21. Create Database DYPIT using MongoDB Create following Collections Teachers(Tname,dno,dname,experience,salary,date_of_joining) Students(Sname,roll_no,class)

1. Find the information about all teachers

```
db.Teachers.find().pretty()
```

2. Find the average salary teachers of computer department

```
db.Teachers.aggregate([{$match:{"dname":"Computer"}},{$group : {_id : "$dname", salary_maximum : {$avg : "$salary"}}}])
```

3. Find the minimum and maximum salary of e&TC department teachers

```
db.Teachers.aggregate([{$match:{"dname":"E&TC"}},{$group : {_id : "$dname", salary_maximum : {$max : "$salary"}, salary_minimum:{$min : "$salary"}}}])
```

4. Find the information about all teachers of computer,IT,and E&TC department having salary greate than or equl to 10000/-

```
    db.Teachers.find({'salary':{$gte:10000}}).pretty()
```

5. Find the student information having roll_no = 2 or Sname=xyz

Same as above questions

6. Update the experience of teacher-praveen to 10years, if the entry is not available in database consider the entry as new entry.

Same s above questions.

7. Update the deparment of all the teachers working in IT deprtment to COMP

Same as above

8. find the teachers name and their experience from teachers collection
9. db.Teachers.find({},{dname:0,dno:0,salary:0,date_of_joining:0}).pretty()Using Save() method insert one entry in department collection

Same as above

10. Find the total salary all teachers.

db.Teachers.aggregate([{$group : {_id : "", total_salary : {$sum : "$salary"}}}])

22. Create Database DYPIT using MongoDB Create following Collections Teachers(Tname,dno,dname,experience,salary,date_of_joining) Students(Sname,roll_no,class)

1. Display the department wise average salary

db.Teachers.aggregate([{$group : {_id : "$dname", salary_avarage : {$avg : "$salary"}}}])

2. display the no. Of employees working in each department

db.Teachers.aggregate([{ $unwind: "$dname" }, { $sortByCount: "$dname" }])

3. Display the department wise total salary of departments having total salary greater than or equals to 50000/-

4. Write the queries using the different operators like max, min. Etc.

Refer above quetion

5. Create unique index on any field for above given collections

db.Teachers.createIndex({Tname:1}, {unique:true})

6. Create compound index on any fields for above given collections

7. Show all the indexes created in the database DYPIT

db.Teachers.getIndexes()

8. Show all the indexes created in above collections.

```
db.Teachers.getIndexes()
```