**Assignment 9**

db.students.insertMany([

{roll:1,name:"Ram"},

{roll:2,name:"Sham"},

{roll:3,name:"Ramesh"}]

)

db.students.updateOne({roll:1},{$set:{name:"rahul"}}

roll,name,branch,cgpa,city

db.students.insertMany([

{roll:1,name:"Ram",cgpa:8.7,branch:"COMP",native\_city:"Mumbai"},

{roll:2,name:"Sham",cgpa:8.5,branch:"COMP",native\_city:"Pune"},

{roll:3,name:"Ramesh",cgpa:8.4,branch:"COMP",native\_city:"Delhi"},

{roll:3,name:"Suresh",cgpa:9,branch:"COMP",native\_city:"Delhi"}

])

db.getCollectionNames()

db.students.updateOne(

db.students.find({cgpa:{$gte:8.4}},{name}

db.student.aggregate ([{$group : {\_id : "$subject", marks : {$sum : "$marks"}}}]);

db.student.find({ "Branch": "Computer" }).sort({ "CGPA": -1 }).limit(1)

**Assignment 10**

db.students.insertMany([

{rollno:1,name:'Ram',prn:'P1',Branch:'COMP',

TOC:99,

SPOS:99,

SPM:99,

DBMS:99},

{rollno:2,name:'Sham',prn:'P2',Branch:'IT',

TOC:10,

SPOS:23,

SPM:89,

DBMS:10},

{rollno:3,name:'Ramesh',prn:'P3',Branch:'IT',

TOC:56,

SPOS:23,

SPM:89,

DBMS:10},

{rollno:4,name:'Suresh',prn:'P4',Branch:'COMP',

TOC:56,

SPOS:56,

SPM:47,

DBMS:78},

{rollno:5,name:'Sita',prn:'P5',Branch:'ENTC',

TOC:56,

SPOS:3,

SPM:89,

DBMS:12},

{rollno:6,name:'Gita',prn:'P6',Branch:'ENTC',

TOC:56,

SPOS:23,

SPM:66,

DBMS:43},

{rollno:7,name:'Rajesh',prn:'P7',Branch:'COMP',

TOC:56,

SPOS:23,

SPM:79,

DBMS:41},

{rollno:8,name:'Raghu',prn:'P8',Branch:'IT',

TOC:56,

SPOS:23,

SPM:62,

DBMS:78}

]);

db.students.aggregate([

{ $group: { \_id: "$Branch", count: { $sum: 1 } } }

])

db.students.aggregate([

{

$addFields: {

totalMarks: {

$sum:["$TOC","$SPOS","$SPM","$DBMS"]

} }

},{ $sort: { totalMarks: -1 }},

{

$limit:1

}

])

db.students.aggregate([

{ $project: {

\_id:0, rollno:1, name:1, prn:1, Branch:1,

subjects: {

$objectToArray: {

TOC: "$TOC", SPOS: "$SPOS", SPM: "SPM", DBMS: "$DBMS",

}

}}}

{

$unwind:"$subjects"

}

{ $group : {

\_id: "subjects.k",

maxMarks: {$max: "$subjects.v"},

topper: {

$first: {

rollno: "$rollno",name:"$name",prn:"$prn", Branch:"$Branch"

}}}}

} ])

db.students.aggregate([

{ $project: {

\_id: 0,

TOC: 1,

SPOS: 1,

SPM: 1,

DBMS: 1

}

},

{

$group: {

\_id: null,

avgTOC: { $avg: "$TOC" },

avgSPOS: { $avg: "$SPOS" },

avgSPM: { $avg: "$SPM" },

avgDBMS: { $avg: "$DBMS" }

}

},

{

$project: { \_id: 0, avgTOC: 1, avgSPOS: 1, avgSPM: 1, avgDBMS: 1

}

}

])

db.students.aggregate([

{

$project: {

\_id: 0,

rollno: 1,

name: 1,

prn: 1,

Branch: 1,

TOC: 1

}

},

{

$match: {

TOC: { $ne: null }

}

},

{

$sort: {

TOC: 1

}

},

{

$limit: 1

}

])

db.students.createIndex({ "rollno": 1 })

db.students.createIndex({ "prn": 1 })

db.students.getIndexes()

db.students.dropIndex("rollno\_1")

**Assignment 11**

> use mydata

switched to db mydata

> db.createCollection("orders")

{ "ok" : 1 }

> db.orders.insert({"Cust\_ID":"A123","Amount":"500","Status":"A"})

WriteResult({ "nInserted" : 1 })

> db.orders.insert({"Cust\_ID":"A123","Amount":"250","Status":"A"});

WriteResult({ "nInserted" : 1 })

> db.orders.insert({"Cust\_ID":"B212","Amount":"200","Status":"A"});

WriteResult({ "nInserted" : 1 })

> db.orders.insert({"Cust\_ID":"A123","Amount":"300","Status":"D"});

WriteResult({ "nInserted" : 1 })

> db.orders.find()

{ "\_id" : ObjectId("635153e571cd95ab8e93d98b"), "Cust\_ID" : "A123", "Amount" : "500", "Status" : "A" }

{ "\_id" : ObjectId("635153f571cd95ab8e93d98c"), "Cust\_ID" : "A123", "Amount" : "250", "Status" : "A" }

{ "\_id" : ObjectId("6351540071cd95ab8e93d98d"), "Cust\_ID" : "B212", "Amount" : "200", "Status" : "A" }

{ "\_id" : ObjectId("6351540a71cd95ab8e93d98e"), "Cust\_ID" : "A123", "Amount" : "300", "Status" : "D" }

> db.orders.mapReduce(function(){emit(this.Cust\_ID,this.Amount);},function(key, values) {return Array.sum( values)},{query:{"Status":"A"},out: "order\_totals"})

{

"result" : "order\_totals",

"timeMillis" : 437,

"counts" : {

"input" : 3,

"emit" : 3,

"reduce" : 1,

"output" : 2

},

"ok" : 1

}

> var mapFunc1 = function(){emit(this.Cust\_ID,this.Amount);};

> var reduceFunc1 = function(keyCustID,valuePrices){return Array.sum(valuePrices);};

> db.orders.mapReduce(mapFunc1,reduceFunc1,{out:"Map\_Example2"})

{

"result" : "Map\_Example2",

"timeMillis" : 315,

"counts" : {

"input" : 4,

"emit" : 4,

"reduce" : 1,

"output" : 2

},

"ok" : 1

}

> db.order\_totals.find()

{ "\_id" : "A123", "value" : "500250" }

{ "\_id" : "B212", "value" : "200" }

> db.Map\_Example2.find()

{ "\_id" : "A123", "value" : "500250300" }

{ "\_id" : "B212", "value" : "200" }

> db.orders.mapReduce(function(){emit(this.Cust\_ID,this.Amount);},function(key, values) {return Array.avg( values)},{query:{"Status":"A"},out: "order\_totals"})

{

"result" : "order\_totals",

"timeMillis" : 346,

"counts" : {

"input" : 3,

"emit" : 3,

"reduce" : 1,

"output" : 2

},

"ok" : 1

}

> db.order\_totals.find().pretty()

{ "\_id" : "A123", "value" : 250125 }

{ "\_id" : "B212", "value" : "200" }

**Assignment 12**

from pymongo import MongoClient

# Modify the connection string based on your provided URL

connection\_string = "mongodb://localhost:27017/himal"

try:

conn = MongoClient(connection\_string)

print("Connected to MongoDB")

except Exception as e:

print(f"Could not connect to MongoDB: {e}")

exit()

# Access the "t4" collection

collection = conn.himal.t2

while True:

print("MENU:\n1. INSERT\n2. UPDATE\n3. DELETE\n4. SHOW TABLE\n5. EXIT\n")

ch = int(input("ENTER CHOICE: "))

if ch == 1:

rollno = input("ENTER ROLL NO: ")

name = input("ENTER NAME: ")

marks = input("ENTER MARKS: ")

record = {

"rollno": rollno,

"name": name,

"marks": marks

}

try:

collection.insert\_one(record)

print("Record inserted successfully.")

except Exception as e:

print(f"Error inserting record: {e}")

elif ch == 2:

rollno = input("ENTER ROLL NO: ")

marks = input("ENTER MARKS: ")

filter = {'rollno': rollno}

newvalues = {"$set": {'marks': marks}}

try:

collection.update\_one(filter, newvalues)

print("Record updated successfully.")

except Exception as e:

print(f"Error updating record: {e}")

elif ch == 3:

rollno = input("ENTER ROLL NO: ")

try:

collection.delete\_one({"rollno": rollno})

print("Record deleted successfully.")

except Exception as e:

print(f"Error deleting record: {e}")

elif ch == 4:

try:

cursor = collection.find()

for record in cursor:

print(record)

except Exception as e:

print(f"Error fetching records: {e}")

elif ch == 5:

conn.close()

print("Connection to MongoDB closed.")

exit()