

1. In database Employee.

1. find the average salary of each dept.

```
> db.empdetails.aggregate([{$group:{_id:"$dept","avg_salary":{$avg:"$salary"}}}])
{ "_id" : "developer", "avg_salary" : 45000 }
{ "_id" : "designer", "avg_salary" : 55000 }
{ "_id" : "tester", "avg_salary" : 72500 }
```

2. find the minimum salary of each dept.

```
> db.empdetails.aggregate([{$group:{_id:"$dept","Min_salary":{$min:"$salary"}}}])
{ "_id" : "developer", "Min_salary" : 45000 }
{ "_id" : "designer", "Min_salary" : 55000 }
{ "_id" : "tester", "Min_salary" : 50000 }
```

3. find the average salary of each dept.

```
> db.empdetails.aggregate([{$group:{_id:"$dept","Max_salary":{$max:"$salary"}}}])
{ "_id" : "tester", "Max_salary" : 95000 }
{ "_id" : "developer", "Max_salary" : 45000 }
{ "_id" : "designer", "Max_salary" : 55000 }
```

4. find the no.of employees of each dept.

```
> db.empdetails.aggregate([{$group:{_id:"$dept","No of employees":{$sum:1}}}])
{ "_id" : "tester", "No of employees" : 2 }
{ "_id" : "developer", "No of employees" : 1 }
{ "_id" : "designer", "No of employees" : 1 }
```

5. sort the collection empDetails in descending order of name

```
> db.empdetails.find().sort({name:-1})
{ "_id" : ObjectId("629b20db8badb7dab6e83357"), "name" : "sanu", "age" : 23, "email" : "ebc@gml.com", "salary" : 55000, "dept" : "designer" }
{ "_id" : ObjectId("629b21138badb7dab6e83358"), "name" : "janu", "age" : 21, "email" :
```

```
"fbc@gml.com", "salary" : 95000, "dept" : "tester" }

{ "_id" : ObjectId("629b214a8badb7dab6e83359"), "name" : "ganu", "age" : 20, "email" :
"kbc@gml.com", "salary" : 50000, "dept" : "tester" }

{ "_id" : ObjectId("629b209b8badb7dab6e83355"), "name" : "anu", "age" : 23, "email" :
"green@gml.com", "salary" : 45000, "dept" : "developer" }
```

6. Create a text index for 'name' and search for names mohan and bhuvan

```
> db.empdetails.createIndex({name:"text"})

{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}

> db.empdetails.find({$text:{$search:"sanu ganu"}})

{ "_id" : ObjectId("629b214a8badb7dab6e83359"), "name" : "ganu", "age" : 20, "email" :
"kbc@gml.com", "salary" : 50000, "dept" : "tester" }

{ "_id" : ObjectId("629b20db8badb7dab6e83357"), "name" : "sanu", "age" : 23, "email" :
"ebc@gml.com", "salary" : 55000, "dept" : "designer" }
```

2. create a database Inventory and create an orders collection. Apply MapReduce operation for finding the total purchase of each customer.

```
> use inventory
switched to db inventory
> db.createCollection("order")
```

```

{ "ok" : 1 }

> db.order.insert({custid:200,name:"maya",item:"rice",price:340})
WriteResult({ "nInserted" : 1 })

> db.order.insert({custid:200,name:"maya",item:"wheat",price:250})
WriteResult({ "nInserted" : 1 })

> db.order.insert({custid:201,name:"manu",item:"rice",price:340})
WriteResult({ "nInserted" : 1 })

> db.order.insert({custid:202,name:"meera",item:"sugar",price:150})
WriteResult({ "nInserted" : 1 })

> db.order.insert({custid:202,name:"meera",item:"wheat",price:250})
WriteResult({ "nInserted" : 1 })


> var mapFunction=function(){emit(this.custid,this.price);};
> var reduceFunction=function(key,values){return Array.sum(values);}
> db.order.mapReduce(mapFunction,reduceFunction,{out:"map_example"})
{ "result" : "map_example", "ok" : 1 }

```

```

> db.map_example.find();
{ "_id" : 200, "value" : 590 }
{ "_id" : 201, "value" : 340 }
{ "_id" : 202, "value" : 400 }

```

```

> var mapFunction=function(){emit(this.custid,this.price);};
> var reduceFunction=function(key,values){return Array.avg(values);}
> db.order.mapReduce(mapFunction,reduceFunction,{query:{custid:{$gt:201}},out:"map_example"})
{ "result" : "map_example", "ok" : 1 }

> db.map_example.find();
{ "_id" : 202, "value" : 200 }

```

Example Queries

```
>db.emp1.aggregate([{$group:{_id:"$dept","avg sal":{$avg:"$sal"}}}])
```

```
> db.emp1.aggregate([{$group:{_id:"$dept","minimum sal":{$min:"$sal"}}}])
```

```
> db.emp1.aggregate([{$group:{_id:"$dept","maximum sal":{$max:"$sal"}}}])
```

```
> db.emp1.aggregate([{$group:{_id:"$dept","no of emp":{$sum:1}}}])
```

```
> db.bookdata.find().sort({"Author":1})
```

```
> db.bookdata.find().sort({"Author":-1})
```

```
> db.bookdata.createIndex({"Title":"text"})
```

```
> db.bookdata.find({$text:{$search:"DBMS COA"}})
```