1. **Demonstrate the usage of $match, $group, aggregate pipelines.**

**Queries :**

$match

db.employees.aggregate([{ $match : { Name:”Pavithra” }}]).pretty()

$group

db.employees.aggregate([{ $group : {\_id:”$Name”}}])

aggregrate pipelines

db.employees.aggregate([{$match:{salary:{$gt:30000}}},{$group:{\_id:”$Name”}}]).pretty()

1. **Demonstrate the map-reduce aggregate function of this dataset**

**Query**

Var mapFunc = function(){emit(this.Name, this.salary);};

Var reduceFunc = function(empname,empsalary){return Array.sum(empsalary);};

db.employees.mapReduce(mapFunc, reducerFunc, {out:”output”});

1. **Find out all the employees, who are paying tax>50000 and are not eligible for pay raise.**

**Query**

db.employees.find({$and: [{tax: { $gt: 50000}},{ eligible\_for\_payraise: false }]}).pretty()

1. **Demonstrate the alter and drop commands on this dataset.**

**Query**

Alter

db.employees.update({“Name”: “Pavithra”},{$set: {salary:50000}});

Drop

db.output.drop();