Question 22.
Create Database DYPIT using MongoDB

Step 1: use DYPIT

```
db.Teachers.insertMany([
    { Tname: "Sharad Sir", dno: 1, dname: "Mathematics", experience: 10, salary:
55000, date_of_joining: new Date("2015-08-01") },
    { Tname: "Pranal Mam", dno: 2, dname: "Science", experience: 8, salary: 48000,
date_of_joining: new Date("2016-01-15") },
    { Tname: "Arti Mam", dno: 1, dname: "Mathematics", experience: 5, salary:
45000, date_of_joining: new Date("2017-03-10") },
    { Tname: "Kapil Sir", dno: 3, dname: "History", experience: 6, salary: 52000,
date_of_joining: new Date("2016-05-22") },
    { Tname: "Pramod sir", dno: 2, dname: "Science", experience: 4, salary: 47000,
date_of_joining: new Date("2018-10-08") }
]);
```

Step 2:

```
db.Students.insertMany([
    { Sname: "Kedar M", roll_no: 21, class: "TE-A" },
    { Sname: "Soham K", roll_no: 32, class: "TE-B" },
    { Sname: "Nishant K", roll_no: 13, class: "TE-A" },
    { Sname: "Vivek J", roll_no: 24, class: "TE-C" },
    { Sname: "Vedant M", roll_no: 35, class: "TE-A" }
]);
```

Query 1: Display the Department-Wise Average Salary

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

Query 2: Display the Number of Employees Working in Each Department

```
    db.Teachers.aggregate([
    { $group: { _id: "$dname", numEmployees: { $sum: 1 } } }
]);
```

Query 3: Display the Department-Wise Total Salary of Departments with Total Salary ≥ 50000

```
    db.Teachers.aggregate([
        { $group: { _id: "$dname", totalSalary: { $sum: "$salary" } } },
        { $match: { totalSalary: { $gte: 50000 } } }
]);
```

Query 4: Using Aggregation Operators like $max, $min

1. Find Maximum Salary per Department:

```
        db.Teachers.aggregate([
    { $group: { _id: "$dname", maxSalary: { $max: "$salary" } } }
]);
```

2. Find Minimum Salary per Department:

```
        db.Teachers.aggregate([
    { $group: { _id: "$dname", minSalary: { $min: "$salary" } } }
]);
```

Query 5: Create unique index on any field for above given collections

1. creating index on Teachers Collection (e.g., unique index on Tname to ensure

```
unique teacher names)
        db.Teachers.createIndex({ Tname: 1 }, { unique: true });

    2. creating index on Students Collection (e.g., unique index on roll_no to
ensure unique roll numbers):
        db.Students.createIndex({ roll_no: 1 }, { unique: true });

Query 6:Create compound index on any fields for above given collections
    1. creating compound index on Teachers Collection (e.g., on dno and
experience):
        db.Teachers.createIndex({ dno: 1, experience: -1 });

Query 7:Show all the indexes created in the database DYPIT
    db.getCollectionNames().forEach(function(collection) {
    print("Indexes for collection: " + collection);
    printjson(db.getCollection(collection).getIndexes());
});

Query 8: Show all the indexes created in above collections.
    1. db.Teachers.getIndexes();
    2. db.Students.getIndexes();
```