Assignment 9

MongoDB Queries: Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.).

> use Abhi;

> db.createCollection('Student');

{ "ok" : 1 }

> db.Student.insert({'Rno':'1','Name':'Piyush','Class':'TE

COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'2','Name':'Abhi','Class':'TE

COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'3','Name':'Ashley','Class':'TE

COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'4','Name':'Hitesh','Class':'TE

COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'5','Name':'Pratik','Class':'TE

COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'6','Name':'Pratik','Class':'TE

COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.find();

> db.Student.find().pretty();

> db.Student.update({'Name':'Hitesh'},{$set:

{'Name':'Henry'}});

> db.Student.find().pretty();

> db.Student.remove({'ADD':'MP'});

> db.Student.find().pretty();

>db.Student.save({\_id:ObjectId("5b8fad4ef00832a0a50b5036"),"RNO ":"1","NAME":"PIYUSH","CLASS":"TE COMP","ADD":"MP"});

> db.Student.find().pretty();

> db.Student.find({$and:[{"Name":"Piyush"},{"Rno":"2"}]});

> db.Student.find({$and:[{"Name":"Piyush"},

{"Rno":"1"}]}).pretty();

> db.Student.find({$and:[{"Name":"Piyush"},

{"Rno":"2"}]}).pretty();

> db.Student.find({$or:[{"Name":"Piyush"},

{"Rno":"2"}]}).pretty();

> db.Student.find({$or:[{"Name":"Piyush"},{"Class":"TE

COMP"}]}).pretty();

> db.Student.find({$nor:[{"Name":"Piyush"},{"Class":"TE

COMP"}]}).pretty();

> db.Student.find({$nor:[{"Name":"Piyush"},

{"Rno":"2"}]}).pretty();

> db.Student.find({"Rno":{$not:{$lt:"3"}}}).pretty();

> db.Student.find( {"Rno": { $eq:"5"}}).pretty();

> db.Student.find( {"Rno": { $ne:"5"}}).pretty();

> db.Student.find( {"Rno": { $gt:"5"}}).pretty();

> db.Student.find( {"Rno": { $gte:"5"}}).pretty();

> db.Student.find( {"Rno": { $lt:"5"}}).pretty();

> db.Student.find( {"Rno": { $lte:"5"}}).pretty();

> db.Student.find( {"Rno": { $lt:"5",$gt:"2"}}).pretty();

> db.Student.find( {"Rno": { $lte:"5",$gte:"2"}}).pretty();

> db.Student.find( {"Rno": { $lte:"5",$gt:"2"}}).pretty();

> db.Student.find( {"Rno": { $lt:"5",$gte:"2"}}).pretty();

Assignment 10

MongoDB - Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB

//USE DATABASE

> use comp;

switched to db comp //CREATE COLLECTION WEBSITE

> db.createCollection('website');

{ "ok" : 1 }

//INSERT VALUES IN WEBSITE

>

db.website.insert({'roll':'1','name':'harsh','amount':1000,'ur l':'www.yahoo.com'}); WriteResult({ "nInserted" : 1 })

>

db.website.insert({'roll':'2','name':'jitesh','amount':2000,'u rl':'www.yahoo.com'});

WriteResult({ "nInserted" : 1 })

>

db.website.insert({'roll':'3','name':'rina','amount':3000,'url ':'www.google.com'});

WriteResult({ "nInserted" : 1 })

>

db.website.insert({'roll':'4','name':'ash','amount':4000,'url' :'www.gmail.com'});

WriteResult({ "nInserted" : 1 })

>

db.website.insert({'roll':'5','name':'ash','amount':1000,'url' :'www.pvg.com'});WriteResult({ "nInserted" : 1 })

//SUM AGGREGATE

> db.website.aggregate({$group:{\_id:"$name","total":

{$sum:"$amount"}}});

//AVG AGGREGATE

> db.website.aggregate({$group:{\_id:"$name","total":

{$avg:"$amount"}}});

//MIN AGGREGATION

> db.website.aggregate({$group:{\_id:"$name","total":

{$min:"$amount"}}});

//MAX AGGREGATION

> db.website.aggregate({$group:{\_id:"$name","total":

{$max:"$amount"}}});

//FIRST AGGREGATION

> db.website.aggregate({$group:{\_id:"$name","total":

{$first:"$amount"}}});

//LAST AGGREGATION

> db.website.aggregate({$group:{\_id:"$name","total":

{$last:"$amount"}}});

//PUSH AGGREGATION

> db.website.aggregate({$group:{\_id:"$name","total":

{$push:"$amount"}}});

{ "\_id" : "ash", "total" : [ 4000, 1000 ] }

{ "\_id" : "rina", "total" : [ 3000 ] }

{ "\_id" : "jitesh", "total" : [ 2000 ] }

{ "\_id" : "harsh", "total" : [ 1000, 1000 ] }

//COUNT AGGREGATION

> db.website.aggregate({$group:{\_id:"$name","total":

{$sum:1}}});

//ADDTOSET AGGREGATE

> db.website.aggregate({$group:{\_id:"$name","total":

{$addToSet:"$amount"}}});

//INDEXING

> db.createCollection('website1');

{ "ok" : 1 }

> db.website1.insert({'r':1,'name':'harsh'});

WriteResult({ "nInserted" : 1 })

> db.website1.find().pretty()

> db.website1.createIndex({'name':1})

//CREATE INDEXING

> db.website1.createIndex({'name':-1})

> db.website1.getIndexses()

> db.website1.getIndexes()

> db.website1.createIndex({'name':-1})

//DROP INDEX

> db.website.dropIndex({'name':-1})

> db.website1.dropIndex({'name':1})

//GET INDEXING

> db.website1.getIndexes()

> db.website1.find().pretty()

> db.website1.createIndex({'name':1})

> db.website1.getIndexes()

> db.website1.dropIndex({'name':1})

> db.website1.getIndexes()

> db.website1.createIndex({'name':1,'r':-1})

> db.website1.getIndexes()

Assignment 11

MongoDB - Map reduces operations: Implement Map reduces operation with suitable example using MongoDB.

> use Abhi switched to db Abhi

> db.createCollection('Journal');

{ "ok" : 1 }

> db.Journal.insert({'book\_id':1,'book\_name':'Javacd OOP','amt':500,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':1,'book\_name':'Java OOP','amt':400,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book\_id':1,'book\_name':'Java','amt':300,'s tatus':'Not Available'}); WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book\_id':2,'book\_name':'Java','amt':300,'s tatus':'Available'});

WriteResult({ "nInserted" : 1 })

>db.Journal.insert({'book\_id':2,'book\_name':'OPP','amt':200,'status':'Available'}); WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':2,'book\_name':'C++','amt':200,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':3,'book\_name':'C++','amt':150,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':3,'book\_name':'C++','amt':200,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':4,'book\_name':'OPP C++','amt':300,'status':'Not Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':5,'book\_name':'OPP C++','amt':400,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':5,'book\_name':'C++','amt':400,'status':'Available'});

WriteResult({ "nInserted" : 1 })

> db.Journal.insert({'book\_id':5,'book\_name':'C++ Java','amt':400,'status':'Not Available'});

> var mapfunction=function(){ emit(this.book\_id,this.amt)};

> var reducefunction=function(key,value){return Array.sum(value);};

> db.Journal.mapReduce(mapfunction,reducefunction,{'out':'new'});

> db.Journal.mapReduce(mapfunction,reducefunction,{'out':'new'}).find().pretty();

> db.new.find().pretty();