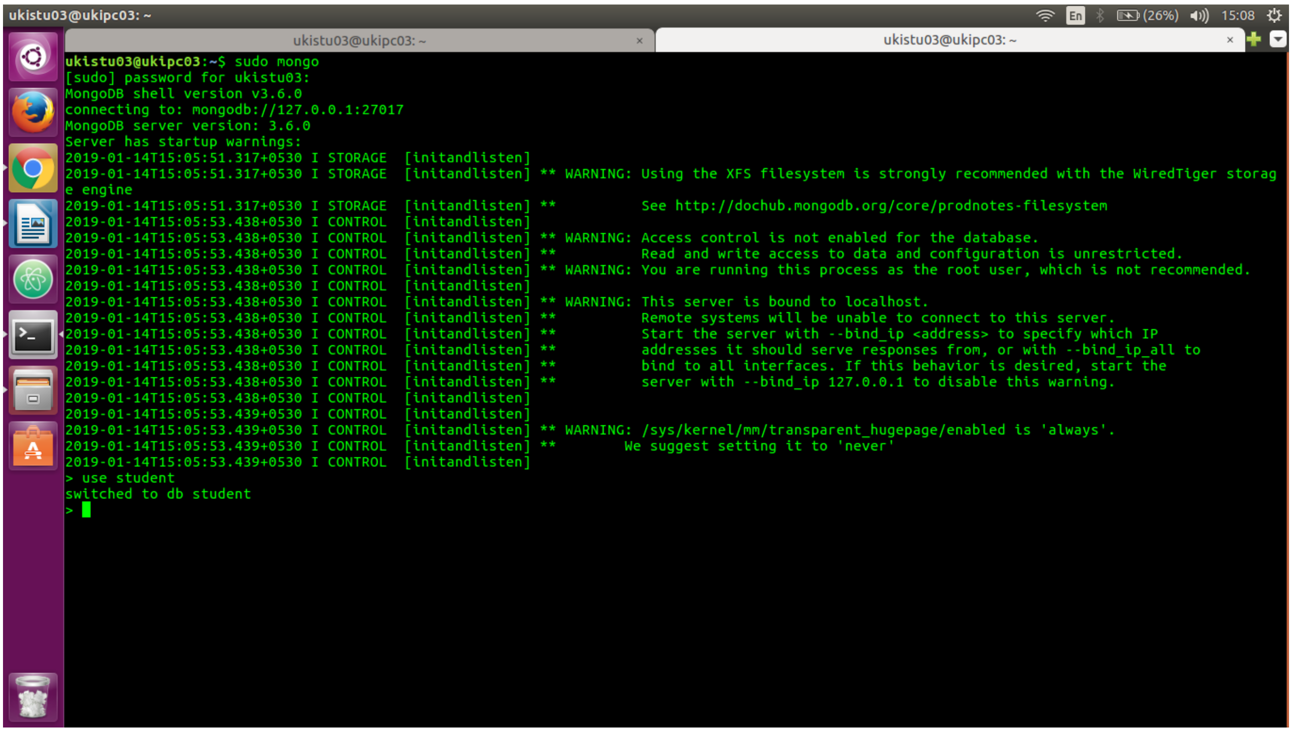
1) Create a Database called student

> use student

switched to db student

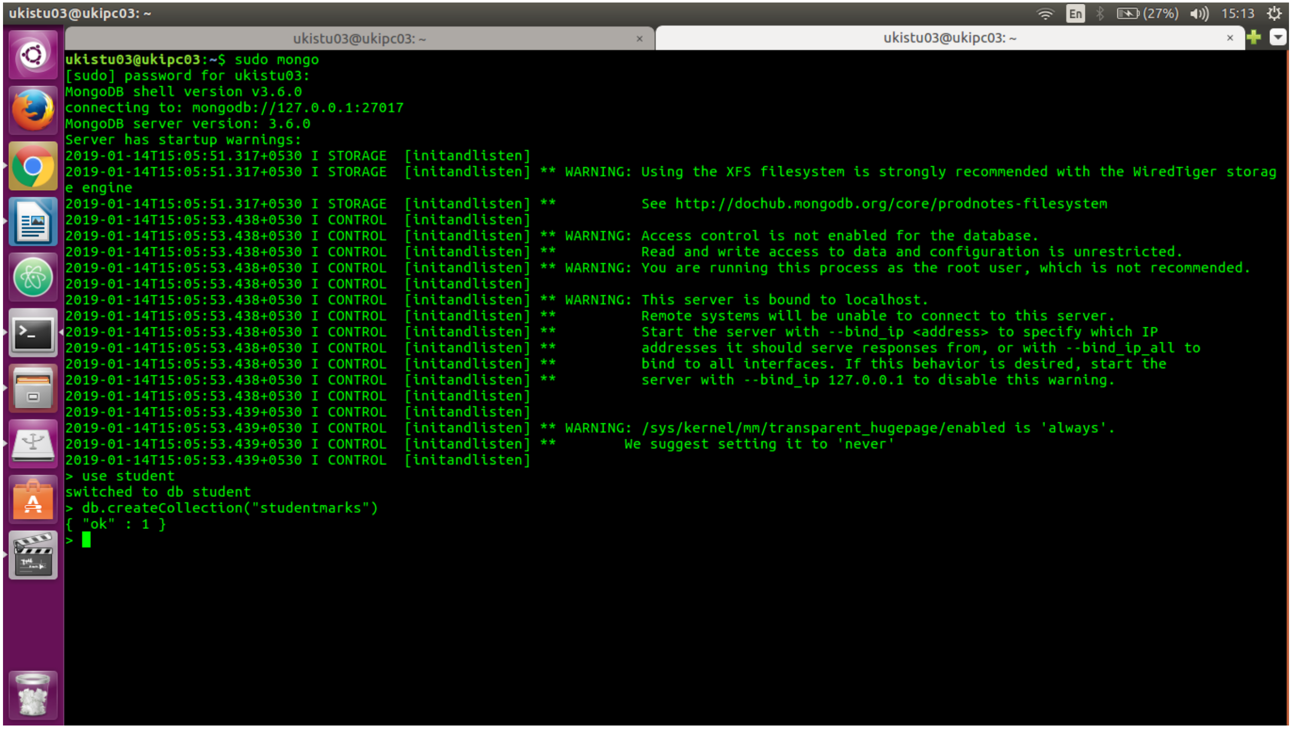


```
ukistu03@ukipc03:~$ sudo mongo
[sudo] password for ukistu03:
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten]
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storag
e engine
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
> use student
switched to db student
>
```

2) Create a collection called studentmarks

> db.createCollection("studentmarks")

{ "ok" : 1 }



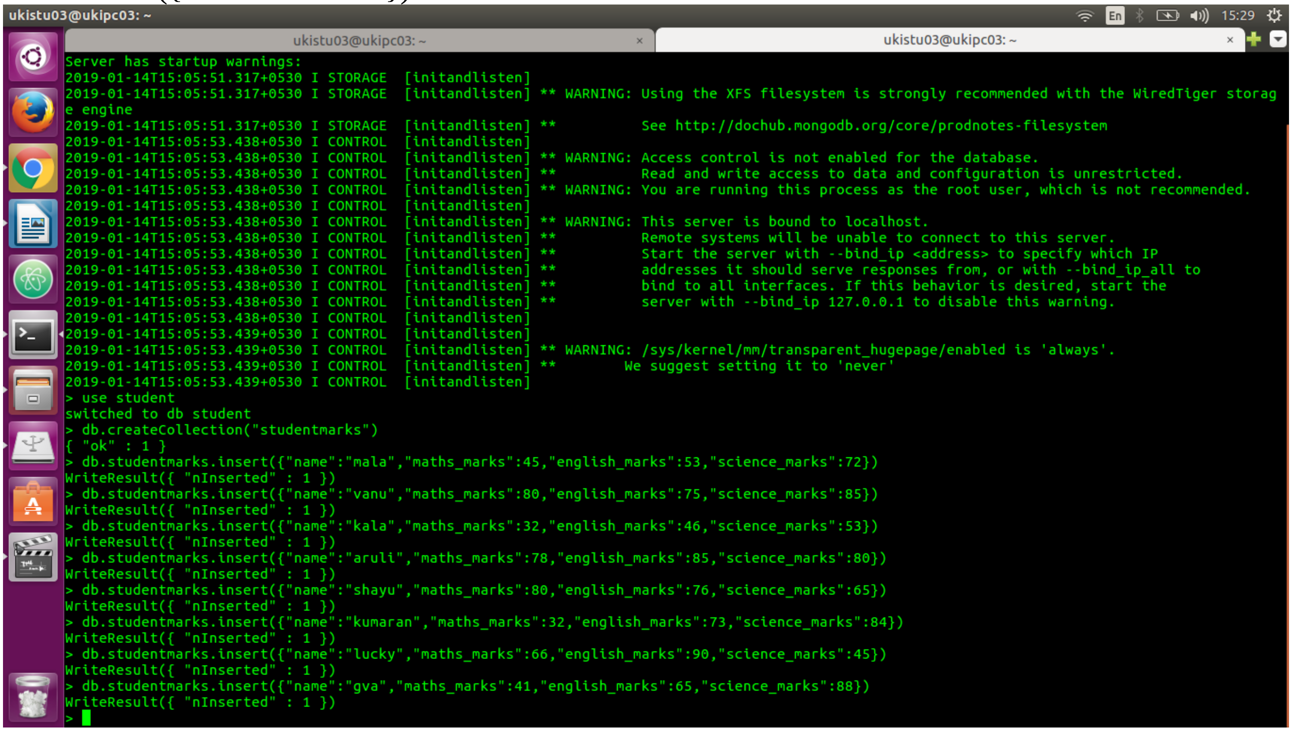
```
ukistu03@ukipc03:~$ sudo mongo
[sudo] password for ukistu03:
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten]
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storag
e engine
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
> use student
switched to db student
> db.createCollection("studentmarks")
{ "ok" : 1 }
>
```

3) Create the documents listed in above table.

>db.studentmarks.insert({"name":"mala","maths_marks":45,"english_marks":53,"science_marks":72})

WriteResult({ "nInserted" : 1 })

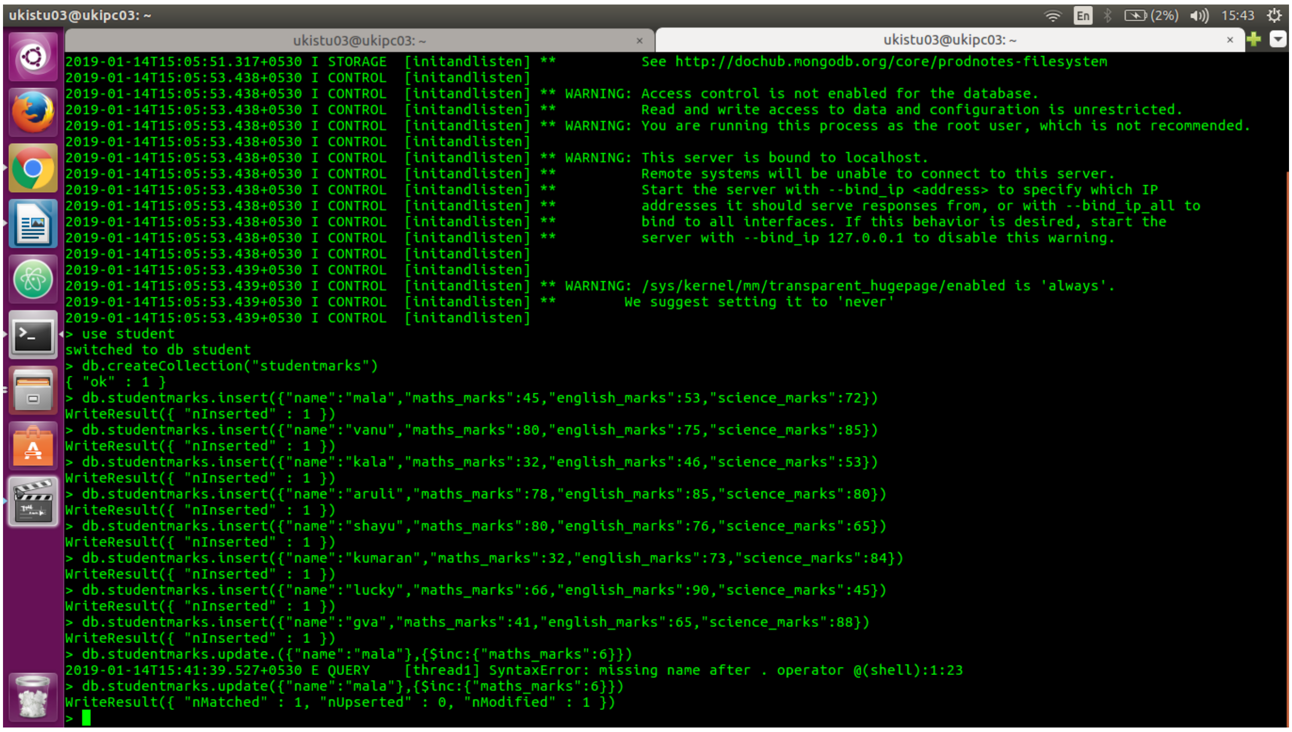
```
>
db.studentmarks.insert({"name":"vanu","maths_marks":80,"english_marks":75,"science_marks":85})
WriteResult({ "nInserted" : 1 })
>
db.studentmarks.insert({"name":"kala","maths_marks":32,"english_marks":46,"science_marks":53})
WriteResult({ "nInserted" : 1 })
>
db.studentmarks.insert({"name":"aruli","maths_marks":78,"english_marks":85,"science_marks":80})
WriteResult({ "nInserted" : 1 })
>
db.studentmarks.insert({"name":"shayu","maths_marks":80,"english_marks":76,"science_marks":65})
WriteResult({ "nInserted" : 1 })
>
db.studentmarks.insert({"name":"kumaran","maths_marks":32,"english_marks":73,"science_marks":84})
WriteResult({ "nInserted" : 1 })
>
db.studentmarks.insert({"name":"lucky","maths_marks":66,"english_marks":90,"science_marks":45})
WriteResult({ "nInserted" : 1 })
>
db.studentmarks.insert({"name":"gva","maths_marks":41,"english_marks":65,"science_marks":88})
WriteResult({ "nInserted" : 1 })
```



```
Server has startup warnings:
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten]
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storag
e engine
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
> use student
switched to db student
> db.createCollection("studentmarks")
{ "ok" : 1 }
> db.studentmarks.insert({"name":"mala","maths_marks":45,"english_marks":53,"science_marks":72})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"vanu","maths_marks":80,"english_marks":75,"science_marks":85})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"kala","maths_marks":32,"english_marks":46,"science_marks":53})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"aruli","maths_marks":78,"english_marks":85,"science_marks":80})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"shayu","maths_marks":80,"english_marks":76,"science_marks":65})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"kumaran","maths_marks":32,"english_marks":73,"science_marks":84})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"lucky","maths_marks":66,"english_marks":90,"science_marks":45})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"gva","maths_marks":41,"english_marks":65,"science_marks":88})
WriteResult({ "nInserted" : 1 })
>
```

4) Increase the maths marks of Mala by 6 marks

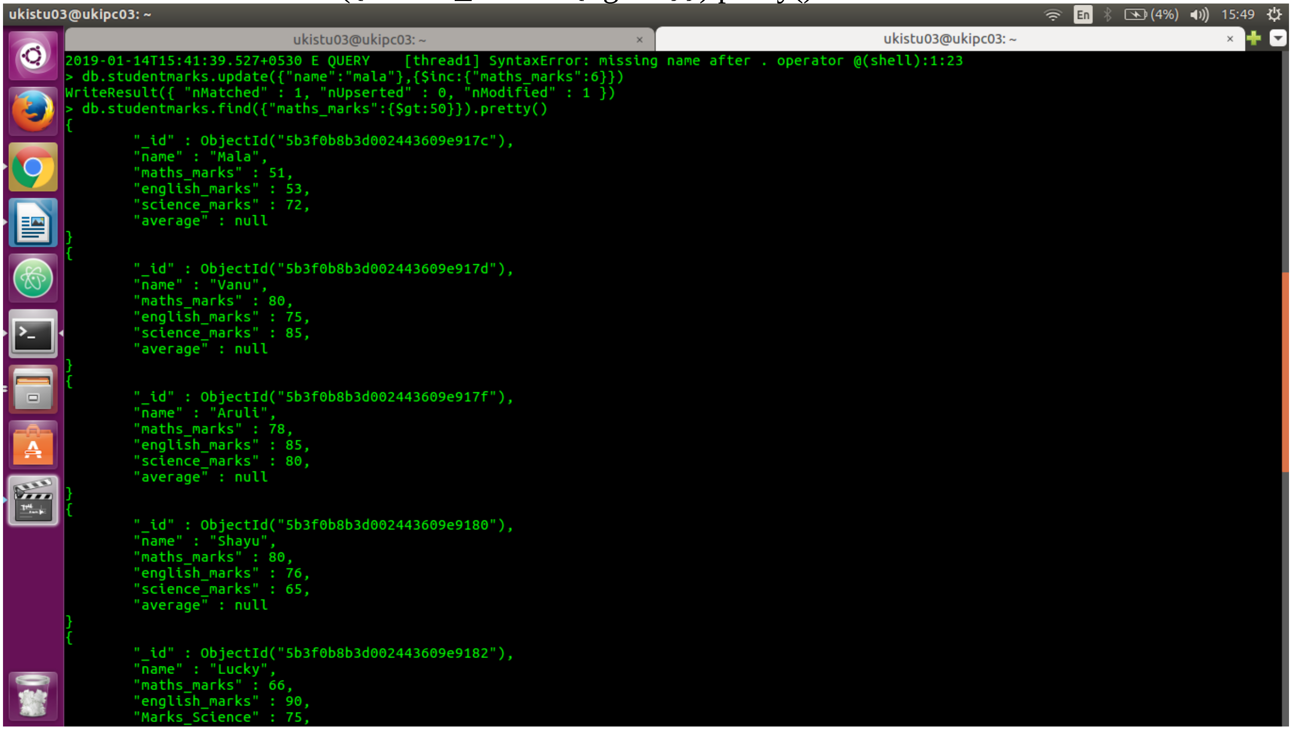
```
> db.studentmarks.update({"name":"mala"},{$inc:{"maths_marks":6}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```



```
2019-01-14T15:05:51.317+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-14T15:05:53.438+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-14T15:05:53.439+0530 I CONTROL  [initandlisten]
> use student
switched to db student
> db.createCollection("studentmarks")
{ "ok" : 1 }
> db.studentmarks.insert({"name":"mala","maths_marks":45,"english_marks":53,"science_marks":72})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"vanu","maths_marks":80,"english_marks":75,"science_marks":85})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"kala","maths_marks":32,"english_marks":46,"science_marks":53})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"aruli","maths_marks":78,"english_marks":85,"science_marks":80})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"shayu","maths_marks":80,"english_marks":76,"science_marks":65})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"kumaran","maths_marks":32,"english_marks":73,"science_marks":84})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"lucky","maths_marks":66,"english_marks":90,"science_marks":45})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.insert({"name":"gva","maths_marks":41,"english_marks":65,"science_marks":88})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.update.({"name":"mala"},{$inc:{"maths_marks":6}})
2019-01-14T15:41:39.527+0530 E QUERY    [thread1] SyntaxError: missing name after . operator @(shell):1:23
> db.studentmarks.update({"name":"mala"},{$inc:{"maths_marks":6}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> 
```

5) List the names of students who got more than 50 marks in Maths Subject.

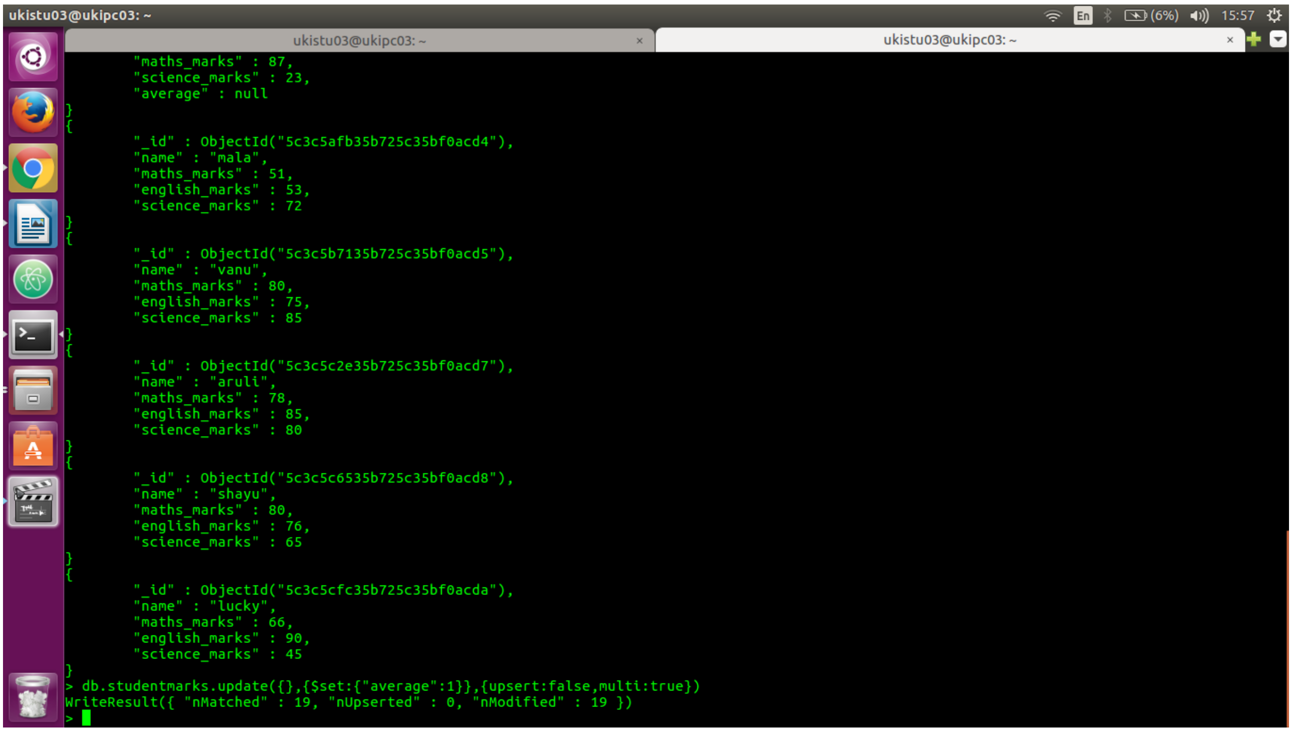
> db.studentmarks.find({"maths_marks":{$gt:50}}).pretty()



```
2019-01-14T15:41:39.527+0530 E QUERY    [thread1] SyntaxError: missing name after . operator @(shell):1:23
> db.studentmarks.update({"name":"mala"},{$inc:{"maths_marks":6}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.studentmarks.find({"maths_marks":{$gt:50}}).pretty()
{
        "_id" : ObjectId("5b3f0b8b3d002443609e917c"),
        "name" : "Mala",
        "maths_marks" : 51,
        "english_marks" : 53,
        "science_marks" : 72,
        "average" : null
}
{
        "_id" : ObjectId("5b3f0b8b3d002443609e917d"),
        "name" : "Vanu",
        "maths_marks" : 80,
        "english_marks" : 75,
        "science_marks" : 85,
        "average" : null
}
{
        "_id" : ObjectId("5b3f0b8b3d002443609e917f"),
        "name" : "Aruli",
        "maths_marks" : 78,
        "english_marks" : 85,
        "science_marks" : 80,
        "average" : null
}
{
        "_id" : ObjectId("5b3f0b8b3d002443609e9180"),
        "name" : "Shayu",
        "maths_marks" : 80,
        "english_marks" : 76,
        "science_marks" : 65,
        "average" : null
}
{
        "_id" : ObjectId("5b3f0b8b3d002443609e9182"),
        "name" : "Lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "Marks_Science" : 75,
```

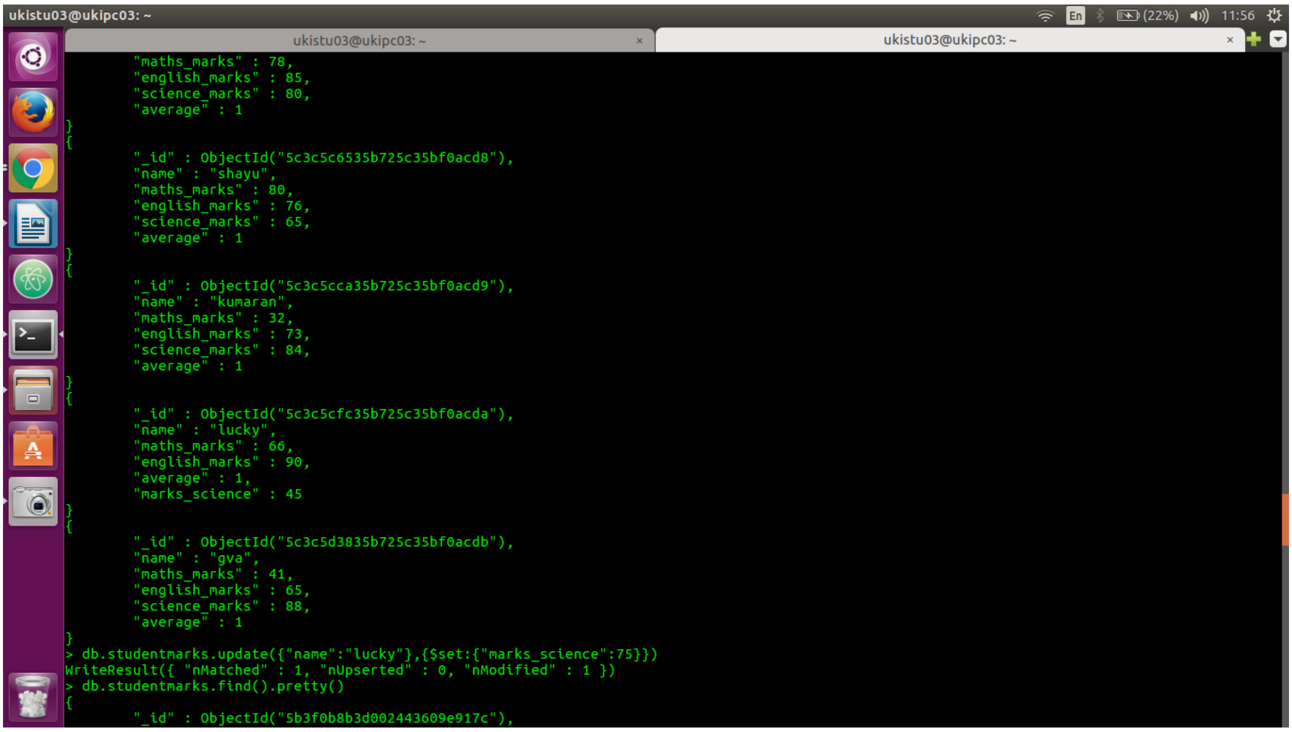
6)Add a new column(field) for Average for all students.

> db.studentmarks.update({},{$set:{"average":1}},{upsert:false,multi:true})



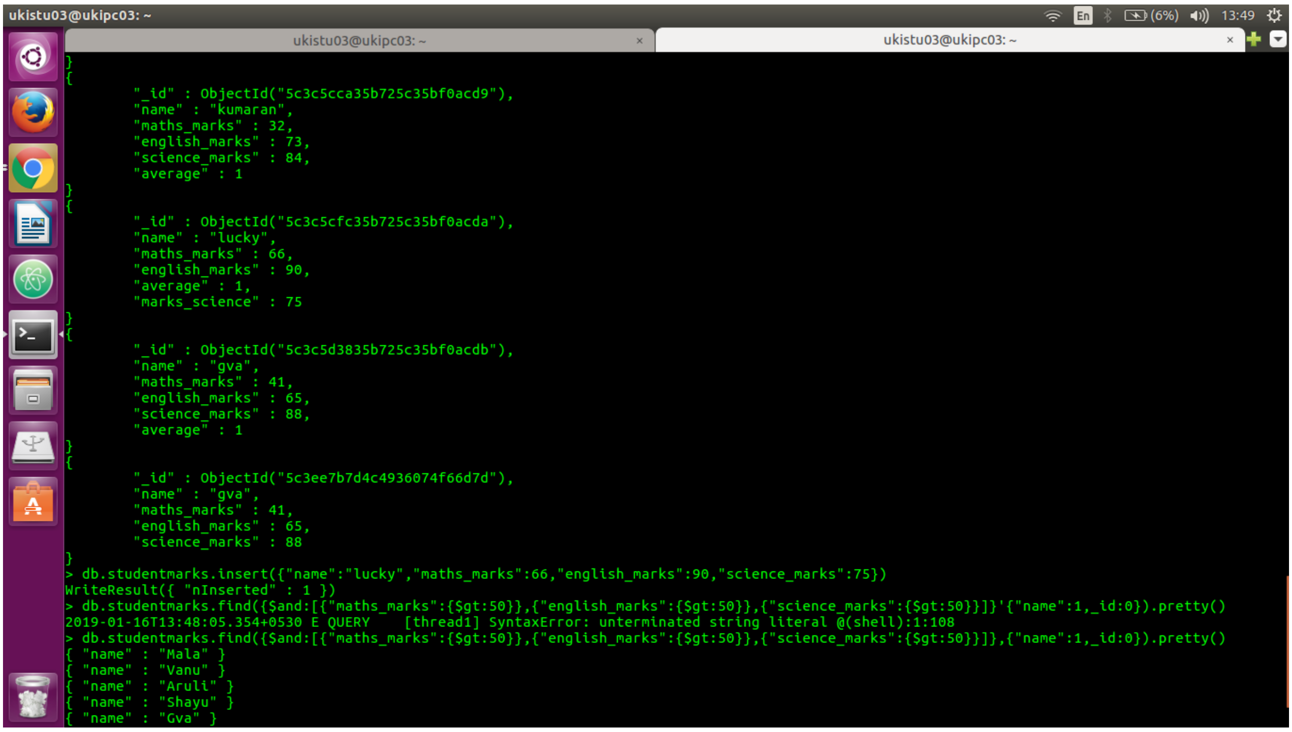
```
        "maths_marks" : 87,
        "science_marks" : 23,
        "average" : null
}
{
        "_id" : ObjectId("5c3c5afb35b725c35bf0acd4"),
        "name" : "mala",
        "maths_marks" : 51,
        "english_marks" : 53,
        "science_marks" : 72
}
{
        "_id" : ObjectId("5c3c5b7135b725c35bf0acd5"),
        "name" : "vanu",
        "maths_marks" : 80,
        "english_marks" : 75,
        "science_marks" : 85
}
{
        "_id" : ObjectId("5c3c5c2e35b725c35bf0acd7"),
        "name" : "aruli",
        "maths_marks" : 78,
        "english_marks" : 85,
        "science_marks" : 80
}
{
        "_id" : ObjectId("5c3c5c6535b725c35bf0acd8"),
        "name" : "shayu",
        "maths_marks" : 80,
        "english_marks" : 76,
        "science_marks" : 65
}
{
        "_id" : ObjectId("5c3c5cfc35b725c35bf0acda"),
        "name" : "lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "science_marks" : 45
}
> db.studentmarks.update({},{$set:{"average":1}},{upsert:false,multi:true})
WriteResult({ "nMatched" : 19, "nUpserted" : 0, "nModified" : 19 })
> 
```

7) Update Marks_Science=75 to Lucky .
> db.studentmarks.update({"name":"lucky"},{$set:{"marks_science":75}})



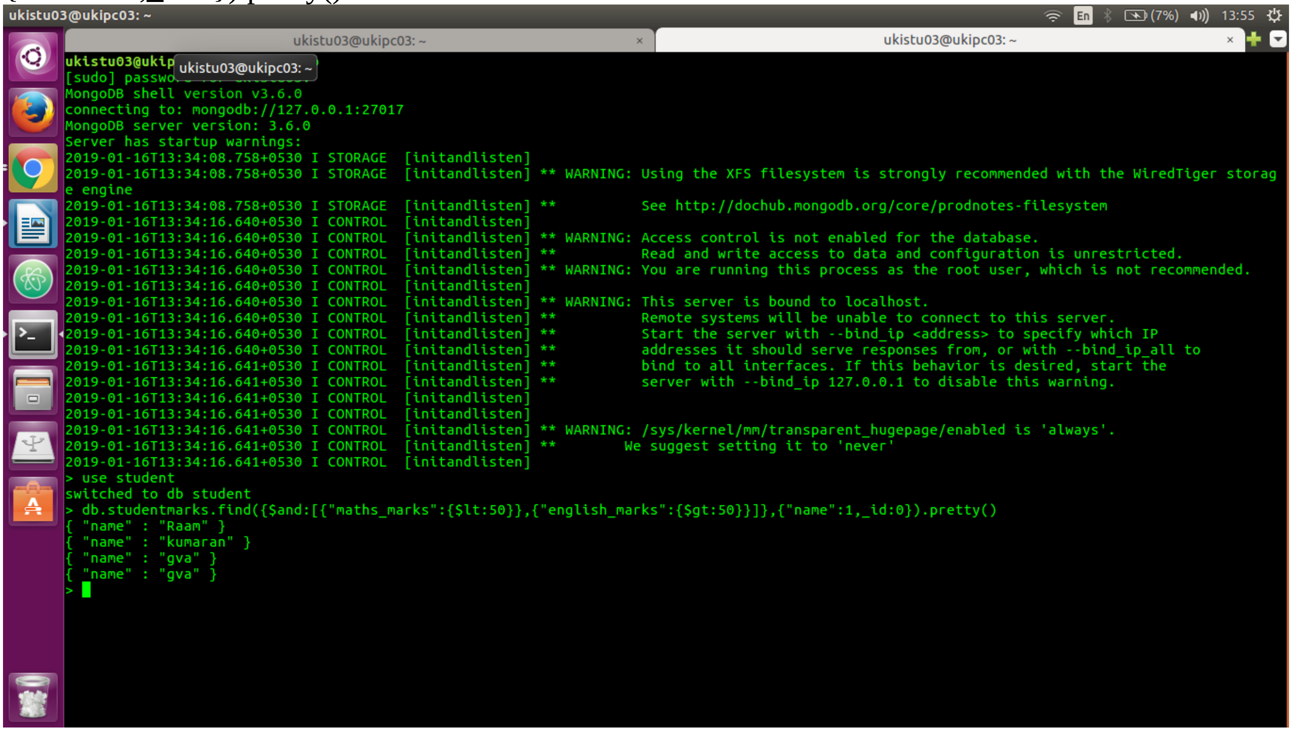
```
        "maths_marks" : 78,
        "english_marks" : 85,
        "science_marks" : 80,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5c6535b725c35bf0acd8"),
        "name" : "shayu",
        "maths_marks" : 80,
        "english_marks" : 76,
        "science_marks" : 65,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cca35b725c35bf0acd9"),
        "name" : "kumaran",
        "maths_marks" : 32,
        "english_marks" : 73,
        "science_marks" : 84,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cfc35b725c35bf0acda"),
        "name" : "lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "average" : 1,
        "marks_science" : 45
}
{
        "_id" : ObjectId("5c3c5d3835b725c35bf0acdb"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88,
        "average" : 1
}
> db.studentmarks.update({"name":"lucky"},{$set:{"marks_science":75}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.studentmarks.find().pretty()
{
        "_id" : ObjectId("5b3f0b8b3d002443609e917c"),
```

8) List the names who got more than 50 marks in all subjects.
>db.studentmarks.find({$and:[{"maths_marks":{$gt:50}},{"english_marks":{$gt:50}},
{"science_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()



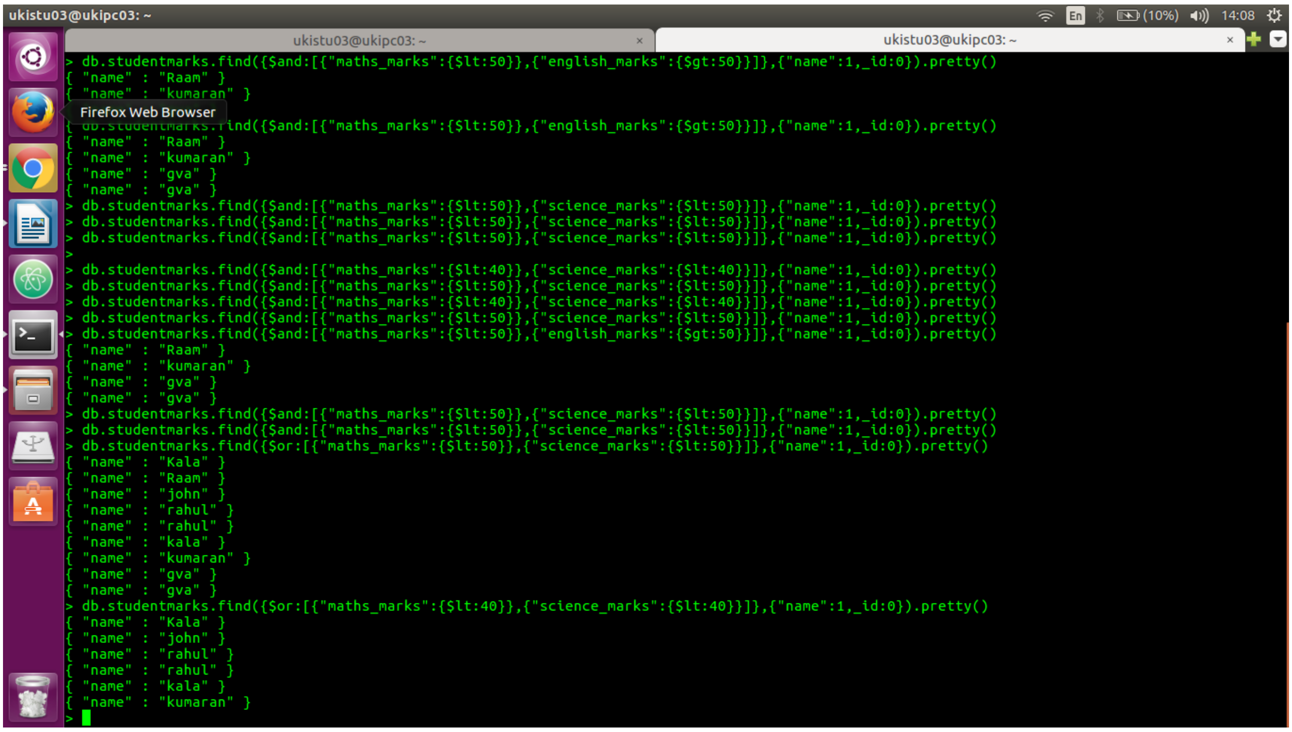
```
}
{
        "_id" : ObjectId("5c3c5cca35b725c35bf0acd9"),
        "name" : "kumaran",
        "maths_marks" : 32,
        "english_marks" : 73,
        "science_marks" : 84,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cfc35b725c35bf0acda"),
        "name" : "lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "average" : 1,
        "marks_science" : 75
}
{
        "_id" : ObjectId("5c3c5d3835b725c35bf0acdb"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88,
        "average" : 1
}
{
        "_id" : ObjectId("5c3ee7b7d4c4936074f66d7d"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88
}
> db.studentmarks.insert({"name":"lucky","maths_marks":66,"english_marks":90,"science_marks":75})
WriteResult({ "nInserted" : 1 })
> db.studentmarks.find({$and:[{"maths_marks":{$gt:50}},{"english_marks":{$gt:50}},{"science_marks":{$gt:50}}]}'{"name":1,_id:0}).pretty()
2019-01-16T13:48:05.354+0530 E QUERY    [thread1] SyntaxError: unterminated string literal @(shell):1:108
> db.studentmarks.find({$and:[{"maths_marks":{$gt:50}},{"english_marks":{$gt:50}},{"science_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Mala" }
{ "name" : "Vanu" }
{ "name" : "Aruli" }
{ "name" : "Shayu" }
{ "name" : "Gva" }
```

9) List the names who got less than 50 marks in Maths subject and more than 50 marks in English
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"english_marks":{$gt:50}}]},
{"name":1,_id:0}).pretty()



```
ukistu03@ukip
[sudo] passwo
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2019-01-16T13:34:08.758+0530 I STORAGE  [initandlisten]
2019-01-16T13:34:08.758+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storag
e engine
2019-01-16T13:34:08.758+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten]
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten]
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2019-01-16T13:34:16.640+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2019-01-16T13:34:16.641+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2019-01-16T13:34:16.641+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2019-01-16T13:34:16.641+0530 I CONTROL  [initandlisten]
2019-01-16T13:34:16.641+0530 I CONTROL  [initandlisten]
2019-01-16T13:34:16.641+0530 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-01-16T13:34:16.641+0530 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2019-01-16T13:34:16.641+0530 I CONTROL  [initandlisten]
> use student
switched to db student
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"english_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Raam" }
{ "name" : "kumaran" }
{ "name" : "gva" }
{ "name" : "gva" }
> 
```

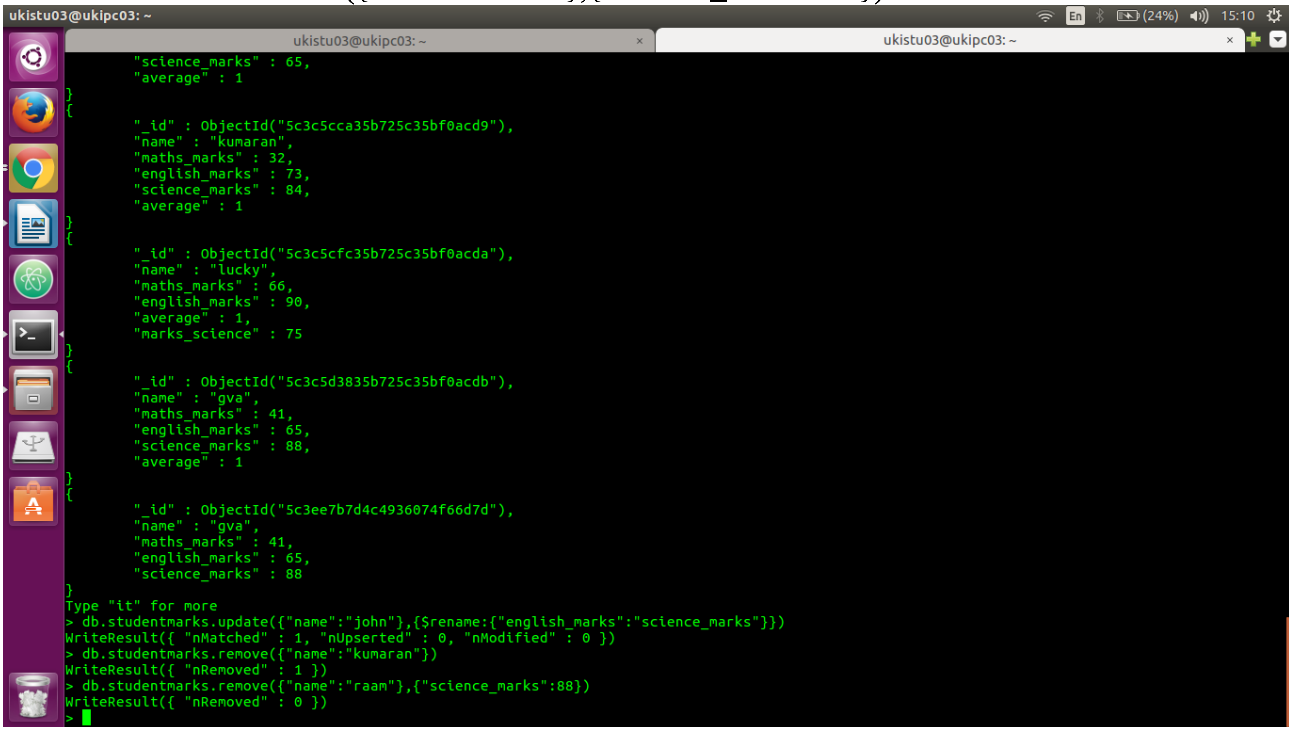
10) List the names who got less than 40 in both Maths and Science.
> db.studentmarks.find({$or:[{"maths_marks":{$lt:40}},{"science_marks":{$lt:40}}]},
{"name":1,_id:0}).pretty()



```
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"english_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Raam" }
{ "name" : "kumaran" }

> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"english_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Raam" }
{ "name" : "kumaran" }
{ "name" : "gva" }
{ "name" : "gva" }
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
>
> db.studentmarks.find({$and:[{"maths_marks":{$lt:40}},{"science_marks":{$lt:40}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:40}},{"science_marks":{$lt:40}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"english_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Raam" }
{ "name" : "kumaran" }
{ "name" : "gva" }
{ "name" : "gva" }
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$or:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Kala" }
{ "name" : "Raam" }
{ "name" : "john" }
{ "name" : "rahul" }
{ "name" : "rahul" }
{ "name" : "kala" }
{ "name" : "kumaran" }
{ "name" : "gva" }
{ "name" : "gva" }
> db.studentmarks.find({$or:[{"maths_marks":{$lt:40}},{"science_marks":{$lt:40}}]},{"name":1,_id:0}).pretty()
{ "name" : "Kala" }
{ "name" : "john" }
{ "name" : "rahul" }
{ "name" : "rahul" }
{ "name" : "kala" }
{ "name" : "kumaran" }
>
```

11) Remove Science column/field for Raam

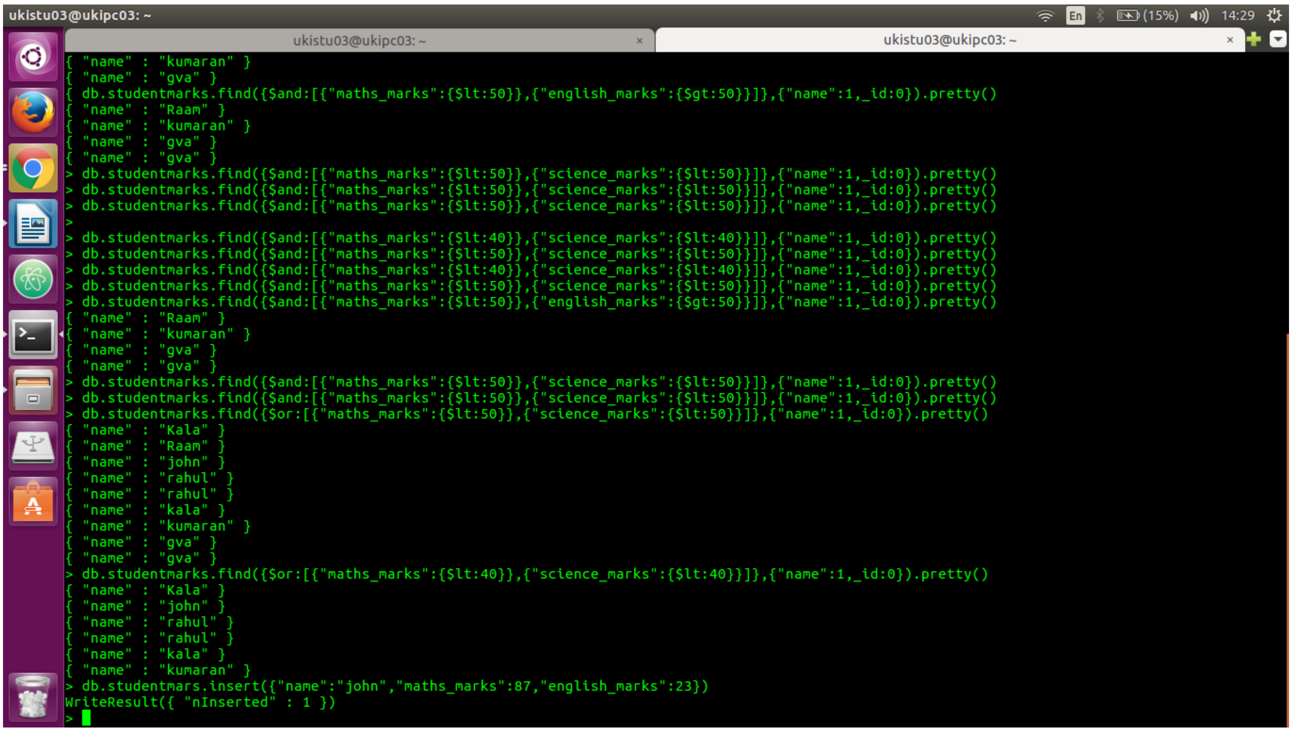
> db.studentmarks.remove({"name":"raam"},{"science_marks":88})



```
        "science_marks" : 65,
        "average" : 1
}
{

        "_id" : ObjectId("5c3c5cca35b725c35bf0acd9"),
        "name" : "kumaran",
        "maths_marks" : 32,
        "english_marks" : 73,
        "science_marks" : 84,
        "average" : 1
}
{

        "_id" : ObjectId("5c3c5cfc35b725c35bf0acda"),
        "name" : "lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "average" : 1,
        "marks_science" : 75
}
{

        "_id" : ObjectId("5c3c5d3835b725c35bf0acdb"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88,
        "average" : 1
}
{

        "_id" : ObjectId("5c3ee7b7d4c4936074f66d7d"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88
}
Type "it" for more
> db.studentmarks.update({"name":"john"},{$rename:{"english_marks":"science_marks"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.studentmarks.remove({"name":"kumaran"})
WriteResult({ "nRemoved" : 1 })
> db.studentmarks.remove({"name":"raam"},{"science_marks":88})
WriteResult({ "nRemoved" : 0 })
>
```

12) Update John’s Math mark as 87 and English mark as 23, if john not available upsert.

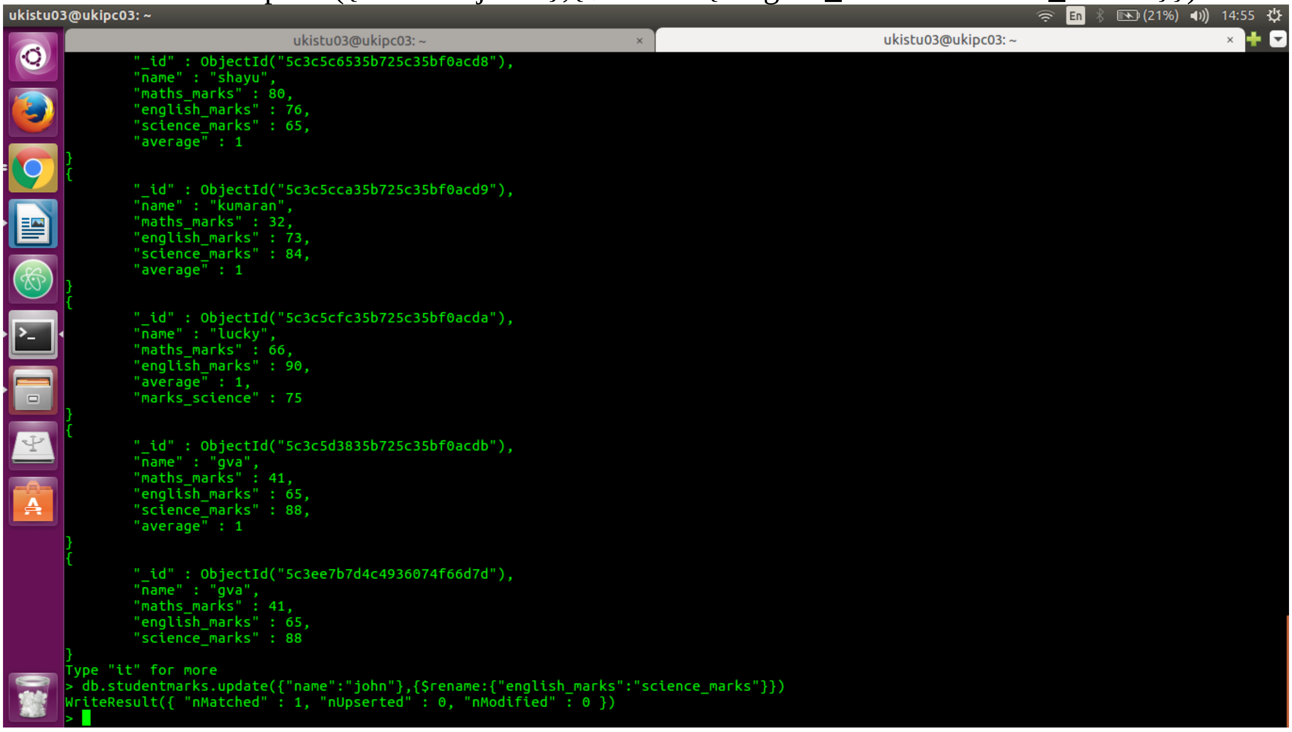
> db.studentmars.insert({"name":"john","maths_marks":87,"english_marks":23})



```
{ "name" : "kumaran" }
{ "name" : "gva" }
{ db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"english_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Raam" }
{ "name" : "kumaran" }
{ "name" : "gva" }
{ "name" : "gva" }
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
>
> db.studentmarks.find({$and:[{"maths_marks":{$lt:40}},{"science_marks":{$lt:40}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:40}},{"science_marks":{$lt:40}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"english_marks":{$gt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Raam" }
{ "name" : "kumaran" }
{ "name" : "gva" }
{ "name" : "gva" }
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$and:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
> db.studentmarks.find({$or:[{"maths_marks":{$lt:50}},{"science_marks":{$lt:50}}]},{"name":1,_id:0}).pretty()
{ "name" : "Kala" }
{ "name" : "Raam" }
{ "name" : "john" }
{ "name" : "rahul" }
{ "name" : "rahul" }
{ "name" : "kala" }
{ "name" : "kumaran" }
{ "name" : "gva" }
{ "name" : "gva" }
> db.studentmarks.find({$or:[{"maths_marks":{$lt:40}},{"science_marks":{$lt:40}}]},{"name":1,_id:0}).pretty()
{ "name" : "Kala" }
{ "name" : "john" }
{ "name" : "rahul" }
{ "name" : "rahul" }
{ "name" : "kala" }
{ "name" : "kumaran" }
> db.studentmars.insert({"name":"john","maths_marks":87,"english_marks":23})
WriteResult({ "nInserted" : 1 })
> 
```

13) Rename the english_marks column/field for John to science_marks

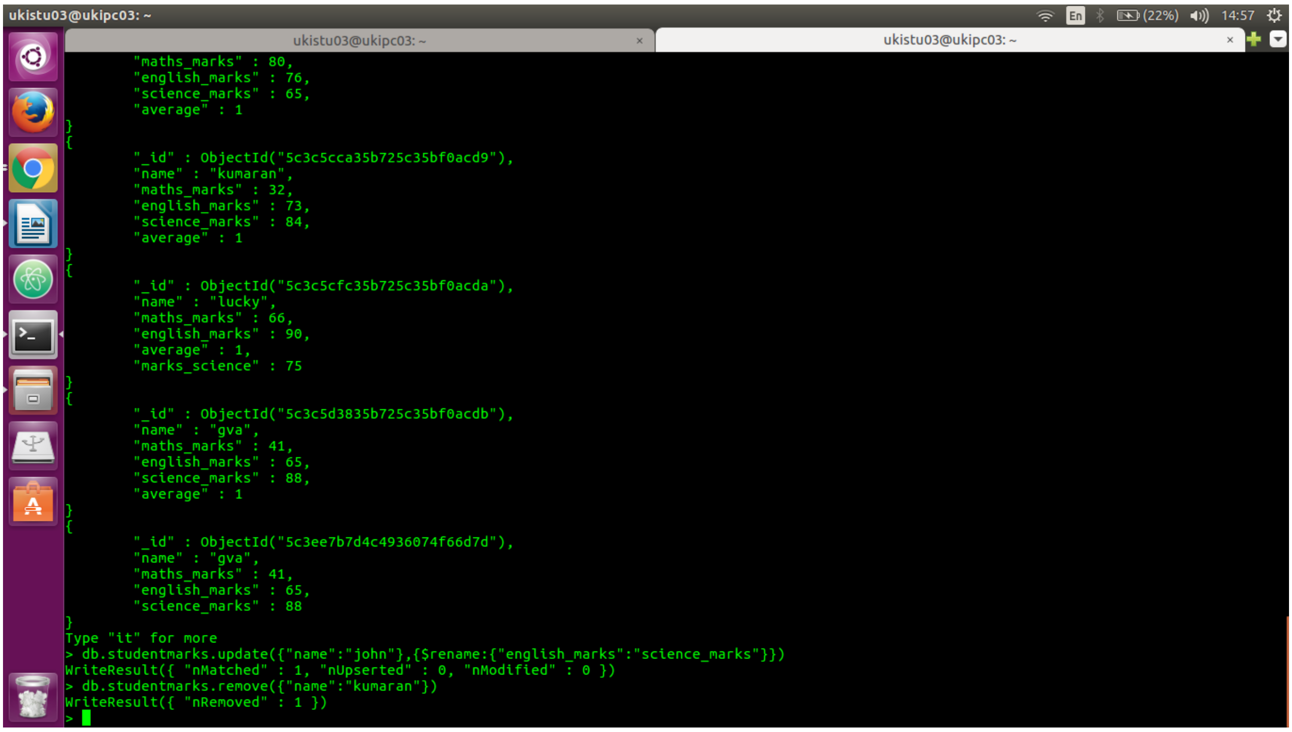
> db.studentmarks.update({"name":"john"},{$rename:{"english_marks":"science_marks"}})



```
        "_id" : ObjectId("5c3c5c6535b725c35bf0acd8"),
        "name" : "shayu",
        "maths_marks" : 80,
        "english_marks" : 76,
        "science_marks" : 65,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cca35b725c35bf0acd9"),
        "name" : "kumaran",
        "maths_marks" : 32,
        "english_marks" : 73,
        "science_marks" : 84,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cfc35b725c35bf0acda"),
        "name" : "lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "average" : 1,
        "marks_science" : 75
}
{
        "_id" : ObjectId("5c3c5d3835b725c35bf0acdb"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88,
        "average" : 1
}
{
        "_id" : ObjectId("5c3ee7b7d4c4936074f66d7d"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88
}
Type "it" for more
> db.studentmarks.update({"name":"john"},{$rename:{"english_marks":"science_marks"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> 
```

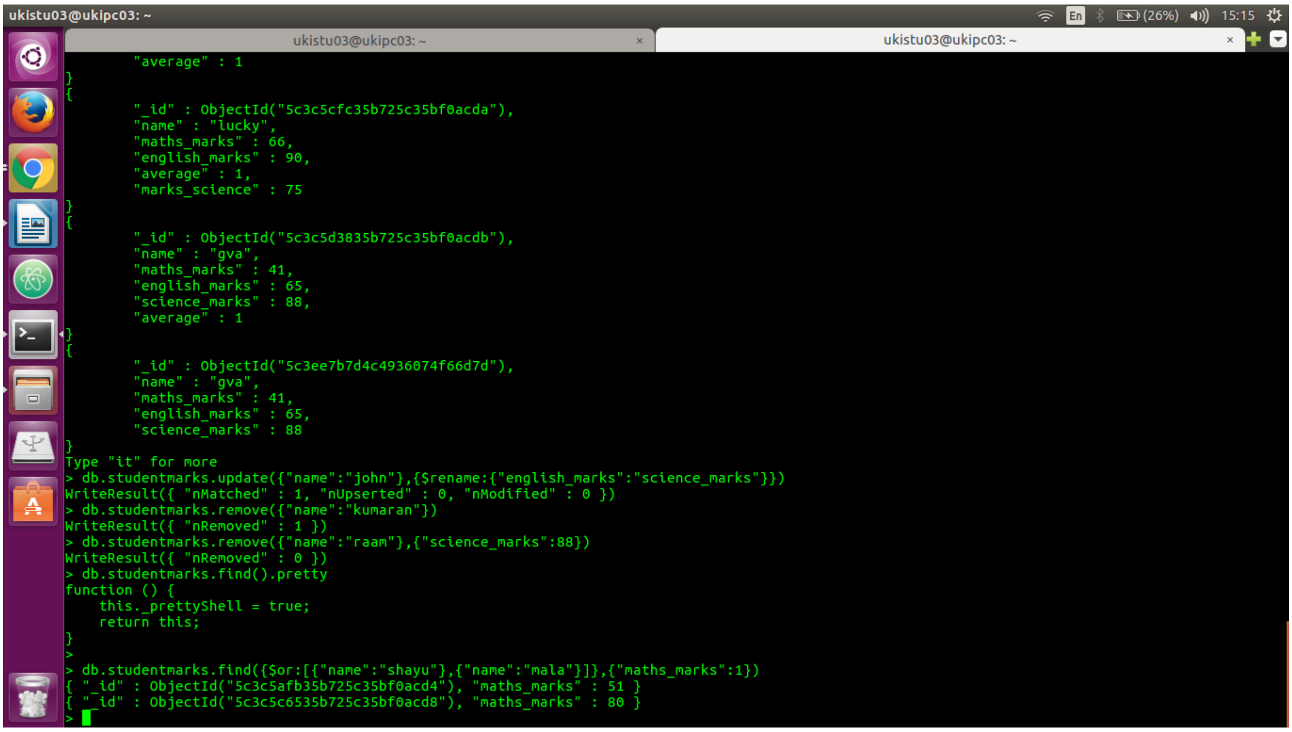
14) Remove Kumaran’s document from collection

> db.studentmarks.remove({"name":"kumaran"})



```
        "maths_marks" : 80,
        "english_marks" : 76,
        "science_marks" : 65,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cca35b725c35bf0acd9"),
        "name" : "kumaran",
        "maths_marks" : 32,
        "english_marks" : 73,
        "science_marks" : 84,
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cfc35b725c35bf0acda"),
        "name" : "lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "average" : 1,
        "marks_science" : 75
}
{
        "_id" : ObjectId("5c3c5d3835b725c35bf0acdb"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88,
        "average" : 1
}
{
        "_id" : ObjectId("5c3ee7b7d4c4936074f66d7d"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88
}
Type "it" for more
> db.studentmarks.update({"name":"john"},{$rename:{"english_marks":"science_marks"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.studentmarks.remove({"name":"kumaran"})
WriteResult({ "nRemoved" : 1 })
> 
```

15) Find Kala’s or Aruli’s math_marks and science_marks
> db.studentmarks.find({$or:[{"name":"shayu"},{"name":"mala"}]},{"maths_marks":1})



```
        "average" : 1
}
{
        "_id" : ObjectId("5c3c5cfc35b725c35bf0acda"),
        "name" : "lucky",
        "maths_marks" : 66,
        "english_marks" : 90,
        "average" : 1,
        "marks_science" : 75
}
{
        "_id" : ObjectId("5c3c5d3835b725c35bf0acdb"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88,
        "average" : 1
}
{
        "_id" : ObjectId("5c3ee7b7d4c4936074f66d7d"),
        "name" : "gva",
        "maths_marks" : 41,
        "english_marks" : 65,
        "science_marks" : 88
}
Type "it" for more
> db.studentmarks.update({"name":"john"},{$rename:{"english_marks":"science_marks"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.studentmarks.remove({"name":"kumaran"})
WriteResult({ "nRemoved" : 1 })
> db.studentmarks.remove({"name":"raam"},{"science_marks":88})
WriteResult({ "nRemoved" : 0 })
> db.studentmarks.find().pretty
function () {
    this._prettyShell = true;
    return this;
}
>
> db.studentmarks.find({$or:[{"name":"shayu"},{"name":"mala"}]},{"maths_marks":1})
{ "_id" : ObjectId("5c3c5afb35b725c35bf0acd4"), "maths_marks" : 51 }
{ "_id" : ObjectId("5c3c5c6535b725c35bf0acd8"), "maths_marks" : 80 }
> 
```