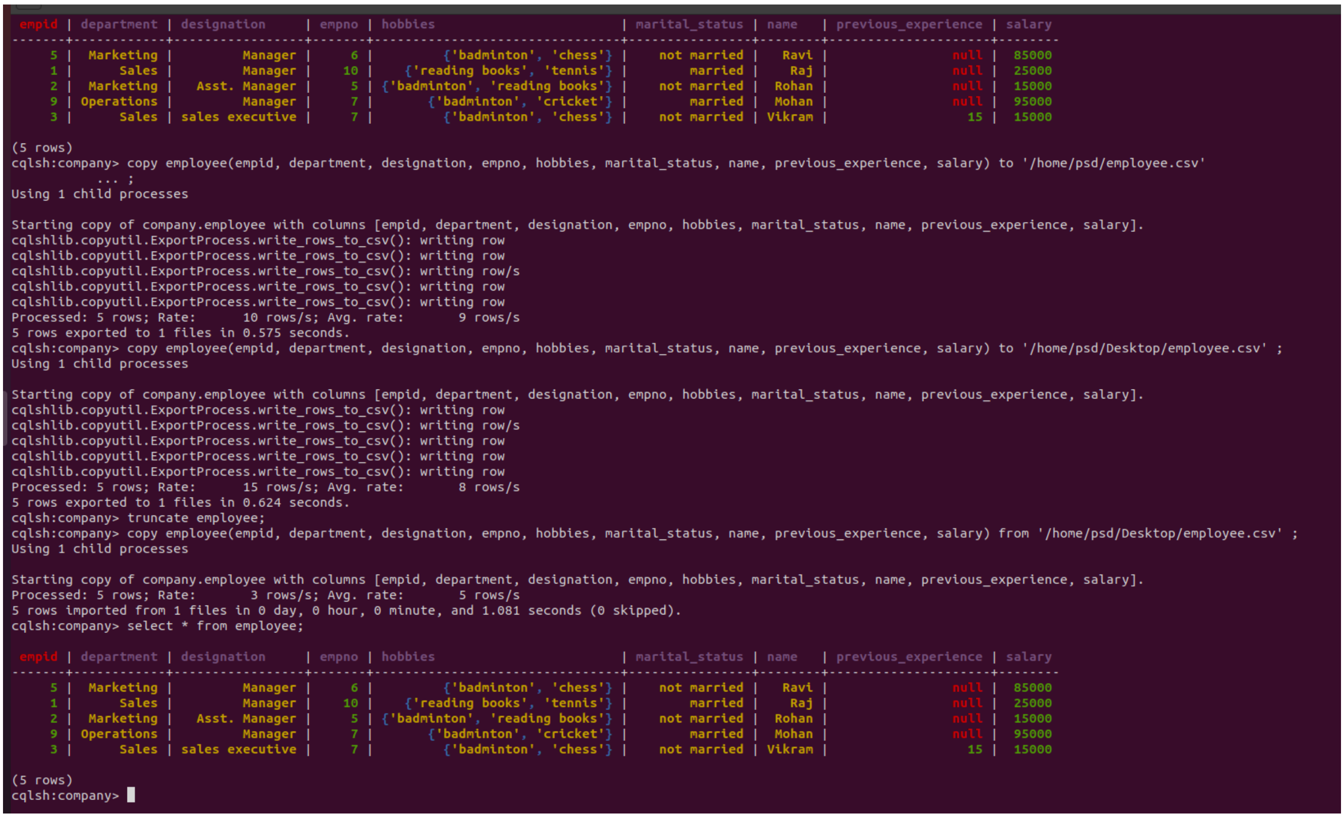
```
         ... apply batch;
SyntaxException: line 4:0 mismatched input 'insert' expecting ')' (...'married', {'tennis', 'reading books'}[insert] into...)
cqlsh:company> begin batch insert into employee(empid, empno, salary, department, designation, marital_status, hobbies) values(1, 10, 25000, 'Sales', 'Manager', 'married', {'tennis', 'reading books'}) in
sert into employee(empid, empno, salary, department, designation, marital_status, hobbies) values(2, 5, 15000, 'Marketing', ' Asst. Manager', 'not married', {'badminton', 'reading books'}) insert into em
ployee(empid, empno, salary, department, designation, marital_status, hobbies) values(3, 7, 15000, 'Sales', 'sales executive', 'not married', {'badminton', 'chess'}) apply batch;
cqlsh:company> select * from employee;

 empid | department | designation     | empno | hobbies                        | marital_status | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  15000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married |  15000

(3 rows)
cqlsh:company> insert into employee(empid, empno, salary, department, designation, marital_status, hobbies)
           ... values(5, 6, 85000, 'Marketing', 'Manager', 'not married', {'badminton', 'chess'});
cqlsh:company> select * from employee;

 empid | department | designation     | empno | hobbies                        | marital_status | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------
     5 |  Marketing |         Manager |     6 |         {'badminton', 'chess'} |    not married |  85000
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  15000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married |  15000

(4 rows)
cqlsh:company> insert into employee(empid, empno, salary, department, designation, marital_status, hobbies) values(9, 7, 95000, 'Operations', 'Manager', 'married', {'badminton', 'cricket'});
cqlsh:company> select * from employee;

 empid | department | designation     | empno | hobbies                        | marital_status | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------
     5 |  Marketing |         Manager |     6 |         {'badminton', 'chess'} |    not married |  85000
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  15000
     9 | Operations |         Manager |     7 |       {'badminton', 'cricket'} |        married |  95000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married |  15000

(5 rows)
cqlsh:company> create index on employee(salary);
cqlsh:company> create index on employee(designation);
cqlsh:company> select name, department from employee where designation='Manager' and salary > 85000 allow filtering;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Undefined column name name in table company.employee"
cqlsh:company> alter table employee name text;
SyntaxException: line 1:21 no viable alternative at input 'name' (alter table [employee] name...)
cqlsh:company> alter table employee add name text;
cqlsh:company> update employee set name='Ravi' where empid=5;
cqlsh:company> update employee set name='Raj' where empid=1;
cqlsh:company> update employee set name='Rohan' where empid=2;
cqlsh:company> update employee set name='Mohan' where empid=9;
cqlsh:company> update employee set name='Vikram' where empid=3;
cqlsh:company> select * from employee;
```



```
cqlsh:company> alter table employee name text;
SyntaxException: line 1:21 no viable alternative at input 'name' (alter table [employee] name...)
cqlsh:company> alter table employee add name text;
cqlsh:company> update employee set name='Ravi' where empid=5;
cqlsh:company> update employee set name='Raj' where empid=1;
cqlsh:company> update employee set name='Rohan' where empid=2;
cqlsh:company> update employee set name='Mohan' where empid=9;
cqlsh:company> update employee set name='Vikram' where empid=3;
cqlsh:company> select * from employee;

 empid | department | designation     | empno | hobbies                        | marital_status | name   | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------+--------
     5 |  Marketing |         Manager |     6 |         {'badminton', 'chess'} |    not married |   Ravi |  85000
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |    Raj |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  Rohan |  15000
     9 | Operations |         Manager |     7 |       {'badminton', 'cricket'} |        married |  Mohan |  95000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married | Vikram |  15000

(5 rows)
cqlsh:company> select name, department from employee where designation='Manager' and salary > 85000 allow filtering;

 name  | department
-------+------------
 Mohan | Operations

(1 rows)
cqlsh:company> alter table employee add previous_experience int;
cqlsh:company> update employee set previous_experience=15  where empid=3;
cqlsh:company> select * from employee;

 empid | department | designation     | empno | hobbies                        | marital_status | name   | previous_experience | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------+---------------------+--------
     5 |  Marketing |         Manager |     6 |         {'badminton', 'chess'} |    not married |   Ravi |                null |  85000
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |    Raj |                null |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  Rohan |                null |  15000
     9 | Operations |         Manager |     7 |       {'badminton', 'cricket'} |        married |  Mohan |                null |  95000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married | Vikram |                  15 |  15000

(5 rows)
cqlsh:company> copy employee(empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary) to '/home/psd/employee.csv'
           ... ;
Using 1 child processes

Starting copy of company.employee with columns [empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary].
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/s
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
Processed: 5 rows; Rate:      10 rows/s; Avg. rate:       9 rows/s
5 rows exported to 1 files in 0.575 seconds.
cqlsh:company> copy employee(empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary) to '/home/psd/Desktop/employee.csv' ;
```

P Sai Deekshith

1BM18CS148 BDA LAB CIE-1



```
 empid | department | designation     | empno | hobbies                        | marital_status | name   | previous_experience | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------+---------------------+--------
     5 |  Marketing |         Manager |     6 |         {'badminton', 'chess'} |    not married |   Ravi |                null |  85000
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |    Raj |                null |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  Rohan |                null |  15000
     9 | Operations |         Manager |     7 |       {'badminton', 'cricket'} |        married |  Mohan |                null |  95000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married | Vikram |                  15 |  15000

(5 rows)
cqlsh:company> copy employee(empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary) to '/home/psd/employee.csv'
           ... ;
Using 1 child processes

Starting copy of company.employee with columns [empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary].
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/s
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
cqlshlib.copyutil.ExportProcess.write_rows_to_csv(): writing row
Processed: 5 rows; Rate:      10 rows/s; Avg. rate:       9 rows/s
5 rows exported to 1 files in 0.575 seconds.
cqlsh:company> copy employee(empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary) to '/home/psd/Desktop/employee.csv' ;
Using 1 child processes

Starting copy of company.employee with columns [empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary].
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:      15 rows/s; Avg. rate:       8 rows/s
5 rows exported to 1 files in 0.624 seconds.
cqlsh:company> truncate employee;
cqlsh:company> copy employee(empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary) from '/home/psd/Desktop/employee.csv' ;
Using 1 child processes

Starting copy of company.employee with columns [empid, department, designation, empno, hobbies, marital_status, name, previous_experience, salary].
Processed: 5 rows; Rate:       3 rows/s; Avg. rate:       5 rows/s
5 rows imported from 1 files in 0 day, 0 hour, 0 minute, and 1.081 seconds (0 skipped).
cqlsh:company> select * from employee;

 empid | department | designation     | empno | hobbies                        | marital_status | name   | previous_experience | salary
-------+------------+-----------------+-------+--------------------------------+----------------+--------+---------------------+--------
     5 |  Marketing |         Manager |     6 |         {'badminton', 'chess'} |    not married |   Ravi |                null |  85000
     1 |      Sales |         Manager |    10 |    {'reading books', 'tennis'} |        married |    Raj |                null |  25000
     2 |  Marketing |   Asst. Manager |     5 | {'badminton', 'reading books'} |    not married |  Rohan |                null |  15000
     9 | Operations |         Manager |     7 |       {'badminton', 'cricket'} |        married |  Mohan |                null |  95000
     3 |      Sales | sales executive |     7 |         {'badminton', 'chess'} |    not married | Vikram |                  15 |  15000

(5 rows)
cqlsh:company> 
```