**MONGODB PROJECT-2**

**Question: 1**

Q.1)

db.categories.update({},{$unset: {Level:1}},false,true)

db.categories.update({},{$unset: {Name:1}},false,true)

var ancestors = [];

var stack = [];

var Level=0;

var item = db.categories.findOne({\_id: "MongoDB"});

stack.push(item);

while (stack.length > 0) {

var current = stack.pop();

var parents = db.categories.find( {\_id: current.parent});

while (parents.hasNext() == true) {

var parent = parents.next();

Level=Level+1;

db.categories.update(

{ \_id: parent.\_id },

{ $set: { Name:parent.\_id,Level: Level} }

);

var parent1=db.categories.findOne( {\_id:parent.\_id},{Name:1,Level:1,\_id:0});

ancestors.push(parent1);

stack.push(parent);

}

}

ancestors;

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

Q.2)

db.categories.update({},{$unset: {Level:1}},false,true)

db.categories.update({},{$unset: {Name:1}},false,true)

var descendants = [];

var stack = [];

var Level=0;

var height=0;

var item = db.categories.findOne({\_id: "Books"});

stack.push(item);

while (stack.length > 0) {

var current = stack.pop();

var children = db.categories.find( {parent: current.\_id});

Level=Level+1;

while (children.hasNext() == true) {

var child = children.next();

db.categories.update(

{ \_id: child.\_id },

{ $set: {Level: Level} }

);

var child1=db.categories.findOne( {\_id:child.\_id},{Level:1});

descendants.push(child1);

stack.push(child);

height=Level+1;

}

}

height;

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

Q.3)

var x = db.categories.findOne({children: "dbm"}).\_id;

x;

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

Q.4)

var descendants=[]

var stack=[];

var item = db.categories.findOne({\_id: "Books"});

stack.push(item);

while (stack.length>0){

var currentnode = stack.pop();

var childrens = db.categories.find({\_id: {$in: currentnode.children}});

while(true === childrens.hasNext()) {

var child = childrens.next();

descendants.push(child.\_id);

if(child.children.length>0){

stack.push(child);

}

}

}

descendants;

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

Q.5)

var parentDoc = db.categories.findOne( {children: "Databases"});

db.categories.aggregate(

{$unwind:"$children"},

{$match:{"\_id":parentDoc.\_id}},

{$project:{children:1,\_id:0,childrenNe:{$ne:["$children","Databases"]}}},

{$match:{"childrenNe":true}},

{$project:{children:1}}

)

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

**Question: 2**

Q.1)

var map = function() {

for( var i in this.awards) {

emit (this.awards[i].award,

{count: 1});

}

}

var reduce = function(key, values) {

var totalCount = 0;

values.forEach(function(value) {

totalCount++;

});

return {

count: totalCount

}

};

db.test.mapReduce(map, reduce, { out: { inline: 1 }})

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

Q.2)

db.test.find({birth: {$not: {$type: 9}}})

db.test.remove({\_id:10})

db.test.aggregate(

{$group:{\_id:{year:{$year:"$birth"}},id:{$addToSet:"$\_id"}}}

)

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

Q.3)

var large = db.test.find().sort({\_id: -1}).limit(1)

var small = db.test.find().sort({\_id:  1}).limit(1)

small.forEach(function(mydoc)

                {

                  printjson(mydoc);

                    });

large.forEach(function(mydoc2)

                {

                  printjson(mydoc2);

                    });

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

Q.4)

db.test.createIndex( {"awards.award": "text"})

db.test.find( { $text: { $search: "\"Turing Award\""}})

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

Q.5)

db.test.createIndex( {"awards.award": "text"})

db.test.find( { $text: { $search: "Turing \"National Medal\""}})

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