Name : Anushree Gattani
Roll No. : 31404
Batch : K4
DBMSL

Assignment 11

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

Querie (with outputs) - MapReduce

1. mydb> db.collection1.insertMany([
{
product: "Product A",
category: "Category 1",
sales: 100,
price: 29.99,
status: "active"
},
{
product: "Product B",
category: "Category 2",
sales: 150,
price: 19.99,
status: "active"
},
{
product: "Product C",
category: "Category 1",
sales: 80,
price: 39.99,
status: "inactive"
},
{
product: "Product D",
category: "Category 3",
sales: 200,
price: 49.99,
status: "active"
},
{
product: "Product E",
category: "Category 1",
sales: [100, 150],
price: 29.99,
status: "active"
},
{
product: "Product F",
category: "Category 2",
sales: [200],
price: 19.99,

```
    status: "active"
  },
  {
    product: "Product G",
    category: "Category 1",
    sales: [80],
    price: 39.99,
    status: "inactive"
  },
  {
    product: "Product H",
    category: "Category 3",
    sales: [300, 100],
    price: 49.99,
    status: "active"
  }
]);
```

```
  {
    _id: ObjectId('66f0ed1631fdddb1ee964038'),
    product: 'Product F',
    category: 'Category 2',
    sales: [ 200 ],
    price: 19.99,
    status: 'active'
  },
  {
    _id: ObjectId('66f0ed1631fdddb1ee964039'),
    product: 'Product G',
    category: 'Category 1',
    sales: [ 80 ],
    price: 39.99,
    status: 'inactive'
  },
  {
    _id: ObjectId('66f0ed1631fdddb1ee96403a'),
    product: 'Product H',
    category: 'Category 3',
    sales: [ 300, 100 ],
    price: 49.99,
    status: 'active'
  }
]
```

2. Find sum of prices according to category

```
mydb> db.collection1.mapReduce(
function()
{
        emit(this.category,this.price);
},
function(key,value)
{
        return Array.sum(value);},
{
        query:{status:"active"},
        out:"outprice"
});
```

```
mydb> db.collection1.mapReduce(function(){emit(this.category,this.price);},function(key
,value){return Array.sum(value);},{query:{status:"active"},out:"outprice"});
{ result: 'outprice', ok: 1 }
mydb> db.outprice.find();
[
  { _id: 'Category 3', value: 99.98 },
  { _id: 'Category 1', value: 59.98 },
  { _id: 'Category 2', value: 39.98 }
]
```

3. Find min of price

```
mydb> db.collection1.mapReduce(
function()
{
        emit(this.category,this.price);
},
function(key,value)
{
```

```
        return Math.min.apply(null,value);
},
{
        query:{status:"active"},
        out:"minprice"
});
```