­­Assignment -2 Big Data

####### 3.1

var map1 = function() {

emit(this.stock\_symbol, this.stock\_price\_high)

}

var reduce1 = function (stock, prices) {

return (prices.reduce((accumulator, value) => {return accumulator + value;}, 0)/prices.length)

}

db.stocks.mapReduce(map1, reduce1, {out:"output\_3\_1"})

Text

Description automatically generated

####### 3.2

var map2 = function() {

emit(this.stock\_symbol, { sum: this.stock\_price\_high, count: 1})

}

var reduce2 = function (stock, prices) {

var sum = 0, count = 0;

for(var idx =0; idx<prices.length;idx++){

sum += prices[idx].sum;

count += prices[idx].count;

}

return {sum:sum, count:count};

}

var finilize2 = function(key, val){

val.avg = val.sum/val.count;

return val;

}

db.stocks.mapReduce(map2, reduce2, {out:"output\_3\_2", finalize:finilize2 })

Text

Description automatically generated

########4

db.stocks.aggregate(

[

{

$group:

{

\_id: "$stock\_symbol",

avg\_price\_open: { $avg: "$stock\_price\_open" },

avg\_price\_high: { $avg: "$stock\_price\_high" },

avg\_price\_low: { $avg: "$stock\_price\_low" },

avg\_price\_close: { $avg: "$stock\_price\_close" },

avg\_price\_adj\_close: { $avg: "$stock\_price\_adj\_close" },

}

}

]

)

Text

Description automatically generated

#################5.1

var map3 = function() {

emit(this.Gender, 1)

}

var reduce3 = function (key, prices) {

count=0;

for(var idx =0; idx<prices.length;idx++){

count+=1;

}

return count;

}

db.users.mapReduce(map3, reduce3, {out:"output\_5\_1\_1"})

db.output\_5\_1\_1.find()

Text

Description automatically generated

###############5.1.2

var map4 = function() {

var yearStr = this.Title.substring(this.Title.length-5, this.Title.length-1);

emit(yearStr, 1)

}

var reduce4 = function (key, prices) {

count=0;

for(var idx =0; idx<prices.length;idx++){

count+=1;

}

return count;

}

db.movies.mapReduce(map4, reduce4, {out:"output\_5\_1\_2"})

Text

Description automatically generated

###############5.1.3

var map5 = function() {

emit(this.Rating, 1)

}

var reduce5 = function (key, prices) {

count=0;

for(var idx =0; idx<prices.length;idx++){

count+=1;

}

return count;

}

db.ratings.mapReduce(map5, reduce5, {out:"output\_5\_1\_3"})

Text

Description automatically generated

###############5.2.1

db.users.aggregate(

[

{

$group:

{

\_id: "$Gender",

count: { $count :{}}

}

}

]

)

Text

Description automatically generated

###############5.2.2

db.movies.aggregate(

[

{

$group:

{

\_id: {$substrCP:["$Title",

{$subtract: [{$strLenCP:"$Title"},5]},

{$subtract: [{$strLenCP:"$Title"},4]}

]},

count: { $count :{}}

}

}

]

)

Text

Description automatically generated

###############5.2.3

db.ratings.aggregate(

[

{

$group:

{

\_id: "$Rating",

count: { $count :{}}

}

}

]

)

Text

Description automatically generated

###############6.1.1

var map6 = function() {

emit(this.ipaddress, 1)

}

var reduce6 = function (key, prices) {

count=0;

for(var idx =0; idx<prices.length;idx++){

count+=1;

}

return count;

}

db.access.mapReduce(map6, reduce6, {out:"output\_6\_1\_1"})

Text

Description automatically generated

###############6.1.2

var map7 = function() {

var month= this.time.substr(3,3)

emit(month, 1)

}

var reduce7 = function (key, prices) {

count=0;

for(var idx =0; idx<prices.length;idx++){

count+=1;

}

return count;

}

db.access.mapReduce(map7, reduce7, {out:"output\_6\_1\_2"})

Text

Description automatically generated

###############6.2.1

db.access.aggregate(

[

{

$group:

{

\_id: "$ipaddress",

count: { $count :{}}

}

}

]

)

Graphical user interface, text

Description automatically generated

###############6.2.2

db.access.aggregate(

[

{

$group:

{

\_id: {$substrCP:["$time",3, 3]},

count: { $count :{}}

}

}

]

)

Text

Description automatically generated