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
2. Find the average salary teachers of computer department
3. Find the minimum and maximum salary of e&TC department teachers
4. Find the information about all teachers of computer,IT,and E&TC department having salary greate than or equl to 10000/-
5. Find the student information having roll_no = 2 or Sname=xyz
6. Update the experience of teacher-praveen to 10years, if the entry is not available in database consider the entry as new entry.
7. Update the deparment of all the teachers working in IT deprtment to COMP
8. find the teachers name and their experience from teachers collection
9. Using Save() method insert one entry in department collection
10. Find the total salary all teachers.

```
use DYPIT

db.createCollection("Teachers");

db.createCollection("Students");


db.Teachers.insertMany([
      { Tname: "Praveen", dno: 101, dname: "Computer", experience: 8, salasalary:
12000, date_of_joining: new Date("2018-08-01") },
      { Tname: "Asha", dno: 102, dname: "E&TC", experience: 5, salary: 15000,
date_of_joining: new Date("2019-02-12") },
      { Tname: "Kiran", dno: 103, dname: "IT", experience: 4, salary: 9000,
date_of_joining: new Date("2020-05-17") },
      { Tname: "John", dno: 104, dname: "Computer", experience: 6, salary:
 11000, date_of_joining: new Date("2017-10-10") }
  ]);


db.Students.insertMany([
      { Sname: "Alice", roll_no: 1, class: "10th" },
      { Sname: "Bob", roll_no: 2, class: "10th" },
      { Sname: "xyz", roll_no: 3, class: "12th" }
  ]);


 db.Teachers.find({});


 db.Teachers.aggregate([
      { $match: { dname: "Computer" } },
      { $group: { _id: "$dname", avgSalary: { $avg: "$salary" } } }
  ]);
  db.Teachers.aggregate([
      { $match: { dname: "E&TC" } },
      { $group: {
          _id: "$dname",
          minSalary: { $min: "$salary" },
          maxSalary: { $max: "$salary" }
      }}
  ]);


  db.Teachers.find({
      dname: { $in: ["Computer", "IT", "E&TC"] },
      salary: { $gte: 10000 }
  });


db.Students.find({
      $or: [{ roll_no: 2 }, { Sname: "xyz" }]
  });

  db.Teachers.updateOne(
      { Tname: "Praveen" },
      { $set: { experience: 10 } },
      { upsert: true }
  );


  db.Teachers.updateMany(
      { dname: "IT" },
      { $set: { dname: "COMP" } }
  );


  db.Teachers.find({}, { Tname: 1, experience: 1, _id: 0 });


  db.Department.insertOne({
      dname: "Mechanical",
      dno: 105,
      location: "Block B"
  });


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



-----------------------------------------------------------------------------------
--------
-----------------------------------------------------------------------------------
---------
```

outputs of 19, 20, 21

19.

Create Database DYPIT using MongoDB

```
> use DYPIT
switched to db DYPIT
```

Create following Collections

Teachers(Tname,dno,dname,experience,salary,date_of_joining)

Students(Sname,roll_no,class)

```
  db.createCollection("Teachers")
{ ok: 1 }
  db.createCollection("Students")
{ ok: 1 }


  db.Teachers.insertMany([
      { Tname: "Praveen", dno: 1, dname: "Computer", experience: 8, salary: 12000,
date_of_joining: new Date("2015-08-
01") },
      { Tname: "Rajesh", dno: 2, dname: "IT", experience: 6, salary: 9000,
date_of_joining: new Date("2018-09-01") },
      { Tname: "Sneha", dno: 3, dname: "E&TC", experience: 10, salary: 15000,
date_of_joining: new Date("2012-06-01") }
  ])
```

O/P:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672ee1d9ba6a7b47a0c73bf8'),
    '1': ObjectId('672ee1d9ba6a7b47a0c73bf9'),
    '2': ObjectId('672ee1d9ba6a7b47a0c73bfa')
  }
}
```

```
  db.Students.insertMany([
      { Sname: "John", roll_no: 1, class: "10" },
      { Sname: "xyz", roll_no: 2, class: "10" },
      { Sname: "Alice", roll_no: 3, class: "10" }
  ])
```

o/p:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672ee1eeba6a7b47a0c73bfb'),
    '1': ObjectId('672ee1eeba6a7b47a0c73bfc'),
    '2': ObjectId('672ee1eeba6a7b47a0c73bfd')
  }
```

1. Find the information about all teachers

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

o/p:-

```
[
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bf8'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'Computer',
    experience: 8,
    salary: 12000,
    date_of_joining: ISODate('2015-08-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bf9'),
    Tname: 'Rajesh',
    dno: 2,
    dname: 'IT',
    experience: 6,
    salary: 9000,
    date_of_joining: ISODate('2018-09-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bfa'),
    Tname: 'Sneha',
    dno: 3,
    dname: 'E&TC',
    experience: 10,
    salary: 15000,
    date_of_joining: ISODate('2012-06-01T00:00:00.000Z')
  }
```

2. Find the information about all teachers of computer department

```
  db.Teachers.find({ dname: "Computer" })
[
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bf8'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'Computer',
    experience: 8,
    salary: 12000,
    date_of_joining: ISODate('2015-08-01T00:00:00.000Z')
  }
]
```

3. Find the information about all teachers of computer,IT,and e&TC department

```
  db.Teachers.find({ dname: { $in: ["Computer", "IT", "E&TC"] } })

o/p:
[
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bf8'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'Computer',
    experience: 8,
    salary: 12000,
    date_of_joining: ISODate('2015-08-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bf9'),
    Tname: 'Rajesh',
    dno: 2,
    dname: 'IT',
    experience: 6,
    salary: 9000,
    date_of_joining: ISODate('2018-09-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bfa'),
    Tname: 'Sneha',
    dno: 3,
    dname: 'E&TC',
    experience: 10,
    salary: 15000,
    date_of_joining: ISODate('2012-06-01T00:00:00.000Z')
  }
]
```

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({ dname: { $in: ["Computer", "IT", "E&TC"] }, salary: { $gte: 
10000 } })
```

o/p:

```
[
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bf8'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'Computer',
    experience: 8,
    salary: 12000,
    date_of_joining: ISODate('2015-08-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bfa'),
    Tname: 'Sneha',
    dno: 3,
    dname: 'E&TC',
    experience: 10,
    salary: 15000,
    date_of_joining: ISODate('2012-06-01T00:00:00.000Z')
  }
]
```

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

```
  db.Students.find({ $or: [{ roll_no: 2 }, { Sname: "xyz" }] })
```

O/P:

```
[
  {
    _id: ObjectId('672ee1eeba6a7b47a0c73bfc'),
    Sname: 'xyz',
    roll_no: 2,
    class: '10'
  }
]
```

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

```
  db.Teachers.updateOne(
      { Tname: "Praveen" },
      { $set: { experience: 10 } },
      { upsert: true }
  )
```

```
O/P:
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

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

```
  db.Teachers.updateMany(
      { dname: "IT" },
      { $set: { dname: "COMP" } }
  );
```

```
O/P:
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

8. find the teachers name and their experience from teachers collection

```
  db.Teachers.find({}, { Tname: 1, experience: 1, _id: 0 })
```

```
O/P:
[
  { Tname: 'Praveen', experience: 10 },
  { Tname: 'Rajesh', experience: 6 },
  { Tname: 'Sneha', experience: 10 }
]
```

9. Using Save() method insert one entry in department collection

```
  db.Teachers.find({}, { Tname: 1, experience: 1, _id: 0 })
```

```
O/P:
[
  { Tname: 'Praveen', experience: 10 },
  { Tname: 'Rajesh', experience: 6 },
  { Tname: 'Sneha', experience: 10 }
]
```

10. Using Save() method change the dept of teacher Rajesh to IT

```
  db.Teachers.updateOne(
      { Tname: "Rajesh" },
      { $set: { dname: "IT" } }
  )
```

```
O/P:
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

11. Delete all the doccuments from teachers collection having IT dept.

```
  db.Teachers.deleteMany({ dname: "IT" })
```

O/P:

```
{ acknowledged: true, deletedCount: 1 }
```

12. display with pretty() method, the first 3 doccuments in teachers collection in ascending order

```
  db.Teachers.find().sort({ Tname: 1 }).limit(3).pretty()
```

O/P:

```
[
  {
    _id: ObjectId('672eeeb6ba6a7b47a0c73bfe'),
    Tname: 'Anita',
    dno: 4,
    dname: 'Mechanical',
    experience: 5,
    salary: 11000,
    date_of_joining: ISODate('2017-05-15T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bf8'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'Computer',
    experience: 10,
    salary: 12000,
    date_of_joining: ISODate('2015-08-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ee1d9ba6a7b47a0c73bfa'),
    Tname: 'Sneha',
    dno: 3,
    dname: 'E&TC',
    experience: 10,
    salary: 15000,
    date_of_joining: ISODate('2012-06-01T00:00:00.000Z')
  }
]
```

```
-----------------------------------------------------------------------------------
-----------------------
```

20

1.Create Database DYPIT

```
 use DYPIT
switched to db DYPIT
```

2. Create following Collections

```
Teachers(Tname,dno,dname,experience,salary,date_of_joining )
Students(Sname,roll_no,class)
  db.createCollection("Teachers")
{ ok: 1 }
  db.createCollection("Students")
{ ok: 1 }


  db.Teachers.insertMany([
    {
      Tname: "John",
      dno: 1,
      dname: "Computer",
      experience: 5,
      salary: 30000,
      date_of_joining: ISODate("2018-05-10")
    },
    {
      Tname: "Alice",
      dno: 2,
      dname: "IT",
      experience: 3,
      salary: 28000,
      date_of_joining: ISODate("2019-06-20")
    },
    {
      Tname: "David",
      dno: 3,
      dname: "E&TC",
      experience: 4,
      salary: 27000,
      date_of_joining: ISODate("2020-01-15")
    },
    {
      Tname: "Sara",
      dno: 1,
      dname: "Computer",
      experience: 8,
      salary: 32000,
      date_of_joining: ISODate("2016-04-11")
    },
    {
      Tname: "Praveen",
      dno: 2,
      dname: "IT",
      experience: 2,
      salary: 24000,
      date_of_joining: ISODate("2021-02-19")
    },
    {
      Tname: "Mary",
      dno: 4,
      dname: "Mechanical",
      experience: 6,
      salary: 26000,
      date_of_joining: ISODate("2017-07-25")
    }
  ])
```

O/P:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672ef59aba6a7b47a0c73bff'),
    '1': ObjectId('672ef59aba6a7b47a0c73c00'),
    '2': ObjectId('672ef59aba6a7b47a0c73c01'),
    '3': ObjectId('672ef59aba6a7b47a0c73c02'),
    '4': ObjectId('672ef59aba6a7b47a0c73c03'),
    '5': ObjectId('672ef59aba6a7b47a0c73c04')
  }
}
```

```
  db.Students.insertMany([
    {
      Sname: "Ravi",
      roll_no: 21,
      class: "10"
    },
    {
      Sname: "xyz",
      roll_no: 25,
      class: "10"
    },
    {
      Sname: "Arjun",
      roll_no: 30,
      class: "11"
    },
    {
      Sname: "Simran",
      roll_no: 35,
      class: "11"
    },
    {
      Sname: "Priya",
      roll_no: 40,
      class: "12"
    }
  ])
```

O/P:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672ef5aaba6a7b47a0c73c05'),
    '1': ObjectId('672ef5aaba6a7b47a0c73c06'),
    '2': ObjectId('672ef5aaba6a7b47a0c73c07'),
    '3': ObjectId('672ef5aaba6a7b47a0c73c08'),
    '4': ObjectId('672ef5aaba6a7b47a0c73c09')
  }
}
```

3. Find the information about two teachers

```
  db.Teachers.find().limit(2)
```

O/P:

```
[
  {
    _id: ObjectId('672ef59aba6a7b47a0c73bff'),
    Tname: 'John',
    dno: 1,
    dname: 'Computer',
    experience: 5,
    salary: 30000,
    date_of_joining: ISODate('2018-05-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c00'),
    Tname: 'Alice',
    dno: 2,
    dname: 'IT',
    experience: 3,
    salary: 28000,
    date_of_joining: ISODate('2019-06-20T00:00:00.000Z')
  }
]
```

4. Find the information about all teachers of computer department

```
  db.Teachers.find({ dname: "Computer" })
```

O/P:

```
[
  {
    _id: ObjectId('672ef59aba6a7b47a0c73bff'),
    Tname: 'John',
    dno: 1,
    dname: 'Computer',
    experience: 5,
    salary: 30000,
    date_of_joining: ISODate('2018-05-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c02'),
    Tname: 'Sara',
    dno: 1,
    dname: 'Computer',
    experience: 8,
    salary: 32000,
    date_of_joining: ISODate('2016-04-11T00:00:00.000Z')
  }
]
```

5. Find the information about all teachers of computer,IT,and e&TC department

```
  db.Teachers.find({ dname: { $in: ["Computer", "IT", "E&TC"] } })
```

O/P:

```
[
  {
    _id: ObjectId('672ef59aba6a7b47a0c73bff'),
    Tname: 'John',
    dno: 1,
    dname: 'Computer',
    experience: 5,
    salary: 30000,
    date_of_joining: ISODate('2018-05-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c00'),
    Tname: 'Alice',
    dno: 2,
    dname: 'IT',
    experience: 3,
    salary: 28000,
    date_of_joining: ISODate('2019-06-20T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c01'),
    Tname: 'David',
    dno: 3,
    dname: 'E&TC',
    experience: 4,
    salary: 27000,
    date_of_joining: ISODate('2020-01-15T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c02'),
    Tname: 'Sara',
    dno: 1,
    dname: 'Computer',
    experience: 8,
    salary: 32000,
    date_of_joining: ISODate('2016-04-11T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c03'),
    Tname: 'Praveen',
    dno: 2,
    dname: 'IT',
    experience: 2,
    salary: 24000,
    date_of_joining: ISODate('2021-02-19T00:00:00.000Z')
  }
]
```

6. Find the information about all teachers of computer,IT,and E&TC department having salary greate than or equl to 25000/-

```
  db.Teachers.find({ dname: { $in: ["Computer", "IT", "E&TC"] }, salary: { $gte: 25000 } })
```

```
O/P:
[
  {
    _id: ObjectId('672ef59aba6a7b47a0c73bff'),
    Tname: 'John',
    dno: 1,
    dname: 'Computer',
    experience: 5,
    salary: 30000,
    date_of_joining: ISODate('2018-05-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c00'),
    Tname: 'Alice',
    dno: 2,
    dname: 'IT',
    experience: 3,
    salary: 28000,
    date_of_joining: ISODate('2019-06-20T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c01'),
    Tname: 'David',
    dno: 3,
    dname: 'E&TC',
    experience: 4,
    salary: 27000,
    date_of_joining: ISODate('2020-01-15T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c02'),
    Tname: 'Sara',
    dno: 1,
    dname: 'Computer',
    experience: 8,
    salary: 32000,
    date_of_joining: ISODate('2016-04-11T00:00:00.000Z')
  }
]
```

7. Find the student information having roll_no = 25 or Sname=xyz

```
  db.Students.find({ $or: [{ roll_no: 25 }, { Sname: "xyz" }] })
```

O/P:

```
[
  {
    _id: ObjectId('672ef5aaba6a7b47a0c73c06'),
    Sname: 'xyz',
    roll_no: 25,
    class: '10'
  }
]
```

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

```
  db.Teachers.updateOne(
    { Tname: "Praveen" },
    { $set: { experience: 10, Tname: "Praveen" } },
    { upsert: true }
  )
```

O/P:

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

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

```
  db.Teachers.updateMany(
    { dname: "IT" },
    { $set: { dname: "COMP" } }
  )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

10. find the teachers name and their experience from teachers collection

```
  db.Teachers.find({}, { Tname: 1, experience: 1, _id: 0 })
```

O/P:

```
[
  { Tname: 'John', experience: 5 },
  { Tname: 'Alice', experience: 3 },
  { Tname: 'David', experience: 4 },
  { Tname: 'Sara', experience: 8 },
  { Tname: 'Praveen', experience: 10 },
  { Tname: 'Mary', experience: 6 }
]
```

11. Using Save() method insert one entry in department collection

```
  db.Department.insertOne({ dno: 1, dname: "Computer", location: "Building A" })
```

O/P:

```
{
  acknowledged: true,
  insertedId: ObjectId('672ef645ba6a7b47a0c73c0a')
}
```

13. Delete all the documents from teachers collection having IT dept.

```
  db.Teachers.deleteMany({ dname: "IT" })
```

O/P:

```
{ acknowledged: true, deletedCount: 0 }
```

14. display with pretty() method, the first 5 documents in teachers collection in ascending order

```
  db.Teachers.find().sort({ Tname: 1 }).limit(5).pretty()
```

O/P:

```
[
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c00'),
    Tname: 'Alice',
    dno: 2,
    dname: 'COMP',
    experience: 3,
    salary: 28000,
    date_of_joining: ISODate('2019-06-20T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c01'),
    Tname: 'David',
    dno: 3,
    dname: 'E&TC',
    experience: 4,
    salary: 27000,
    date_of_joining: ISODate('2020-01-15T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73bff'),
    Tname: 'John',
    dno: 1,
    dname: 'Computer',
    experience: 5,
    salary: 30000,
    date_of_joining: ISODate('2018-05-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c04'),
    Tname: 'Mary',
    dno: 4,
    dname: 'Mechanical',
    experience: 6,
    salary: 26000,
    date_of_joining: ISODate('2017-07-25T00:00:00.000Z')
  },
  {
    _id: ObjectId('672ef59aba6a7b47a0c73c03'),
    Tname: 'Praveen',
    dno: 2,
    dname: 'COMP',
    experience: 10,
    salary: 24000,
    date_of_joining: ISODate('2021-02-19T00:00:00.000Z')
  }
]
```

```
-----------------------------------------------------------------------------------
------
```

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

```
  db.createCollection("Teachers");

  db.createCollection("Students");


  db.Teachers.insertMany([
      { Tname: "Praveen", dno: 101, dname: "Computer", experience: 8, salasalary:
12000, date_of_joining: new Date("2018-08-01") },
      { Tname: "Asha", dno: 102, dname: "E&TC", experience: 5, salary: 15000,
date_of_joining: new Date("2019-02-12") },
      { Tname: "Kiran", dno: 103, dname: "IT", experience: 4, salary: 9000,
date_of_joining: new Date("2020-05-17") },
      { Tname: "John", dno: 104, dname: "Computer", experience: 6, salary:
 11000, date_of_joining: new Date("2017-10-10") }
  ]);
```

```
O/P:
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672f00c3ba6a7b47a0c73c0b'),
    '1': ObjectId('672f00c3ba6a7b47a0c73c0c'),
    '2': ObjectId('672f00c3ba6a7b47a0c73c0d'),
    '3': ObjectId('672f00c3ba6a7b47a0c73c0e')
  }
}


  db.Students.insertMany([
      { Sname: "Alice", roll_no: 1, class: "10th" },
      { Sname: "Bob", roll_no: 2, class: "10th" },
      { Sname: "xyz", roll_no: 3, class: "12th" }
  ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672f00c3ba6a7b47a0c73c0f'),
    '1': ObjectId('672f00c3ba6a7b47a0c73c10'),
    '2': ObjectId('672f00c3ba6a7b47a0c73c11')
  }
}
```

1. Find the information about all teachers

```
  db.Teachers.find({});

O/P:
[
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c0b'),
    Tname: 'Praveen',
    dno: 101,
    dname: 'Computer',
    experience: 8,
    salary: 12000,
    date_of_joining: ISODate('2018-08-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c0c'),
    Tname: 'Asha',
    dno: 102,
    dname: 'E&TC',
    experience: 5,
    salary: 15000,
    date_of_joining: ISODate('2019-02-12T00:00:00.000Z')
  },
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c0d'),
    Tname: 'Kiran',
    dno: 103,
    dname: 'IT',
    experience: 4,
    salary: 9000,
    date_of_joining: ISODate('2020-05-17T00:00:00.000Z')
  },
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c0e'),
    Tname: 'John',
    dno: 104,
    dname: 'Computer',
    experience: 6,
    salary: 11000,
    date_of_joining: ISODate('2017-10-10T00:00:00.000Z')
  }
]
```

2. Find the average salary teachers of computer department

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

O/P:

```
[ { _id: 'Computer', avgSalary: 11500 } ]
```

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

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

O/P:

```
[ { _id: 'E&TC', minSalary: 15000, maxSalary: 15000 } ]
```

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({
      dname: { $in: ["Computer", "IT", "E&TC"] },
      salary: { $gte: 10000 }
  });
```

O/P:

```
[
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c0b'),
    Tname: 'Praveen',
    dno: 101,
    dname: 'Computer',
    experience: 8,
    salary: 12000,
    date_of_joining: ISODate('2018-08-01T00:00:00.000Z')
  },
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c0c'),
    Tname: 'Asha',
    dno: 102,
    dname: 'E&TC',
    experience: 5,
    salary: 15000,
    date_of_joining: ISODate('2019-02-12T00:00:00.000Z')
  },
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c0e'),
    Tname: 'John',
    dno: 104,
    dname: 'Computer',
    experience: 6,
    salary: 11000,
    date_of_joining: ISODate('2017-10-10T00:00:00.000Z')
  }
]
```

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

```
  db.Students.find({
      $or: [{ roll_no: 2 }, { Sname: "xyz" }]
  });
```

O/P:

```
[
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c10'),
    Sname: 'Bob',
    roll_no: 2,
    class: '10th'
  },
  {
    _id: ObjectId('672f00c3ba6a7b47a0c73c11'),
    Sname: 'xyz',
    roll_no: 3,
    class: '12th'
  }
]
```

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

```
  db.Teachers.updateOne(
      { Tname: "Praveen" },
      { $set: { experience: 10 } },
      { upsert: true }
  );
```

O/P:

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

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

```
  db.Teachers.updateMany(
      { dname: "IT" },
      { $set: { dname: "COMP" } }
  );
```

O/P:

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

8. find the teachers name and their experience from teachers collection

```
  db.Teachers.find({}, { Tname: 1, experience: 1, _id: 0 });
```

O/P:

```
[
  { Tname: 'Praveen', experience: 10 },
  { Tname: 'Asha', experience: 5 },
  { Tname: 'Kiran', experience: 4 },
  { Tname: 'John', experience: 6 }
]
```

9. Using Save() method insert one entry in department collection

```
  db.Department.insertOne({
      dname: "Mechanical",
      dno: 105,
      location: "Block B"
  });
```

O/P:

```
{
  acknowledged: true,
  insertedId: ObjectId('672f01a3ba6a7b47a0c73c12')
}
```

10. Find the total salary all teachers.

```
  db.Teachers.aggregate([
      { $group: { _id: null, totalSalary: { $sum: "$salary" } } }
  ]);
```

O/P:

```
[ { _id: null, totalSalary: 47000 } ]
```