2. **Aggregation and indexing with suitable example using MongoDB.**
   Create an orders collection with keys order_id, cust_id, cust_name, phone_no(array field), email_id(optional field), item_name, DtOfOrder, quantity, amount, status(P :pending / D:delivered)

i. Create a simple index on cust_id and also create a simple index on Item_name. Try to make a duplicate entry.
   - ➢ db.orders.createIndex({Cust_id:1})
   - ➢ db.orders.createIndex({Item_name:1})
   - ➢ db.orders.getIndexes()

ii. Create unique index on the order_id key and try to make duplicate entry.
   - ➢ db.orders.createIndex({Order_id:1}, {unique:true})

iii. Create a multikey index on phone_no and find the customers with 2 phone nos.
   - ➢ db.orders.createIndex({Phone_no:1})
   - ➢ db.orders.find({Phone_no:{$size:2}}).pretty()

iv. Create a sparse index on email_id key and show the effects with and without indexing. (Hint:use find() before and after aplying index. Also use .explain())
   - ➢ db.orders.find({Email_id:"aryan@gmail.com"}).explain()
   - ➢ db.orders.createIndex({Email_id:1},{sparse:true})
   - ➢ db.orders.find({Email_id:"aryan@gmail.com"}).explain()

v. Display all indexes created on order collection and Also show the size of indexes.
   - ➢ db.orders.getIndexes()
   - ➢ db.orders.totalIndexSize()

vi. Delete all indexes.
   - ➢ db.orders.dropIndexes()

vii. A) Find Total no of orders received so far
   db.orders.find({Status:'D'}).count()

   B) how many orders are pending.
   db .orders.find({Status:'P'}).count()

viii. Display all customer names of orders collection with no repetition
   db.orders.distinct("Cust_name")
   ["Aryan", "Carol", "Sam"]

ix. A)Find Total no of orders received so far
   db.orders.find({Status:'D'}).count()
   2

   B)how many orders are pending.
   db.orders.find({Status:'P'}).count()
   4

x. Show results and details of sorting documents based on amount

```
db.orders.find().sort({Amt:1}).pretty()
```

xi. Show how many orders are placed by each customer.

```
db.orders.aggregate({$group:{_id:"$Cust_name",cnt_of_order:{$sum:1}}})
```

xii. Display all customer ids and their total pending order amount in descending order.

```
db.orders.aggregate({$match:{Status:'P'}}, {$group:{_id:"$Cust_id",
pend_amt: {$sum:"$Amt"}}},{$sort:{pend_amt:-1}})
```

xiii. Display all customer ids in ascending order with total order amount which have been is delivered.

```
db.orders.aggregate({$match:{Status:'D'}},{$group:{_id:"$Cust_id",tot_amt:{$su
m: "$Amt"}}},{$sort:{_id:1}})
```

xiv. Show top three Selling Items from orders collection.

```
db.orders.aggregate({$group:{_id:"$Item_name",totqty:{$sum:"$Qty"}}}, {$sort:
{totqty:-1}},{$limit:3})
```

xv. Find the date on which maximum orders are received.

```
db.orders.aggregate({$group:{_id:"$DtOfOrder",cnt_of_order:{$sum:1}}},{$sort:
{cnt_of_order:-1}},{$limit:1})
{ "_id" : ISODate("2017-02-12T00:00:00Z"), "cnt_of_order" : 3 }
```

xvi. Find which customer has placed maximum orders.

```
db.orders.aggregate({$group:{_id:"$Cust_name",cnt_orderid:{$sum:1}}},{$sort:
{cnt_orderid:-1}},{$limit:1})
{ "_id" : "Sam", "cnt_orderid" : 2 }
```