```
> db.sales.aggregate([{$group:{_id:null, maxpara:{$max:'$quantity'}}},{$project:{_id:0}}])
< {
    maxpara: 30
  }
> db.sales.aggregate([{$group:{_id:'item', maxpara:{$max:'$quantity'}}},])
< {
    _id: 'item',
    maxpara: 30
  }
> db.sales.aggregate([{$group:{_id:'$item', maxpara:{$max:'$quantity'}}}])
< {
    _id: 'Cappuccino',
    maxpara: 20
  }
  {
    _id: 'Lattes',
    maxpara: 30
  }
  {
    _id: 'Mochas',
    maxpara: 11
  }
  {
    _id: 'Americanos',
    maxpara: 22
  }
```

```
> db.sales.aggregate([{$group:{_id:'$item',maxpara:{$max:{$multiply:['$price','$quantity']}}}}
                     ,{$project:{_id:1, maxpara:1}}])
< {
    _id: 'Cappuccino',
    maxpara: 170
  }
  {
    _id: 'Mochas',
    maxpara: 275
  }
  {
    _id: 'Lattes',
    maxpara: 750
  }
  {
    _id: 'Americanos',
    maxpara: 210
  }
```

```
> db.sales.aggregate([{$group:{_id:'$item',maxpara:{$max:{$multiply:['$price','$quantity']}}}}
                    ,{$project:{_id:1, maxpara:1}},{$sort:{maxpara:-1}},{$limit:1}])
< {
    _id: 'Lattes',
    maxpara: 750
```

```
> db.sales.aggregate([
    {
      $addFields: {
        totalSale: { $multiply: ["$price", "$quantity"] }
      }
    },
    {
      $sort: { item: 1, totalSale: -1 }
    },
    {
      $group: {
        _id: "$item",
        salesList: { $push: "$totalSale" }
      }
    },
    {
      $project: {
        secondMaxSale: { $arrayElemAt: ["$salesList", 1] }
      }
    }
  ])
< {
    _id: 'Cappuccino',
    secondMaxSale: 140
  }
  {
    _id: 'Lattes',
    secondMaxSale: 375
```

```
> db.sales.getIndexes()
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]
> db.sales.createIndex({item:1})
< item_1
> db.sales.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { item: 1 }, name: 'item_1' }
  ]
```

```
> db.createCollection("users")
< { ok: 1 }
> db.users.insertMany([
      { email:  "john@test.com", name: "john"},
      { email:  "jane@test.com", name: "jane"},
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('6841a1779e1e02d8e429238f'),
      '1': ObjectId('6841a1779e1e02d8e4292390')
    }
  }
> db.users.createIndex({email:1},{unique:true})
< email_1
```

```
> db.users.insertMany([
    { email:  "john@test.com", name: "john"},
    { email:  "jane@test.com", name: "jane"},
   {email:"john@test.com", name: "johny"}
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('6841a2489e1e02d8e4292391'),
      '1': ObjectId('6841a2489e1e02d8e4292392'),
      '2': ObjectId('6841a2489e1e02d8e4292393')
    }
  }
> db.users.createIndex({email:1},{unique:1})
✖ ▸ MongoServerError[DuplicateKey]: Index build failed: b33f0036-4b30-470c-9e11-be374646
```

```
> db.users.deleteOne({name:"johny"})
< {
    acknowledged: true,
    deletedCount: 1
  }
> db.users.createIndex({email:1},{unique:1})
< email_1
```

```
> db.locations.insertOne({address:"Downtown San Jose, CA, USA", lat: 37.335480, long:-121.89302
< {
    acknowledged: true,
    insertedId: ObjectId('6841a41b9e1e02d8e4292394')
  }
> db.locations.createIndex({lat:1, long:1},{uniques:true})
< lat_1_long_1
```

```
package connection;

import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;
import com.mongodb.client.MongoDatabase;

public class mongoDB {

        public static void main(String[] args) {
                try {
        MongoClient db
          = new MongoClient("localhost", 27017);

        MongoCredential credential;
        credential
          = MongoCredential
              .createCredential(
                "GFGUser", "mongoDb",
                "password".toCharArray());
```

```
            System.out.println(
               "Successfully Connected"
               + " to the database");

            MongoDatabase database
               = db.getDatabase("mongoDb");
            System.out.println("Credentials are: "
                        + credential);
         }
         catch (Exception e) {
            System.out.println(
               "Connection establishment failed");
            System.out.println(e);
         }

            }

}
```