Lab 2

**MongoDB**

1) Using MongoDB

i) Create a database for Students and Create a Student Collection (\_id,Name, USN,Semester, Dept\_Name, CGPA, Hobbies(Set)).

> use Students switched to db Students ii) Insert required documents to the collection.

> db.Student.insert({Studname:"Nithin",USN:"1BM19CS106",Semester:

"VII",Dept\_name:"Computer

Science”, CGPA:9.6, Hobbies:["Sleep","eat"]});

WriteResult ({“nInserted" : 1 })

> db.Student.insert({Studname:"Rahul",USN:"1BM19CS105",Semester:

"VI",Dept\_name:"Computer

Science”, CGPA:8.6, Hobbies:["Sleep","eat"]});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({Studname:"Hailey",USN:"1BM19CS015",Semester

:"VIII",Dept\_name:"Computer

Science”, CGPA:7.4, Hobbies:["Sleep","eat","repeat"]}); WriteResult({ "nInserted" : 1 })

1. First Filter on “Dept\_Name:CSE” and then group it on “Semester”and compute the Average CPGA for that semester and filter those documents where the “Avg\_CPGA” is greater than 7.5.

> db.Student.aggregate({$match:{Dept\_name:"Computer

Science"}}, {$group: {\_id:"$Semester”, AvgCGPA: {$avg:"$CGPA"}}},{$m atch:{AvgCGPA:{$gt:7.5}}});

{“\_id”: "VIII", "AvgCGPA”: 8.6 }

{“\_id" : "VII", "AvgCGPA" : 8.533333333333333 } { "\_id" : "VI", "AvgCGPA" : 8.266666666666667 }

1. Command used to export MongoDB JSON documents from“Student” Collection into the “Students” database into a CSV file “Output.txt”.

2)Create a mongo dB collection Bank. Demonstrate the following by choosing fields of your choice.

> db.createCollection("Bank");

{“ok" : 1 }

1. Insert three documents

db. Bank.insert({\_id:1, name:"Ramesh”, state:"Gujarat",country:"India",language:["gujarati","marat hi","english"]})

db. Bank.insert({\_id:2, name:"Mahesh”, state:"Gujarat",country:"India",language:["gujarati","marwa di","english"]})

db.Bank.insert({\_id:3,name:"Ghela

bhai”, state:"Maharashta",country:"India",language:["marathi","marwadi","english"]})

1. Use Arrays (Use Pull and Pop operation)

db.Bank.update({\_id: 1}, {$push: {language: "hindi"}}) db.Bank.update({\_id: 2}, {$pull: {language: "english"}})

1. Use Index
2. Use Cursors
3. Updation

3) Consider a table “Students” with the following columns:

1. StudRollNo / \_id
2. StudName
3. Grade
4. Hobbies
5. DOJ

Write MongoDB queries for the following:

1. To display only the students name from all the documents ofthe Students collection.

> db.Students.find({},{Studname:1,\_id:0});

{“Studname”: "raj”}

{“Studname”: "varun”}

{“Stud name”: "Lodi”}

{“Studname”: "Modi" }

{“Studname”: "Nithin" }

1. To display only the student’s name, grade as well as theidentifier from the document of the Student collection where the \_id column is 1. > db.Students.find({\_id:{$eq:ObjectId("625fd1171e24dbace73bd604")}

},{Studname:1,Grade:1,\_id:1});

{ "\_id" : ObjectId("625fd1171e24dbace73bd604"), "Studname" : "raj",

"Grade" : "VII" }

1. To find those documents where the grade is not set to VIII.

> db.Students.find({Grade:{$ne:"VII"}});

{“\_id" : ObjectId("625fd11d1e24dbace73bd605"), "Studname" : "varun", "Grade" : "VIII", "Hobbies" : [ "cricket" ], "DOJ" : "12/8/2021" } { "\_id" : ObjectId("625fd1241e24dbace73bd606"), "Studname" : "Lodi", "Grade" : "VIII", "Hobbies" : [ "Sleep" ], "DOJ" : "12/8/2021" } { "\_id" : ObjectId("625fd12d1e24dbace73bd607"), "Studname" :

"Modi", "Grade”: "VI", "Hobbies" : [ "Sleep", "eat" ], "DOJ" : "12/7/2001" }

1. To find those documents from the students collection wherethe hobbies is set to ’cricket’ and the student name is set to ‘varun’.

> db.Student.find({Hobbies :{

$in: ['cricket']},Studname:{$eq:"varun"}}).pretty ();

{

"\_id”: ObjectId("625fd0771e24dbace73bd602"),

"Studname”: "varun",

"Grade”: "VIII",

"Hobbies”: [

"cricket"

],

"DOJ”: "12/8/2021"

}

5.To find documents from the Students collection where the student name ends in ‘j’

> db.Student.find({Studname:/j$/}).pretty();

{

"\_id" : ObjectId("625fd09b1e24dbace73bd603"),

"Studname”: "raj",

"Grade”: "VII",

"Hobbies”: [

"cricket"

],

"DOJ" : "12/8/2021"

}

4) Using MongoDB,

i) Create a database for Faculty and Create a faculty

Collection (Faculty\_id, Name, Designation, Department, Age, Salary, Specialization(Set)). > use faculty switched to db faculty

> db.createCollection("Faculty");

{ "ok" : 1 } ii) Insert required documents to the collection.

>

db. Faculty.insert({Name:"Modi”, Designation:"Teacher",Department:" CSE",Age:90,Salary:40000,Specialization:["Eating","Talking","Web dev"]});

WriteResult({ "nInserted" : 1 })

> db.Faculty.insert({Name:"NIRANJAN",Designation:"Teacher",Depart ment:"MECH",Age:90,Salary:120000,Specialization:["Eating","Talking"

,"Web dev"]});

WriteResult({ "nInserted" : 1 })

> db. Faculty.insert({Name:"ugrasen”, Designation:"Assisstant",Departm ent:"MECH",Age:20,Salary:1000,Specialization:["Eating","Talking","We b dev"]});

WriteResult({ "nInserted" : 1 })

> db. Faculty.insert({Name:"rahul”, Designation:"Assisstant",Departmen t:"MECH",Age:20,Salary:111000,Specialization:["Eating","Talking","We b dev"]});

WriteResult({ "nInserted" : 1 })

1. First Filter on “Dept\_Name:MECH” and then group it on

“Designation” and compute the Average Salary for that Designation and filter those documents where the “Avg\_Sal” is greater than 6500.

>

db. Faculty.aggregate({$match: {Department:"MECH"}},{$group:{\_id:"$

Designation”, AvgSAL:{$avg:"$Salary"}}},{$match:{AvgSAL:{$gt:6500}

}});

{“\_id" : "Assisstant", "AvgSAL" : 56000 } { "\_id" : "Teacher", "AvgSAL" : 120000 }

1. Demonstrate usage of import and export commands