ASSIGNMENT NO:1C

1. Show following insert validation . Store all validation in a single Rule

a. Accept the city from(Pune, Mumbai, Chennai, Delhi),

b. Salary Key (should be greater than 1000).

c. Accept the Grade ( A,B,C)

d. empid is a mandatory key.

e. name key is string datatype..

> db.createCollection("Emp",{validator:{$and:[{emp\_id:{$exist:"true"}},

... {name:{$type:"string"}},

... {sal:{$gt:1000}},

... {city:{$in:['Pune','Mumbai','Chennai','Delhi']}},

... {grade:{$in:['A','B','C']}}]}})

{ "ok" : 1 }

-----------------------------------------------------------------

2. Insert the documents in collection student

> db.Emp.insert({emp\_id:1,name:"Suyash",city:"Pune",age:20,salary:100005525,grad

e:"A"})

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({emp\_id:2,name:"Dilip",city:"Pune",age:20,salary:1000055,grade:"

A"})

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({emp\_id:3,name:"Rutvik",city:"Pune",age:20,salary:100004555,grad

e:"A"})

WriteResult({ "nInserted" : 1 })

-----------------------------------------------------------------

3. Write a MongoDB query to display the total no. of documents in the collection employee

> db.Emp.find().count();

3

-----------------------------------------------------------------

4. Update the document with the salary to 30000 whose grade Is A.

> db.Emp.find().pretty()

{

"\_id" : ObjectId("5b7d2d75565782858d99ecfe"),

"emp\_id" : 1,

"name" : "Suyash",

"city" : "Pune",

"age" : 20,

"salary" : 100005525,

"grade" : "A"

}

{

"\_id" : ObjectId("5b7d2db4565782858d99ecff"),

"emp\_id" : 2,

"name" : "Dilip",

"city" : "Pune",

"age" : 20,

"salary" : 1000055,

"grade" : "A"

}

{

"\_id" : ObjectId("5b7d2dd3565782858d99ed00"),

"emp\_id" : 3,

"name" : "Rutvik",

"city" : "Pune",

"age" : 20,

"salary" : 100004555,

"grade" : "A"

}

> db.Emp.update({grade:"A"},{$set:{salary:30000}},{multi:true})

WriteResult({ "nMatched" : 3, "nUpserted" : 0, "nModified" : 3 })

> db.Emp.find().pretty()

{

"\_id" : ObjectId("5b7d2d75565782858d99ecfe"),

"emp\_id" : 1,

"name" : "Suyash",

"city" : "Pune",

"age" : 20,

"salary" : 30000,

"grade" : "A"

}

{

"\_id" : ObjectId("5b7d2db4565782858d99ecff"),

"emp\_id" : 2,

"name" : "Dilip",

"city" : "Pune",

"age" : 20,

"salary" : 30000,

"grade" : "A"

}

{

"\_id" : ObjectId("5b7d2dd3565782858d99ed00"),

"emp\_id" : 3,

"name" : "Rutvik",

"city" : "Pune",

"age" : 20,

"salary" : 30000,

"grade" : "A"

}

-----------------------------------------------------------------

6. Replace the document whose empid is 3.

> db.Emp.find({emp\_id:3}).pretty()

{

"\_id" : ObjectId("5b7d2dd3565782858d99ed00"),

"emp\_id" : 3,

"name" : "Rutvik",

"city" : "Pune",

"age" : 20,

"salary" : 30000,

"grade" : "A"}

> db.Emp.replaceOne({emp\_id:3},{city:"chennai",grade:"C",salary:5000,name:"Aayush",age:21},{upsert:true})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.Emp.find({emp\_id:3}).pretty()

{

"\_id" : ObjectId("5b7d2dd3565782858d99ed00"),

"emp\_id" : 3,

"city" : "chennai",

"grade" : "C",

"salary" : 5000,

"name" : "Aayush",

"age" : 21

}

-----------------------------------------------------------------

7. Find the document with name Dilip and replace it with Gaurav (use findoneAndReplace)

> db.Emp.find({name:'Dilip'}).pretty()

{

"\_id" : ObjectId("5b7d2db4565782858d99ecff"),

"emp\_id" : 2,

"name" : "Dilip",

"city" : "Pune",

"age" : 20,

"salary" : 30000,

"grade" : "A"

}

> db.Emp.findOneAndReplace({name:"Dilip"},{name:"Gaurav"},{upsert: true})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.Emp.find({name:'Gaurav'}).pretty()

{

"\_id" : ObjectId("5b7d2db4565782858d99ecff"),

"emp\_id" : 2,

"name" : "Gaurav",

"city" : "Pune",

"age" : 20,

"salary" : 30000,

"grade" : "A"

}

-----------------------------------------------------------------

8. Add employee Amit with proper key values from collection if not exists to the collection .(Use of upsert)

> > db.Emp.update({name:"Amit"},{$set:{city:"delhi",emp\_id:4,grade:"B",salary:15000,age:20}},{upsert:true})

WriteResult({

"nMatched" : 0,

"nUpserted" : 1,

"nModified" : 0,

"\_id" : ObjectId("5b86328e73912495416f8bd1")

})

> db.Emp.find({name:'Amit'}).pretty()

{

"\_id" : ObjectId("5b86328e73912495416f8bd1"),

"city" : "delhi",

"emp\_id" : 4,

"grade" : "B",

"salary" : 15000,

"name" : "Amit"

}

-----------------------------------------------------------------

9. Add employee Vishal with proper key values and return the updated document

> db.Emp.update({name:"Vishal"},{$set:{city:"chennai",age:21,emp\_id:5,grade:"B",salary:19000}},{upsert:true})

WriteResult({

"nMatched" : 0,

"nUpserted" : 1,

"nModified" : 0,

"\_id" : ObjectId("5b863e0c93782dd18802e02e")

})

> db.Emp.find({name:'Dilip'}).pretty()

{

"\_id" : ObjectId("5b863e0c93782dd18802e02e"),

"city" : "chennai",

"emp\_id" : 5,

"grade" : "B",

"salary" : 19000,

"name" : "Vishal"

}

-----------------------------------------------------------------

10. Remove the documents where city is Delhi .

> db.Emp.find({city:'delhi'}).pretty()

{

"\_id" : ObjectId("5b86328e73912495416f8bd1"),

"city" : "delhi",

"empid" : 4,

"grade" : "B",

"salary" : 15000,

"name" : "Amit"

}

> db.Emp.remove({city:"delhi"})

WriteResult({ "nRemoved" : 1 })

> db.Emp.find({city:'Delhi'}).pretty()

-----------------------------------------------------------------