1. Create a database “Student” with the following attributes Rollno, Age, ContactNo, Email-Id.
2. Insert appropriate values

```
>_MongoSH Beta
<
> use mydb
< 'switched to db mydb'
> db.createCollection("student")
< { ok: 1 }
> db.student.insert({id:1,name:"someone",roll:12,age:20,phone:8997638492,mail:"someone@example.com"});
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 1f b0 4a bb cb 2a de 45 d2 a7> } } }
> db.student.insert({id:2,name:"rocky",roll:13,age:20,phone:8999388492,mail:"rocky@plateau.com"});
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 1f e8 4a bb cb 2a de 45 d2 a8> } } }
> db.student.insert({id:3,name:"bob",roll:14,age:20,phone:8999193492,mail:"bob@apple.com"});
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 20 11 4a bb cb 2a de 45 d2 a9> } } }
> db.student.insert({id:4,name:"rob",roll:15,age:20,phone:4995194492,mail:"rob@banana.com"});
< { acknowledged: 1,
    insertedIds:
```

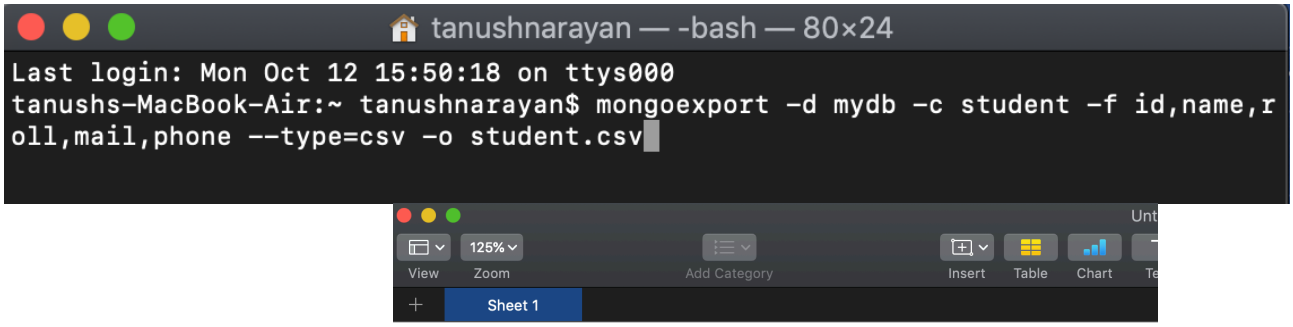
3. Write query to update Email-Id of a student with rollno 10.

```
> db.student.insert({id:4,roll:10,name:"rob",age:21,phone:73348912636,mail:"rob@robbery.com"})
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 28 4f 70 00 39 3d fe 19 3d 63> } } }
> db.student.update({roll:10},{$set:{mail:"rob@banana.com"}})
< { acknowledged: 1,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0 }
> db.student.find({roll:10})
< { _id: ObjectID("5f84284f7000393dfe193d63"),
    id: 4,
    roll: 10,
    name: 'rob',
    age: 21,
    phone: 73348912636,
    mail: 'rob@banana.com' }
>
```

4. Replace the student name from “ABC” to “FEM” of rollno 11.

```
> db.student.update({name:"rob"},{$set:{name:"bob"}})
< { acknowledged: 1,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0 }
> db.student.find({roll:10})
< { _id: ObjectID("5f84284f7000393dfe193d63"),
    id: 4,
    roll: 10,
    name: 'bob',
    age: 21,
    phone: 73348912636,
    mail: 'rob@banana.com' }
>
```

5. Export the created table into local file system



Untitled

_id	id	roll	name	age	phone	mail
5f84284f7000393dfe193d63	4	10	bob	21	73348912636	rob@banana.com

6. Drop table

```
> db.student.drop()
< true
> db.student.find({})
<
>
```

2 Perform the following DB operations using MongoDB.

1. Create a collection by name Customers with the following attributes.
Cust_id, Acc_Bal, Acc_Type

```
> use mydb
< 'switched to db mydb'
> db.createCollection("customers")
< { ok: 1 }
>
```

2. Insert at least 5 values into the table

```
> db.customers.insert({id:1,"cust_id":1,"acc_bal":20000,"acc_type":"fd"})
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 35 9f 70 00 39 3d fe 19 3d 64> } } }
> db.customers.insert({id:2,"cust_id":2,"acc_bal":150000,"acc_type":"saving"})
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 35 c4 70 00 39 3d fe 19 3d 65> } } }
> db.customers.insert({id:3,"cust_id":3,"acc_bal":3000,"acc_type":"saving"})
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 35 e7 70 00 39 3d fe 19 3d 66> } } }
> db.customers.insert({id:4,"cust_id":4,"acc_bal":34000,"acc_type":"fd"})
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 35 fd 70 00 39 3d fe 19 3d 67> } } }
> db.customers.insert({id:5,"cust_id":5,"acc_bal":7000,"acc_type":"saving"})
< { acknowledged: 1,
    insertedIds:
     { '0':
        { _bsontype: 'ObjectID',
          id: <Buffer 5f 84 36 82 70 00 39 3d fe 19 3d 68> } } }
```

3. Write a query to display those records whose total account balance is greater than
1200 of account type ‘Z’ for each customer_id.

```
> db.customers.find({"acc_bal":{$gte:7000},"acc_type":"saving"})
< { _id: ObjectID("5f8435c47000393dfe193d65"),
    id: 2,
    cust_id: 2,
    acc_bal: 150000,
    acc_type: 'saving' }
  { _id: ObjectID("5f8436827000393dfe193d68"),
    id: 5,
    cust_id: 5,
    acc_bal: 7000,
    acc_type: 'saving' }
```

4. Determine Minimum and Maximum account balance for each customer_id.

```
> db.customers.aggregate([{$group:{"_id":"$custid","max_bal":{$max:"$acc_bal"},"min_bal":{$min:"$acc_bal"}}}])
< [ { _id: null, max_bal: 150000, min_bal: 3000 } ]
```

5. Export the created collection into local file system

customers

_id	id	cust_id	acc_bal	acc_type
5f84359f7000393dfe193d64	1	1	20000	fd
5f8435c47000393dfe193d65	2	2	150000	saving
5f8435e77000393dfe193d66	3	3	3000	saving
5f8435fd7000393dfe193d67	4	4	34000	fd
5f8436827000393dfe193d68	5	5	7000	saving

```
Last login: Mon Oct 12 15:50:18 on ttys000
tanushs-MacBook-Air:~ tanushnarayan$ mongoexport -d mydb -c customers -f cust_id
,acc_bal,acc_type --type=csv -o customer.csv
```